Comparative Study of Genetic Algorithm Performed in a Single Generation for two Different Fitness Functions Technique f(x) = x^2 and f(x) = x^2+1

Dipanjan Kumar Dey
Department of Computer Science & Engineering,
Assistant professor at Prajnanananda Institute of Technology and Management (PITM)
Dist-Kolkata, West Bengal (India)

ABSTRACT
A genetic algorithm is one of a class of algorithms that searches a solution space for the optimal solution to a problem. First part of this work consists of basic information about Genetic algorithm like what are Individual, Population, Crossover, Genes, Binary Encoding, Flipping, Crossover probability, Mutation probability. What is it used for, what is their aim. In this article the methods of selection, crossover and mutation are specified. In the second part of this paper providing two different fitness functions $f(x) = x^2$ and $f(x) = x^2+1$.Solving maximizing problems for two different fitness functions $f(x) = x^2$ and $f(x) = x^2+1$ using genetic algorithm in a single generation. A single generation of a Genetic algorithm is performed here with encoding, selection, crossover and mutation. In this paper shown the best string from initial population is same (identical) for two different fitness functions $f(x) = x^2$ and $f(x) = x^2+1$.The purpose of this paper is to present a specific varying fitness function (multiple fitness function) technique. The author of this paper was among the first that proposed the different fitness function technique used in GA for selecting the best string.

Keywords
Genetic algorithm, optimization, selection, crossover, mutation

1. INTRODUCTION
The Genetic Algorithm (GA) is one of the most important methods used to solve many combinatorial optimization problems. Chromosomes are selected from the population to be parents to crossover. The problem is how to select these chromosomes. Genetic algorithm is based on the Darwin's theory of evolutions; the basic rule is "survival of the fitness" i.e. According to Darwin's evolution theory the best ones should survive and create new offspring. There are many methods how to select the best chromosomes, for example roulette wheel selection, Boltzmann selection, tournament selection, rank selection, steady state selection and some others. Here in this paper solving maximizing problems for two different fitness functions $f(x) = x^2$ and $f(x) = x^2+1$ to select these chromosomes the best ones should survive and create new offspring using Genetic algorithm (GA). Genetic algorithm (GA) is however used the term fitness function instead of objective function and that is the term used in this paper. Another thing is that, in solving a problem is to have some means or procedure for discriminating good solutions from bad solutions. The fitness function $f(x) = x^2 + n$, where $n=0,\pm1,\pm2,\pm3,\dots\dots$.... n may be fraction also. For easy calculation taking the integer values of n say n=0 and n=1. Here using two different fitness functions $f(x) = x^2$ for n=0 and $f(x) = x^2+1$for n=1. For different values of n different fitness functions are produced and give the best string from initial population which is same (identical) result, i.e. the best solution does not change for two different fitness function $f(x) = x^2$ and $f(x) = x^2+1$.

2. GENETIC ALGORITHMS
The basic principles of genetic algorithms (GAs) were introduced by Holland [10].GAs operate on "populations" of potential solutions. Which are usually referred to as "chromosomes". Each chromosome represents a set of parameters for a given problem. The chromosomes evaluate to represent the best solutions for a recombination process, which produces new chromosomes. The new, improved chromosomes replace those with weaker solutions. In this way, each new generation becomes closer to the optimal solution. This continues for many generations until the termination condition is met. Mutations and different combining strategies ensure that a large range of search space is discovered [10]. Crossover is the most important operation of a GA because in this operation, characteristics are exchanged between the individuals of the population.

The main feature of genetic algorithms is to combine both exploration and exploitation in an optimal way. The Exploitation and exploration techniques are responsible for the performance of genetic algorithms. Exploitation means to use the already existing information to find out the better solution and Exploration is to investigate new and unknown solution in exploration space. The authority of genetic algorithms comes from their ability to combine both exploration and exploitation in an optimal way [11]. A generic genetic algorithm consists of following operations namely: Initialization, Selection, Reproduction and Replacement. Crossover and mutation are used to maintain balance between exploitation and exploration. During replacement, the old individuals are replaced by new offspring's [12].

3. OPTIMIZATION
GA can be used in optimization problems when the solution space grows very quickly, or an exact solution is required in a limited time. Optimization is a process that finds a best, or optimal, the solution for the problem. The optimization problems are centered on three factors such as—

1) An objective function- which is to be maximized or minimized. For example, in GA for selecting the best string from initial population we want to maximize the strength. [3]
2) A set of decision variable or unknowns—these variables affect the objective function. The decision variables are independent variables in the optimization problem. For example in this paper x is

used as decision variable, where x can take values 0 and 31, Here 0 for the string 00000 and 31for the string11111.

3) A set of constraints—that allow the decision variable to take on certain values but exclude others. [3]

Thus an optimization problem is defined as finding values of the variables that maximized or minimized the objective function while satisfying the constraints. [3] [7]

4. INDIVIDUALS

An individual is a single solution. Each individual has fitness. An individual is encoded as a string of binary digits.

5. GENES

Genes are the basic instructions for building a GA.A chromosome is a sequence of genes. A gene represents some data.(eye color, hair color).[3] A gene looks like 11100010 (in binary form). A Gene is a part of chromosome. A gene contains a part of solution. For example if 162759 is a chromosome then 1, 6, 2, 7, 5 and 9 are its genes.

