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 it is to have some means or procedure for discriminating good solutions from bad solutions. The fitness function \( f(x) = x^2 + n \), where \( n=0, \pm 1, \pm 2, \pm 3, \ldots \) 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 31 for the string 11111.

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.

<table>
<thead>
<tr>
<th>Population</th>
<th>Chromosome 1</th>
<th>0 1 1 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Chromosome 2</td>
<td>1 1 0 0 1</td>
</tr>
<tr>
<td></td>
<td>Chromosome 3</td>
<td>0 0 1 0 1</td>
</tr>
<tr>
<td></td>
<td>Chromosome 4</td>
<td>1 0 0 1 1</td>
</tr>
</tbody>
</table>

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 1, mostly used [3], like

Chromosome 1: 1 0 1 1 0 0 1 0 1 1 0 0 1 0 1 0 1 1 0 1 1 0 1 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 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 member of the population: 
   \[ P_i = \frac{f_i}{\sum f_j} \]
   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, \( r \), \( 0 \leq r \leq q_i \)
5. If \( r < q_i \), then select the first chromosome, \( x_i \), else select the individual, \( x_j \), such that \( q_{j-1} \leq r < q_j \)
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 (\( n = 4 \)), with the fitness function \( f(x) = x^2 \), fitness values as shown in the table below. [4]

<table>
<thead>
<tr>
<th>String no.</th>
<th>Initial population (randomly selected)</th>
<th>Value of the variable ( x )</th>
<th>Fitness function ( f(x) = x^2 )</th>
<th>Probability ( p_i ) of selection</th>
<th>Cumulative probability ( q_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_i )</td>
<td>01100</td>
<td>12</td>
<td>144 = ( f(x) )</td>
<td>( \frac{144}{277} ) ( P_i ) = 0.5141</td>
<td>0.0216 = ( q_1 )</td>
</tr>
<tr>
<td>( S_i )</td>
<td>11001</td>
<td>25</td>
<td>625 = ( f(x) )</td>
<td>( \frac{625}{277} ) ( P_i ) = 0.5411</td>
<td>0.6658 = ( q_1 )</td>
</tr>
<tr>
<td>( S_i )</td>
<td>00101</td>
<td>5</td>
<td>25 = ( f(x) )</td>
<td>( \frac{25}{277} ) ( P_i ) = 0.0216</td>
<td>0.6874 = ( q_1 )</td>
</tr>
<tr>
<td>( S_i )</td>
<td>10011</td>
<td>19</td>
<td>361 = ( f(x) )</td>
<td>( \frac{361}{277} ) ( P_i ) = 0.3126</td>
<td>1.0000 = ( q_1 )</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td></td>
<td>1155 = ( \sum f(x) )</td>
<td>1.0000 = ( \Sigma P_i )</td>
<td></td>
</tr>
</tbody>
</table>

Now if we generate a random number, say 0.6712, \( q_3 = 0.6658 < 0.6712 < q_4 = 0.6874 \)

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]

<table>
<thead>
<tr>
<th>String no.</th>
<th>Initial population (randomly selected)</th>
<th>Value of the variable ( x )</th>
<th>Fitness function ( f(x) = x^2 + 1 )</th>
<th>Probability ( p_i ) of selection</th>
<th>Cumulative probability ( q_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_i )</td>
<td>01100</td>
<td>12</td>
<td>146 = ( f(x) )</td>
<td>( \frac{146}{277} ) ( P_i ) = 0.5141</td>
<td>0.0216 = ( q_1 )</td>
</tr>
<tr>
<td>( S_i )</td>
<td>11001</td>
<td>25</td>
<td>626 = ( f(x) )</td>
<td>( \frac{626}{277} ) ( P_i ) = 0.5411</td>
<td>0.6658 = ( q_1 )</td>
</tr>
<tr>
<td>( S_i )</td>
<td>00101</td>
<td>5</td>
<td>26 = ( f(x) )</td>
<td>( \frac{26}{277} ) ( P_i ) = 0.0216</td>
<td>0.6874 = ( q_1 )</td>
</tr>
<tr>
<td>( S_i )</td>
<td>10011</td>
<td>19</td>
<td>362 = ( f(x) )</td>
<td>( \frac{362}{277} ) ( P_i ) = 0.3126</td>
<td>1.0000 = ( q_1 )</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td></td>
<td>1158 = ( \sum f(x) )</td>
<td>1.0000 = ( \Sigma P_i )</td>
<td></td>
</tr>
</tbody>
</table>
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 (i 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 1 1 1 0 1 1 | 0 0 1 0 0 1 1 0 1 1 0
Parent 2 1 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 1 1 1 0 1 1 | 1 1 0 0 0 0 1 1 1 1 0
Offspring 2 1 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 | 1100111100011110 |
| Mutated offspring 2 | 11011001100110110 |