6. POPULATION

A population is a collection of individuals. Population being a combination of various chromosomes. The population size remains constant from generation to generation.

Population	Chromosome 1	0 1 1 0 0
	Chromosome 2	1 1 0 0 1
	Chromosome 3	0 0 1 0 1
	Chromosome 4	1 0 0 1 1

This figure shows the population consists of four chromosomes each having five bits.

7. BINARY ENCODING

Binary encoding is the most common to represent information contained. In genetic algorithm (GA) it was first used because of its relative simplicity. In binary encoding, every chromosome encodes a binary string of bits: 0 or 1are mostly used [3], like

Chromosome 1: 1 0 1 1 0 0 1 0 1 1 0 0 1 0 1 0 1 1 1 0 0 1 0 1

Chromosome 2: 1 1 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 1 1

8. SELECTION

Selection is the stage of a GAs in which individual chromosomes are chosen from a population for recombination (or crossover).

Selection allocates more copies of those solutions with higher fitness values. If the fitness function is higher the better chance that an individual will be selected. The main idea of selection is to prefer better solutions to worse ones.

8.1 Fitness Function

The fitness function plays a very important role in guiding GA. Good fitness functions will help GA to explore the search space more effectively and efficiently. Bad fitness functions can easily make GA get trapped in a local optimum solution and lose the discovery power.

8.2 Fitness Proportionate Selection

This includes methods such as roulette-wheel selection (Holland, 1975; Goldberg, 1989b).In roulette-wheel selection; each individual in the population is assigned a roulette wheel slot sized in proportion to its fitness. That is, in the biased roulette wheel, good solutions have a larger slot size than the less fit solutions. The roulette wheel is spun to obtain a reproduction candidate. The roulette wheel selection scheme can be implemented as follows: [4]

1. Evaluate the fitness, f_i , of each individual in the population.
2. Compute the probability (slot size), P_i, of selecting each memberof the population: $P_i = = \frac{f_i}{\sum_{j=1}^{n} f_j}$ where n = population size
3. Calculate the cumulative probability, q_i, for each individual: , $q_i=\sum_{j=1}^{i} P_j$
4. Generate a uniform random number, $r\in(0,1]$
5. If $r<$ q_1, then select the first chromosome,x_1, else select the individual$x_i$such that $q_{i-1}<r \leq q_i$.
6. Repeat steps 4–5 *n* times to create *n* candidates in the mating pool

To illustrate, For the fitness function $f(x) = x^2$,

Let us consider a population with four individuals ($n = 4$), withthe fitness function $f(x) = x^2$, fitness values as shown in the table below. [4]

Here initial population of size 4 is randomly chosen (01100, 11001, 00101,10011)

String no.	Initial population (randomly selected)	Value of the variable x	fitness function $f(x) = x^2$	Probability (p_i) of selection	Cumulative Probability q_i
S_1	01100	12	$144=f(x_1)$	$\frac{144}{1155}=0.1247=P_1$	$0.1247=q_1$
S_2	11001	25	$625= f(x_2)$	$\frac{625}{1155}=0.5411=P_2$	$0.6658=q_2$
S_3	00101	5	$25= f(x_3)$	$\frac{25}{1155}=0.0216=P_3$	$0.6874=q_3$
S_4	10011	19	$361= f(x_4)$	$\frac{361}{1155}=0.3126=P_4$	$1.0000=q_4$
Sum			$1155=\sum f(x_i)$	$1.0000=\sum p_i$	

Now if we generate a random numberr,

say 0.6712, $q_2 = 0.6658<0.6712 \leq q_3 =0.6874$

Therefore third chromosome (S_3) is selected. [4]

For the fitness function $f(x) = x^2+1$,

Let us consider a population with four individuals ($n = 4$), with the fitness function $f(x) = x^2$, fitness values as shown in the table below. [4]

Here initial population of size 4 is randomly chosen (01100, 11001, 00101, 10011)

String no.	Initial population (randomly selected)	Value of the variable x	fitness function f(x) = x^2+1	Probability (p_i) of selection	Cumulative Probability q_i
S_1	01100	12	145=f(x_1)	$\frac{145}{1159}$ =0.1251=P_1	0.1251=q_1
S_2	11001	25	626=f(x_2)	$\frac{626}{1159}$ =0.5401= P_2	0.6652=q_2
S_3	00101	5	26= f(x_3)	$\frac{26}{1159}$ =0.0224= P_3	0.6876=q_3
S_4	10011	19	362=f(x_4)	$\frac{362}{1159}$ =0.3123= P_4	0.9999≈ 1 = q_4
Sum			1159= $\sum f(x_i)$	0.9999≈ 1 = $\sum p_i$	

Now if we generate a random number r, say 0.6712, q_2 = 0.6652<0.6712 ≤ q_3 =0.6876

Therefore third chromosome (S_3) is selected. [4].

Thus we see that for two different fitness function f(x) = x^2 and f(x) = x^2+1, third chromosome (S_3) is selected for generate a random number r, say 0.6712,

9. CROSSOVER OR RECOMBINATION

Crossover is the most important operation of a GA because in this operation, characteristics are exchanged between the individuals of the population. Crossover is a genetic operator that helps in joining two chromosomes to form a new chromosome.

The main distinctive feature of GA is the use of crossover. Crossover, also called recombination, is an operator that creates new individuals from current population, combining pieces of information coming from different individuals of the population. In fact, it recombines genetic material of two parent individuals to create the offspring of the next generation.

Crossover selects genes from parent chromosomes and creates a new offspring. The simplest way how to do this is to choose randomly some crossover point and everything before this point copy from a first parent and then everything after a crossover point copy from the second parent. 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.

Number of crossovers depends on crossover-rate. Generally crossover rate is2 to 5%.

Number of Crossover =

$$\frac{(No.of\ cells\ in\ a\ chromosome\ *\ No.of\ chromosomes\ *\ crossover\ rate)}{200}$$

The Crossover operators are of many types.

−one simple way is, One-Point crossover or single-point crossover,

The others are Two Point crossover, Uniform crossover, Arithmetic crossover, precedence preservative crossover (PPX), partially matched crossover PMX) and Heuristic crossovers.

Here I am discussing only One-Point crossover or single-point crossover because in this work only it is required. [8]

Crossover can then look like this (| is the crossover point):

Chromosome 1	11011 | 00100110110
Chromosome 2	11011 | 11000011110
Offspring 1	11011|11000011110
Offspring 2	11011 |00100110110

9.1 One-Point Crossover

One-Point crossover operator randomly selects one crossover point and then copy everything before this point from the first parent and then everything after the crossover point copy from the second parent. The Crossover would then look as shown below.

Consider the two parents selected for crossover.

Parent 11 1 0 1 1 |0 0 1 0 0 1 1 0 1 1 0
Parent 21 1 0 1 1 |1 1 0 0 0 0 1 1 1 1 0

Interchanging the parents chromosomes after the crossover points -

The Offspring produced are:

Offspring 11 1 0 1 1|1 1 0 0 0 0 1 1 1 1 0
Offspring 21 1 0 1 1|0 0 1 0 0 1 1 0 1 1 0

Note: The symbol a vertical line |is the chosen as crossover point.

10. MUTATION

Mutation is a very easy method to create a new individual. After a crossover is performed, mutation takes place. A mutation can be applied in several positions, once a new individual "is born". This is to prevent falling all solutions in population into a local optimum of solved problem. Mutation changes randomly the new offspring. For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. Mutation can then be following:

Original offspring 1	1101111000011110
Original offspring 2	1101100100110110
Mutated offspring 1	1100111000011110
Mutated offspring 2	1101101100110110

The mutation depends on the encoding as well as the crossover. For example when we are encoding permutations, mutation could be exchanging two genes. [7]

11. CROSSOVER AND MUTATION PROBABILITY

There are two basic parameters of GA - crossover probability and mutation probability.

11.1 Crossover Probability (P_c)

In most recombination operators, two individuals are randomly selected and are recombined with a probability p_c, called the crossover probability. That is, a uniform random number, r, is generated and if $r \leq p_c$, the two randomly selected individuals undergo recombination. Otherwise, that is, if $r > p_c$, the two offspring are simply copies of their parents.

Crossover probability (P_c) says how often will be crossover performed. If there is no crossover, offspring is exact copy of parents. If there is a crossover, offspring is made from parts of parents' chromosome.

If crossover probability **(P_c)** is **100%**, then all offspring is made by crossover.

If crossover probability **(P_c)** is **0%**, whole new generation is made from exact copies of chromosomes from old population (but this does not mean that the new generation is the same!). Crossover is made in hope that new chromosomes will have good parts of old chromosomes and maybe the new chromosomes will be better. However it is good to leave some part of population survive to next generation. [4]

11.2 Mutation Probability (P_m)

Mutation probability (P_m) says how often will be parts of chromosome mutated. If there is no mutation, offspring is taken after crossover (or copy) without any change. If mutation is performed, part of chromosome is changed. If (P_m) is 100%, whole chromosome is changed, if (P_m) is 0%, nothing is changed.

Mutation is made to prevent falling GA into local extreme, but it should not occur very often, because then GA will in fact change to random search.

Note that the performance of GA is largely influence by two operators called crossover and mutation. These two operators are most important parts in GA.

12. FLIPPING

Flipping of a bit involves changing 0 to 1 and 1 to 0 based on a mutation chromosome generated.

Consider a parent and a mutation chromosome is randomly generated. For a 1 in mutation chromosome, the corresponding bit in parent chromosome is flipped (0 to1and 1to 0).In the following table 1 occurs at 3 places of mutation chromosome, the corresponding bits in parent chromosome are flipped and the child is generated.

Parent	1011 0101
Mutation chromosome	1000 1001
Child	0011 1100

Figure-Mutation flipping concept

13. SOLVING MAXIMIZING PROBLEMFOR TWO DIFFERENT FITNESS FUNCTIONS f(x) = x^2 AND f(x) = x^2+1 USING GENETIC ALGORITHM

A simple example will help us to understand how a GA performed in a single generation. [6]

Let us consider a maximizing problem,

The objective function $f(x) = x^2$, which is to be maximized. Here for simplicity we may assume that x can take only the integer value. An n-bit string can represent integers from 0 to 2^n-1.Here using 5- bit string, so n=5 the integers from 0 to 2^5-1=31.Where x can take values 0(00000) and 31(11111).x is also known as decision variable, The decision variables are independent variables in the optimization problem.[2].$f(x) = x^2$ is also known as fitness function. Domain of f(x) = R, i.e. D_f=R, since $f(x) = x^2$ defined for all real values of x, R= real numbers. Range of f(x) = [0,∝), i.e.R_f = [0,∝).The function $f(x) = x^2$ has minima at x = 0 and the minimum value = f(0) = 0^2 =0. Note that $f(x) = x^2$ has no particular maximum value. [6]