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

Crossover probability (Pc) 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 (Pc) is 100%, then all offspring is made by crossover.

If crossover probability (Pc) 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 (Pm)
Mutation probability (Pm) 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 (Pm) is 100%, whole chromosome is changed, if (Pm) 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 to 1 and 1 to 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.

<table>
<thead>
<tr>
<th>Parent</th>
<th>Mutation chromosome</th>
<th>Child</th>
</tr>
</thead>
<tbody>
<tr>
<td>1011</td>
<td>1000</td>
<td>0011</td>
</tr>
</tbody>
</table>

Figure-Mutation flipping concept

13. SOLVING MAXIMIZING PROBLEM FOR 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_R=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 minimum 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_R=R, since f(x) = x^2+1 defined for all real values of x, R= real numbers. Since x^2+1 ≥ 1, ∀ x ∈ R. ∀ x ∈ R, f(x) ≥ 1, x ∈ R. Thus, range of f(x) = [1,∞), i.e. R_f = [1,∞). The function f(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 × n = 4 × 5 = 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.
For the fitness function $f(x) = x^2$

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

<table>
<thead>
<tr>
<th>String no.</th>
<th>Initial population (randomly selected)</th>
<th>Value of the variable $x$</th>
<th>fitness function $f(x) = x^2$</th>
<th>Probability of selection ($p$)</th>
<th>% Probability of selection</th>
<th>Expected count $= \frac{Fitness}{Average}$</th>
<th>Actual count</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>01100</td>
<td>12</td>
<td>144(=f(x))</td>
<td>$p = \frac{0.1247}{54.11%}$</td>
<td>12.47%</td>
<td>0.498 1</td>
<td></td>
</tr>
<tr>
<td>$S_2$</td>
<td>11001</td>
<td>25</td>
<td>625(=f(x))</td>
<td>$p = \frac{0.5411}{54.11%}$</td>
<td>54.11%</td>
<td>2.164 2</td>
<td></td>
</tr>
<tr>
<td>$S_3$</td>
<td>00101</td>
<td>5</td>
<td>25(=f(x))</td>
<td>$p = \frac{0.0216}{2.16%}$</td>
<td>2.16%</td>
<td>0.086 6</td>
<td></td>
</tr>
<tr>
<td>$S_4$</td>
<td>10011</td>
<td>19</td>
<td>361(=f(x))</td>
<td>$p = \frac{0.3126}{31.26%}$</td>
<td>31.26%</td>
<td>1.250 1</td>
<td></td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td></td>
<td>$\sum f(x)$</td>
<td>$p = \frac{1.0000}{4.0000}$</td>
<td>25%</td>
<td>1.000 4</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td>$\frac{288.75}{4}$</td>
<td></td>
<td>25</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Calculations:
For string $S_1$, 01100, the $x$ value of $01100=0+2^4+1+2^2+1+2^2+0^2+0^2+2^2=14+8+4+0+0+1=12$

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

For string $S_3$, 00101, the $x$ value of $00101=0+2^4+0+2^2+1+2^2+1+2^2=0+0+4+0+4+1=15$

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

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

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 $= \frac{Fitness}{Average} = \frac{144}{288.75} = 0.4987$

For string $S_2$, Expected count $= \frac{Fitness}{Average} = \frac{625}{288.75} = 2.1645$

For string $S_3$, Expected count $= \frac{Fitness}{Average} = \frac{25}{288.75} = 0.0866$

For string $S_4$, Expected count $= \frac{Fitness}{Average} = \frac{361}{288.75} = 1.2502$

The actual count of $S_1$ is 1; hence string $S_1$ occurs once in mating pool.

Thus we see from table-1,

For string $S_2$, 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 considered as 1.