Now for the objective function $f(x) = x^2+1$, which is also known as fitness function, which is to be maximized, Here for simplicity we may assume that x can take only the integer value,[6] where x can take values 0 and 31.x is also known as decision variable, $f(x) = x^2+1$ is also known as fitness function. Domain of f(x) = R, i.e. D_f=R, since $f(x) = x^2+1$ defined for all real values of x, R= real numbers.Since $x^2 \geq 0, \forall x \in R \therefore x^2+1 \geq 1, \forall x \in R \therefore f(x) \geq 1, \forall x \in R$.Thus range of f(x) = [1,∝), i.e. R_f = [1,∝).The functionf(x) = x^2+1 has minima at x = 0 and the minimum value = f(0) = 0^2+1 =1. Note that f(x) = x^2+1 has no maximum value.

Both the fitness functions $f(x) = x^2$ and $f(x) = x^2+1$ are symmetric in the y-axis , represent parabola. Since the coefficient of x^2 is positive so the parabola opens upwards i.e. its concavity is in the positive direction of y-axis.Therefore it is called upward parabola.

Here I am using five bits (binary integer) numbers between 0(00000) and 31(11111).

A single generation of a Genetic algorithm is performed here with encoding, selection, crossover and mutation.

Here initial population of size 4 is randomly chosen (01100, 11001, 00101, 10011).Note that any number of populations can be selected according to the requirement and application.

Another process for selecting population size 4 in the following way:-

The initial population is selected at random, which could be the toss of a coin. An individual is encoded (naturally) as a string of *l* binary digits.

We start with a population of n random strings.

Suppose that l = 5 and n = 4

We toss a fair coin $l \times n=4\times5=20$ times and gets the following initial population:

S_1 = 01100, S_2 = 11001, S_3 = 00101, S_4 = 10011

Note that population size is one of the important parameter in GA. Population size says how many chromosomes are in population (in a single generation).

1) If there are only few chromosomes, then GA would have a few possibilities to perform crossover and only a small part of search space is explored.
2) If there are many chromosomes, then GA slows down.

So be careful for selection of population size. Generally after some limit it is not useful to increase population size, because it does not help to solving the problem faster. The population size depends on the type of encoding and the problem.

<u>For the fitness function f(x) = x^2</u>

13.1 Table-1: The Presentation of Selection [5]

String no.	Initial population (randomly selected)	Value of the variable x	fitness function f(x) = x^2	Probability (p_i) of selection	% Probability of selection	Expected count = $\frac{Fitness}{Average}$	Actual count
S_1	01100	12	144=f(x_1)	0.1247 =P_1	12.47%	0.4987	1
S_2	11001	25	625= f(x_2)	0.5411 = P_2	54.11%	2.1645	2
S_3	00101	5	25= f(x_3)	0.0216 = P_3	2.16%	0.0866	0
S_4	10011	19	361= f(x_4)	0.3126 = P_4	31.26%	1.2502	1
Sum			1155= $\sum$ f(x_i)	1.0000 = $\sum p_i$	100	4.0000	4
Average			288.75	0.2500	25	1.0000	1

Calculations:

For string S_1 01100, the x value of 01100=$0*2^4+1*2^3+1*2^2+0*2^1+0*2^0$=0+8+4+0+0=12

For string S_2 11001, the x value of 11001=$1*2^4+1*2^3+0*2^2+0*2^1+1*2^0$=16+8+0+0+1=25

For string $S_3$00101, the x value of 00101=$0*2^4+0*2^3+1*2^2+0*2^1+1*2^0$=0+0+4+0+1=5

For string $S_4$10011, the x value of 10011=$1*2^4+0*2^3+0*2^2+1*2^1+1*2^0$=16+0+0+2+1=19

Now for fitness function f(x) = x^2, Calculate the fitness value,

For string S_1, x=12, f(x) = x^2 = $(12)^2$ =144

For string S_2, x=25, f(x) = x^2 = $(25)^2$ =625

For string S_3, x=5, f(x) = x^2 = $(5)^2$ =25

For string S_4, x=19, f(x) = x^2 = $(19)^2$ =361

Sum of the fitness value=$\sum f(x_i)$ =144+625+25+361=1155

Probability of selection:

For stringS$_1$, Probability P_1= $\frac{f(x_1)}{\sum f(x_i)} = \frac{144}{1155}$ =0.1247,

∴% Probability= 0.1247*100=12.47

For stringS$_2$, Probability P_2= $\frac{f(x_1)}{\sum f(x_i)} = \frac{625}{1155}$ =0.5411,

∴% Probability=0.5411*100=54.11

For stringS$_3$, Probability P_3= $\frac{f(x_1)}{\sum f(x_i)} = \frac{25}{1155}$ =0.0216,

∴% Probability=0.0216*100=2.16

For stringS$_4$, Probability P_4= $\frac{f(x_1)}{\sum f(x_i)} = \frac{361}{1155}$ =0.3126,

∴% Probability=0.3126*100=31.26

∴ $\sum p_i$ =$P_1+P_2+P_3+P_4$=0.1247+0.5411+0.0216+0.3126=1.0000

Average of$f(x_i) = \frac{\sum f(x_i)}{N}$ =$\frac{1155}{4}$=288.75, N=4=number of population,

Expected count =$\frac{Fitness}{Average}$

The expected count gives an idea of which population can be selected for further processing in the mating pool.

For stringS$_1$, Expected count =$\frac{Fitness}{Average}$=$\frac{f(x_1)}{288.75}$=$\frac{144}{288.75}$=0.4987

For stringS$_2$, Expected count =$\frac{Fitness}{Average}$=$\frac{f(x_2)}{288.75}$=$\frac{625}{288.75}$=2.1645

For stringS$_3$, Expected count =$\frac{Fitness}{Average}$=$\frac{f(x_3)}{288.75}$=$\frac{25}{288.75}$=0.0866

For stringS$_4$, Expected count =$\frac{Fitness}{Average}$=$\frac{f(x_4)}{288.75}$=$\frac{361}{288.75}$=1.2502

Sum of the expected count =0.4987+2.1645+0.0866+1.2502=4.0000

Thus we see from table-1,

For stringS$_1$, Probability of selection is 12.47%, Expected count=0.4987, so there is a chance for it to participate in the crossover cycle is at least once. Hence it actual count can be consider as 1.

For stringS$_2$, Probability of selection is 54.11%, Expected count=2.1645, so there is a fair chance for it to participate in the crossover cycle twice. Hence it actual count can be consider as 2.

For stringS$_3$, Probability of selection is 2.16%, Expected count=0.0866, so there is a very poor chance for it to participate in the crossover cycle. Hence it actual count can be consider as 0.

For stringS$_4$, Probability of selection is 31.26%, Expected count=1.2502, so there is a chance for it to participate in the crossover cycle is once. Hence it actual count can be consider as 1.

13.2 Table-2: The Presentation of the Crossover [5]

String no.	**Mating pool**	**Crossover point**	**Offspring after crossover**	**X value**	**fitness function f(x) = x^2**
S_1	01100	4	01101	13	169
S_2	11001	4	11000	24	576
S_3	11001	2	11011	27	729
S_4	10011	2	10001	17	289
Sum					1763
Average					440.75
Maximum					729

Explanation of table-2

The mating pool in table-2 is formed on the basis of actual count.

The actual count of stringS$_1$ is 1; hence stringS$_1$ occurs once in mating pool.

The actual count of stringS$_2$ is 2; hence stringS$_2$ occurs twice in mating pool.

The actual count of stringS$_3$ is 0; hence stringS$_3$does not occur in mating pool.

The actual count of stringS$_4$ is 1; hence stringS$_4$ occurs once in mating pool.

Now Crossover point is specified. On the basis of crossover point, a single- point crossover is performed and new offspring (children) is produced.

Thus a single- point crossover

Parent 1 0 1 1|0 0
Parent 2 1 1 0|0 1

Offspring 1 0 1 10 1
Offspring 2 1 1 0 00

And

Parent 1 1 1|0 0 1
Parent 2 1 0| 0 1 1

Offspring 1 1 1 01 1
Offspring 2 1 0 0 01

Hence after a single- point crossover new offspring (children) are produced. Now "x" values are decoded as follows

For string S_1 01101, the x value of $01101=0*2^4+1*2^3+1*2^2+0*2^1+1*2^0=0+8+4+0+1=13$

For string S_2 11000, the x value of $11000=1*2^4+1*2^3+0*2^2+0*2^1+0*2^0=16+8+0+0+0=24$

For string S_3 11011, the x value of $11011=1*2^4+1*2^3+0*2^2+1*2^1+1*2^0=16+8+0+2+1=27$

For string S_4 10001, the x value of $10001=1*2^4+0*2^3+0*2^2+0*2^1+1*2^0=16+0+0+0+1=17$

Now for fitness function $f(x) = x^2$, Calculate the fitness value,

For string S_1, x=13, $f(x) = x^2 = (13)^2 = 169$

For string S_2, x=24, $f(x) = x^2 = (24)^2 = 576$

For string S_3, x=27, $f(x) = x^2 = (27)^2 = 729$

For string S_4, x=17, $f(x) = x^2 = (17)^2 = 289$

Sum of the fitness value=$\sum f(x_i)$ =169+576+729+289=1763

Average of the fitness value=$\frac{\text{Sum of the fitness value}}{4}$ $=\frac{1763}{4}$ =440.75

13.3 Table-3: The Presentation of the Mutation [5]

String no.	**Offspring (children) after crossover**	**Mutation Chromosome for flipping**	**Offspring (children) after mutation**	**X value**	**Fitness function f(x) = x^2**
S_1	01101	10000	11101	29	841
S_2	11000	00000	11000	24	576
S_3	11011	00000	11011	27	729
S_4	10001	00100	10101	21	441
Sum					2587
Average					646.75
Maximum					841

Explanation of table-3

After crossover operation, mutation operation is performed and new offspring (children) are produced. I have discussed mutation flipping concept in section 12.Now mutation flipping operation is performed and offspring (children) are produced.