For string $S_3$, 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 considered as 2.

For string $S_4$, 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 considered as 0.

For string $S_4$, 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 considered as 1.

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

<table>
<thead>
<tr>
<th>String no.</th>
<th>Mating pool</th>
<th>Crossover point</th>
<th>Offspring after crossover</th>
<th>X value</th>
<th>Fitness function $f(x) = x^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>01100</td>
<td>1</td>
<td>11010</td>
<td>11</td>
<td>144</td>
</tr>
<tr>
<td>$S_2$</td>
<td>11001</td>
<td>4</td>
<td>11001</td>
<td>11</td>
<td>625</td>
</tr>
<tr>
<td>$S_3$</td>
<td>00101</td>
<td>5</td>
<td>01100</td>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td>$S_4$</td>
<td>10011</td>
<td>19</td>
<td>10011</td>
<td>12</td>
<td>361</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td></td>
<td></td>
<td>79</td>
<td>1729</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td>24.64</td>
<td>440.75</td>
</tr>
<tr>
<td>Maximum</td>
<td></td>
<td></td>
<td></td>
<td>79</td>
<td>729</td>
</tr>
</tbody>
</table>

Explanation of table-2

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

The actual count of string $S_1$ is 1; hence string $S_1$ occurs once in mating pool.

The actual count of string $S_2$ is 2; hence string $S_2$ occurs twice in mating pool.

The actual count of string $S_3$ is 0; hence string $S_3$ does not occur in mating pool.

The actual count of string $S_4$ is 1; hence string $S_4$ 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
Hence after mutation new offspring (children) are produced. Now “x” values are decoded as follows

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 values $\sum f(x_i) = 169+576+729+289 = 1763$
Average of the fitness value $= \frac{\sum \text{of the fitness value}}{4} = \frac{1763}{4} = 440.75$

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

<table>
<thead>
<tr>
<th>String no.</th>
<th>Offspring (children) after crossover</th>
<th>Mutation Chromosome for flipping</th>
<th>Offspring (children) after mutation</th>
<th>$X$ value</th>
<th>Fitness function $f(x) = x^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>10101</td>
<td>10000</td>
<td>11101</td>
<td>29</td>
<td>841</td>
</tr>
<tr>
<td>$S_2$</td>
<td>11000</td>
<td>00000</td>
<td>11000</td>
<td>24</td>
<td>576</td>
</tr>
<tr>
<td>$S_3$</td>
<td>11011</td>
<td>00000</td>
<td>11101</td>
<td>27</td>
<td>729</td>
</tr>
<tr>
<td>$S_4$</td>
<td>10001</td>
<td>00100</td>
<td>10101</td>
<td>21</td>
<td>441</td>
</tr>
<tr>
<td><strong>Sum</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>2587</strong></td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>646.75</strong></td>
</tr>
<tr>
<td><strong>Maximum</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>841</strong></td>
</tr>
</tbody>
</table>

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.

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

For string $S_1$, $x=11101$, the $X$ value of $11101=1*2^4+1*2^3+1*2^2+1*2^1+1*2^0=16+8+4+2+1=29$
For string $S_2$, $x=11000$, the $X$ value of $11000=1*2^4+1*2^3+0*2^2+0*2^1+0*2^0=16+8+4+2+0=24$
For string $S_3$, $x=11011$, the $X$ value of $11011=1*2^4+1*2^3+0*2^2+1*2^1+1*2^0=16+8+4+2+1=27$
For string $S_4$, $x=10101$, the $X$ value of $10101=1*2^4+0*2^3+1*2^2+0*2^1+1*2^0=16+0+4+2+1=21$

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 values $\sum f(x_i) = 841+576+729+441 = 2587$
Average of the fitness value $= \frac{\sum \text{of the fitness values}}{4} = \frac{2587}{4} = 646.75$