Parent	01101	11000	11011	10001
Mutation chromosome for flipping	10000	00000	00000	00100
New children(offspring)	11101	11000	11011	10101

Hence after mutation new offspring (children) are produced. Now "x" values are decoded as follows

For string S_1 11101, the x value of $11101=1*2^4+1*2^3+1*2^2+0*2^1+1*2^0=16+8+4+0+1=29$

For string S_2 11000, the x value of $11000=1*2^4+1*2^3+0*2^2+0*2^1+0*2^0=16+8+0+0+0=24$

For string S_3 11011, the x value of $11011=1*2^4+1*2^3+0*2^2+1*2^1+1*2^0=16+8+0+2+1=27$

For string S_4 10101, the x value of $10101=1*2^4+0*2^3+1*2^2+0*2^1+1*2^0=16+0+4+0+1=21$

Now for fitness function $f(x) = x^2$, Calculate the fitness value,

For string S_1, x=29, $f(x) = x^2 = (29)^2 = 841$

For string S_2, x=24, $f(x) = x^2 = (24)^2 = 576$

For string S_3, x=27, $f(x) = x^2 = (27)^2 = 729$

For string S_4, x=21, $f(x) = x^2 = (21)^2 = 441$

Sum of the fitness value=$\sum f(x_i)$ =841+576+729+441=2587

Average of the fitness value=$\frac{\text{Sum of the fitness value}}{4}$ $=\frac{2587}{4}$ =646.75

14. FOR THE FITNESS FUNCTION f(x) = x^2+1

14.1 Table-4:The Presentation of Selection

String no.	Initial population (randomly selected)	Value of the variable x	fitness function f(x) = x^2+1	Probability (p_i) of selection	% Probability of selection	Expected count = $\frac{Fitness}{Average}$	Actual count
S_1	01100	12	145=f (x_1)	0.1251 =P_1	12.51%	0.5004	1
S_2	11001	25	626= f(x_2)	0.5401 = P_2	54.01%	2.1604	2
S_3	00101	5	26= f(x_3)	0.0224 = P_3	2.24%	0.0897	0
S_4	10011	19	362= f(x_4)	0.3123 = P_4	31.23%	1.2494	1
Sum			1159= $\sum f(x_i)$	.9999≈ 1 = $\sum p_i$	99.99≈ 100	3.9989≈ 4	4
Average			289.75	0.2500	25	1.0000	1
Maximum			626	0.5401	54.01	2.1604	2

Calculations:

For string S_1 01100, the x value of $01100=0*2^4+1*2^3+1*2^2+0*2^1+0*2^0=0+8+4+0+0=12$

For string $S_2$11001, the x value of $11001=1*2^4+1*2^3+0*2^2+0*2^1+1*2^0=16+8+0+0+1=25$

For string S_3 00101, the x value of $00101=0*2^4+0*2^3+1*2^2+0*2^1+1*2^0=0+0+4+0+1=5$

For string $S_4$10011, the x value of $10011=1*2^4+0*2^3+0*2^2+1*2^1+1*2^0=16+0+0+2+1=19$

Now for fitness function $f(x) = x^2+1$,

Calculate the fitness value,

For string S_1, x=12, $f(x_1) = x^2+1 = (12)^2+1 = 144+1=145$

For string S_2, x=25, $f(x_2) = x^2+1 = (25)^2+1 = 625+1=626$

For string S_3, x=5, $f(x_3) = x^2+1 = (5)^2+1 = 25+1=26$

For string S_4,x=19, $f(x_4) = x^2+1 = (19)^2+1 = 361+1=362$

Sum of the fitness value=$\sum f(x_i)$ =145+626+26+362=1159

Probability of selection:

For string S_1, Probability $P_1 = \frac{f(x_1)}{\sum f(x_i)} = \frac{145}{1159}$ =0.1251, ∴% Probability= 0.1251*100=12.51

For string S_2, Probability $P_2 = \frac{f(x_1)}{\sum f(x_i)} = \frac{626}{1159}$ =0.5401, ∴% Probability=0.5401*100=54.01

For string S_3, Probability $P_3 = \frac{f(x_1)}{\sum f(x_i)} = \frac{26}{1159}$ =0.0224, ∴% Probability=0.0224*100=2.24

For string S_4, Probability $P_4 = \frac{f(x_1)}{\sum f(x_i)} = \frac{362}{1159}$ =0.3123, ∴% Probability=0.3123*100=31.23

∴ $\sum p_i = P_1+P_2+P_3+P_4$=0.1251+0.5401+0.0224+0.3123=.9999

Average of $f(x_i) = \frac{\sum f(x_i)}{N} = \frac{1159}{4}$=289.75, N=4=number of population,

Expected count $=\frac{Fitness}{Average}$

The expected count gives an idea of which population can be selected for further processing in the mating pool.

For string S_1, Expected count $=\frac{Fitness}{Average}=\frac{f(x_1)}{289.75}=\frac{145}{289.75}$=0.5004

For string S_2, Expected count $=\frac{Fitness}{Average}=\frac{f(x_2)}{289.75}=\frac{626}{289.75}$=2.1604

For string S_3, Expected count $=\frac{Fitness}{Average}=\frac{f(x_3)}{289.75}=\frac{26}{289.75}$=0.0897

For string S_4, Expected count $=\frac{Fitness}{Average}=\frac{f(x_4)}{289.75}=\frac{362}{289.75}$=1.2494

Sum of the expected count=0.5004+2.1604+0.0897+1.2484=3.9989≈ 4

Thus we see from table-4,

For stringS_1, Probability of selection is 12.51%, Expected count=0.5004, so there is a chance for it to participate in the crossover cycle is at least once. Hence it actual count can be consider as 1.

For stringS_2, Probability of selection is 54.01%, Expected count=2.1604, so there is a fair chance for it to participate in the crossover cycle twice. Hence it actual count can be consider as 2.

For stringS_3, Probability of selection is 2.24%, Expected count=0.0897, so there is a very poor chance for it to participate in the crossover cycle. Hence it actual count can be consider as 0.

For stringS_4, Probability of selection is 31.23%, Expected count=1.2494, so there is a chance for it to participate in the crossover cycle is once. Hence it actual count can be consider as 1.

14.2 Table-5: The Presentation of the Crossover

String no.	Mating pool	Crossover point	Offspring after crossover	X value	fitness function f(x) = x^2+1
S_1	01100	4	01101	13	170
S_2	11001	4	11000	24	577
S_3	11001	2	11011	27	730
S_4	10011	2	10001	17	290
Sum					1767
Average					441.75
Maximum					730

Explanation of table-5

The mating pool in table-5 is formed on the basis of actual count.

The actual count of stringS_1 is 1; hence stringS_1 occurs once in mating pool.

The actual count of stringS_2 is 2; hence stringS_2 occurs twice in mating pool.

The actual count of stringS_3 is 0; hence stringS_3 does not occur in mating pool.

The actual count of stringS_4 is 1; hence stringS_4 occurs once in mating pool.

Now Crossover point is specified. On the basis of crossover point, a single- point crossover is performed and new offspring (children) is produced.

Thus a single- point crossover

Parent 1 0 1 1|0 0
Parent 2 1 1 0|0 1

Offspring 1 0 1 10 1
Offspring2 1 1 0 00

And

Parent 1 1 1|0 0 1
Parent 2 1 0| 0 1 1

Offspring11 1 01 1
Offspring 21 0 0 01

Hence after a single- point crossover new offspring (children) are produced. Now "x" values are decoded as follows

For string S_1 01101, the x value of $01101=0*2^4+1*2^3+1*2^2+0*2^1+1*2^0$=0+8+4+0+1=13

For string S_2 11000, the x value of $11000=1*2^4+1*2^3+0*2^2+0*2^1+0*2^0$=16+8+0+0+0=24

For string S_3 11011, the x value of $11011=1*2^4+1*2^3+0*2^2+1*2^1+1*2^0$=16+8+0+2+1=27

For string S_4 10001, the x value of $10001=1*2^4+0*2^3+0*2^2+0*2^1+1*2^0$=16+0+0+0+1=17

Now for fitness function $f(x) = x^2+1$,

Calculate the fitness value,

For string S_1, x=13,$f(x) = x^2+1 = (13)^2+1$ =170

For string S_2, x=24,f(x) = x^2+1 = $(24)^2$+1 = 577

For string S_3, x=27, f(x) = x^2+1 = $(27)^2$+1 =730

For string S_4, x=17, f(x) = x^2+1 = $(17)^2$+1 =290

Sum of the fitness value= $\sum f(x_i)$ =170+577+730+290=1767

Average of the fitness value=$\frac{\text{Sum of the fitness value}}{4}$ =$\frac{1767}{4}$ =441.75

14.3 Table-6: The Presentation of the Mutation

String no.	Offspring (children) after crossover	Mutation Chromosome for flipping	Offspring (children) after mutation	X value	Fitness function f(x) = x^2+1
S_1	01101	10000	11101	29	842
S_2	11000	00000	11000	24	577
S_3	11011	00000	11011	27	730
S_4	10001	00100	10101	21	442
Sum					2591
Average					647.75
Maximum					842

Explanation of table-6

After crossover operation, mutation operation is performed and new offspring (children) are produced. I have discussed mutation flipping concept in section 12.Nowmutation flipping operation is performed and offspring (children) are produced.

Parent	01101 11000 11011 10001
Mutation chromosome for flipping	10000 00000 00000 00100
New children(offspring)	11101 11000 11011 10101

Hence after mutation new offspring (children) are produced. Now “x” values are decoded as follows

For string S_1 11101, the x value of11101=1*2^4+1*2^3+1*2^2+0*2^1+1*2^0=16+8+4+0+1=29

For string S_2 11000, the x value of 11000=1*2^4+1*2^3+0*2^2+0*2^1+0*2^0=16+8+0+0+0=24

For string S_3 11011, the x value of11011=1*2^4+1*2^3+0*2^2+1*2^1+1*2^0=16+8+0+2+1=27

For string S_4 10101, the x value of10101=1*2^4+0*2^3+1*2^2+0*2^1+1*2^0=16+0+4+0+1=21

Now for fitness function f(x) = x^2+1,

Calculate the fitness value,

For string S_1, x=29,f(x) = x^2+1 = $(29)^2$+1 =842

For string S_2, x=24,f(x) = x^2+1= $(24)^2$+1=577

For string S_3, x=27, f(x) = x^2+1= $(27)^2$+1 =730

For string S_4, x=21,f(x) = x^2+1 = $(21)^2$+1 =442

Sum of the fitness value=$\sum f(x_i)$ =842+577+730+442=2591

Average of the fitness value=$\frac{\text{Sum of the fitness value}}{4}$ =$\frac{2591}{4}$=647.75

15. CONCLUSION

Once selection, crossover and mutation are performed the new population is now ready to be tested. In table-1 the expected count gives an idea of which population can be selected for further processing in the mating pool. The actual count gives an idea to select the individuals who would participate in the crossover cycle.