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

14.1 Table-4: The Presentation of Selection

<table>
<thead>
<tr>
<th>String no.</th>
<th>Initial population (randomly selected)</th>
<th>Value of the variable $x$</th>
<th>Fitness function $f(x) = x^2 + 1$</th>
<th>Probability $(p_i)$ of selection</th>
<th>Expected count as fitness average</th>
<th>Actual count</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>01100</td>
<td>12</td>
<td>$145 = f(x_1)$</td>
<td>$0.1251 = p_1$</td>
<td>0.500</td>
<td>1</td>
</tr>
<tr>
<td>$S_2$</td>
<td>11001</td>
<td>25</td>
<td>$626 = f(x_2)$</td>
<td>$0.5401 = p_2$</td>
<td>2.160</td>
<td>2</td>
</tr>
<tr>
<td>$S_3$</td>
<td>00101</td>
<td>5</td>
<td>$26 = f(x_3)$</td>
<td>$0.0224 = p_3$</td>
<td>0.089</td>
<td>0</td>
</tr>
<tr>
<td>$S_4$</td>
<td>10011</td>
<td>19</td>
<td>$362 = f(x_4)$</td>
<td>$0.3123 = p_4$</td>
<td>1.249</td>
<td>1</td>
</tr>
<tr>
<td><strong>Sum</strong></td>
<td></td>
<td></td>
<td>$1199 = \sum f(x_i)$</td>
<td>$99.99 =\frac{\sum p_i}{\frac{1}{\Sigma}}$</td>
<td><strong>9.998</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td></td>
<td>$289.75 = \frac{\Sigma f(x_i)}{5}$</td>
<td>$25 = \frac{\sum \text{of the fitness values}}{5}$</td>
<td><strong>1.000</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Maximum</strong></td>
<td></td>
<td></td>
<td>$626 = \frac{\sum f(x_i)}{5}$</td>
<td><strong>2.160</strong></td>
<td><strong>2</strong></td>
<td></td>
</tr>
</tbody>
</table>

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 $10011=1*2^4+0*2^3+0*2^2+1*2^1+2^0=16+0+0+2+1=19$

Now for fitness function $f(x) = x^2 + 1$,
Calculate the fitness value,
For string $S_1$, $f(x_1) = x^2 + 1 = (12)^2 + 1 = 144 + 1 = 145$
For string $S_2$, $f(x_2) = x^2 + 1 = (25)^2 + 1 = 625 + 1 = 626$
For string $S_3$, $f(x_3) = x^2 + 1 = (25)^2 + 1 = 25 + 1 = 26$
For string $S_4$, $f(x_4) = x^2 + 1 = (19)^2 + 1 = 361 + 1 = 362$
Sum of the fitness value $= \sum f(x_i) = 145+626+26+362 = 1159$

**Probability of selection:**

For string $S_1$, Probability $P_i = \frac{f(x_i)}{\sum f(x_i)} = \frac{145}{1159} = 0.1251$.
Probability $= 0.1251 \times 100 = 12.51$

For string $S_2$, Probability $P_i = \frac{f(x_i)}{\sum f(x_i)} = \frac{626}{1159} = 0.5401$.
Probability $= 0.5401 \times 100 = 54.01$

For string $S_3$, Probability $P_i = \frac{f(x_i)}{\sum f(x_i)} = \frac{26}{1159} = 0.0224$.
Probability $= 0.0224 \times 100 = 2.24$

For string $S_4$, Probability $P_i = \frac{f(x_i)}{\sum f(x_i)} = \frac{362}{1159} = 0.3123$.
Probability $= 0.3123 \times 100 = 31.23$

Thus we see from table-4,

For string $S_1$, Probability of selection is 12.51%, Expected count $= 0.1251 \times 100 = 12.51$.

For string $S_2$, Probability of selection is 54.01%, Expected count $= 0.5401 \times 100 = 54.01$.

For string $S_3$, Probability of selection is 2.24%, Expected count $= 0.0224 \times 100 = 2.24$.

For string $S_4$, Probability of selection is 31.23%, Expected count $= 0.3123 \times 100 = 31.23$.

**4.2 Table-5: The Presentation of the Crossover**

<table>
<thead>
<tr>
<th>String no.</th>
<th>Matting pool</th>
<th>Crossover point</th>
<th>Offspring after crossover</th>
<th>X value</th>
<th>fitness function $f(x) = x^3 + 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>01100</td>
<td>4</td>
<td>01101</td>
<td>13</td>
<td>120</td>
</tr>
<tr>
<td>$S_2$</td>
<td>11001</td>
<td>4</td>
<td>11000</td>
<td>24</td>
<td>577</td>
</tr>
<tr>
<td>$S_3$</td>
<td>11001</td>
<td>2</td>
<td>11011</td>
<td>27</td>
<td>730</td>
</tr>
<tr>
<td>$S_4$</td>
<td>10011</td>
<td>2</td>
<td>10001</td>
<td>17</td>
<td>290</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td></td>
<td></td>
<td>730</td>
<td></td>
</tr>
</tbody>
</table>

**Maximum**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>730</td>
</tr>
</tbody>
</table>

**Explanation of table-5**

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

The actual count of string $S_1$ is 1; hence string $S_1$ occurs once in matting pool.