For the fitness function f(x) = x^2, we have,

From table-1, table-2, table-3 we observed that how maximum fitness and the population average fitness performances have improved in the new population. In one generation, the population average fitness has improved from 288.75to 646.75, this improved by≈ 124%.During the same period the maximum fitness has improved 625 to 841,this improved by≈ 35%

We also observe that the total fitness has gone from 1155 to 2587this improved by≈ 124%in a single generation.The best string from initial population (randomly selected) 01100,11001,00101,10011 is11001, it receives two chances for its existence because of its high, above-average performances. Thus after mutation (table-3) a new Offspring (11101) is produced which is an excellent choice.

For the fitness function f(x) = x^2+1, we have,

From table-4, table-5, table-6 we observed that how maximum fitness and the population average fitness performances have improved in the new population. In one generation, the population average fitness has improved from 289.75to 647.75this improved by≈ 124%.During the same period the maximum fitness has improved 626 to 842, this improved by≈ 35%.

We also observe that the total fitness has gone from 1159 to 2591this improved by≈ 124% in a single generation.

The best string from initial population (randomly selected) 01100,11001,00101,10011 is11001, it receives two chances for its existence because of its high, above-average performances.

Thus after mutation (table-6) a new Offspring (11101) is produced which is an excellent choice.

Thus we conclude that for two different fitness function f(x) = x^2 and f(x) = x^2+1, the best string from initial population (randomly selected) 01100, 11001, 00101, 10011 is11001and after mutation (table-3& table-6) a new Offspring (11101) is produced which is an excellent choice, i.e. the best solution does not change for two different fitness function f(x) = x^2 and f(x) = x^2+1.This completes the genetic algorithm is performed in one generation.

16. ACKNOWLEDGEMENTS

Author of this paper sincerely thank to Prof. Kumar Sankar Ray, Dept of Electronics & Communication Science Unit, Indian Statistical Institute, Kolkata- 700108 India, for his constant support and encouragement.

17. REFERENCES

[1] Genetic algorithms for optimization – application in the controller synthesis task – Popov A., diploma thesis, department Systems and Control, faculty Automatics, Technical University Sofia, 2003

[2] “Engineering design optimization with genetic algorithm” by Richard H.Dinger.

[3] “Fundamentals of genetic algorithms, Artificial Intelligence” by RC chakraborty.

[4] GENETIC ALGORITHMS, Chapter 4, Kumara Sastry, David Goldberg

[5] Lecture 2: Canonical Genetic Algorithm, Suggested reading: D. E. Goldberg, Genetic Algorithm in Search, Optimization, and Machine Learning, Addison Wesley Publishing Company, January 1989

[6] Lecture 9: Evolutionary Computation and Genetic Algorithms Part-II. Alan Holland, aholland@4c.ucc.ie University College Cork Mar 2010, Universitat de Girona: Intelligent Systems

[7] The Genetic Algorithm for finding the maxima of single variable functions By DezdemonaGjylapi& Vladimir Kasëmi International Journal of Engineering and Science. Vol.4, Issue 3(March 2014), PP 46-54

[8] SAKAWA, MASATOSHI.Genetic Algorithms and Fuzzy MultiobjectiveOptimization (Kluwer Academic Publishers, 2002). pp. 12-26. ISBN 0-7923-7452-5

[9] Goldberg, D. E. and Voessner, S., 1999, Optimizing global-local search Hybrids, in: Proc. of the Genetic and Evolutionary Computation Conf., pp. 220–228.

[10] Holland, J.,"Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence." The University of Michigan Press, 1975.

[11] J. Holland, Adaptation in natural and artificial systems, University of Michigam Press, Ann Arbor, 1975.

[12] Manju Sharma, Novel Knowledge based Selective Tabu Initialization in Genetic algorithm, IJARCSSE, Volume 3, Issue 5, May 2013

[13] K. Deb and S. Argrawal, "Understanding interactions among genetic algorithm parameters,"in Foundations of Genetic Algorithms 5, 1998.

[14] D. E. Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning,Addison Wesley, 1989.

[15] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionaryalgorithms,"IEEE Transactions on Evolutionary Computation, Vol. 3, 1999,

18. AUTHOR PROFILE

Dipanjan Kumar Dey, graduated from Calcutta University, India. M.sc (Mathematics) and M.Tech (Computer Science &Engineering) from M.C.K.V Institute of Engineering (under West Bengal University & Technology, India). He is currently Assistant Professor of Mathematics & Computer Science in Prajnanananda Institute of Technology & Management, West Bengal, India. He is also Faculty member of Institute of Chartered financial Analysis of India (ICFAI) and Academic Counselor, Assistant Coordinator of Indira Gandhi National Open University (IGNOU)study center 2804, Kolkata. He is a Science Journalist having Post Graduate certificate on science journalism and media practice from National Council for the Science and Technology Communication, Govt. of India, New Delhi.

Mr. Dey has to his credit a significant number of research papers published in international journals of repute.

His research interests in Genetic Algorithms, Soft Computing, Fuzzy Set, Artificial intelligence, Mobile Computing.