The actual count of string $S_2$ is 2; hence string $S_2$ occurs twice in matting pool.

The actual count of string $S_3$ is 0; hence string $S_3$ does not occur in matting pool.

The actual count of string $S_4$ is 1; hence string $S_4$ occurs once in matting 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 1 0 1
Offspring 2 1 1 0 0 0

And

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

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

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

For string $S_1$, x value of 01101 = $0^2 + 1^2 + 1^2 = 1 + 1 + 1 = 3$

For string $S_2$, x value of 10001 = $1^2 + 0^2 + 0^2 + 0^2 + 1 + 2 = 4 + 1 = 5$

For string $S_3$, x value of 11011 = $1^2 + 1^2 + 1^2 + 1^2 + 1 + 2 = 1 + 1 + 1 + 1 + 1 + 1 = 8$

For string $S_4$, x value of 11001 = $1^2 + 1^2 + 1^2 + 0^2 = 1 + 1 + 1 = 3$

Now for fitness function $f(x) = x^3 + 1$,

Calculate the fitness value,

For string $S_1$, $f(x) = x^3 + 1 = (13)^3 + 1 = 170$
For string $S_1$, $x = 24$, $f(x) = x^3 + 1 = (24)^3 + 1 = 577$
For string $S_2$, $x = 27$, $f(x) = x^3 + 1 = (27)^3 + 1 = 730$
For string $S_3$, $x = 17$, $f(x) = x^3 + 1 = (17)^3 + 1 = 290$

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

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

### 14.3 Table-6: The Presentation of the Mutation

<table>
<thead>
<tr>
<th>String no.</th>
<th>Offspring (children) after crossover</th>
<th>Mutation Chromosome for Flipping</th>
<th>Offspring (children) after mutation</th>
<th>X value</th>
<th>Fitness function $f(x) = x^3 + 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_1</td>
<td>01101</td>
<td>00000</td>
<td>11101</td>
<td>29</td>
<td>842</td>
</tr>
<tr>
<td>S_2</td>
<td>11000</td>
<td>00000</td>
<td>11000</td>
<td>24</td>
<td>577</td>
</tr>
<tr>
<td>S_3</td>
<td>11001</td>
<td>00100</td>
<td>10101</td>
<td>27</td>
<td>730</td>
</tr>
<tr>
<td>S_4</td>
<td>10001</td>
<td>01100</td>
<td>10101</td>
<td>21</td>
<td>442</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2591</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>647.75</td>
</tr>
<tr>
<td>Maximum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>842</td>
</tr>
</tbody>
</table>

**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. Now mutation flipping operation is performed and offspring (children) are produced.

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 = $1^2 + 2^2 + 2^2 + 0^2 + 2^2 = 16 + 4 + 4 + 1 = 29$
For string $S_2$, 11000, the x value of 11000 = $1^2 + 2^2 + 2^2 + 0^2 + 2^2 = 16 + 4 + 4 + 0 + 0 = 24$
For string $S_3$, 11011, the x value of 11011 = $1^2 + 2^2 + 2^2 + 0^2 + 2^2 + 1^2 = 16 + 4 + 4 + 0 + 0 + 1 = 27$
For string $S_4$, 10101, the x value of 10101 = $1^2 + 2^2 + 2^2 + 0^2 + 2^2 + 1^2 = 16 + 4 + 4 + 0 + 0 + 1 = 21$

Now for fitness function $f(x) = x^3 + 1$.

Calculate the fitness value,

For string $S_1$, $x = 29$, $f(x) = x^3 + 1 = (29)^3 + 1 = 842$
For string $S_2$, $x = 24$, $f(x) = x^3 + 1 = (24)^3 + 1 = 577$
For string $S_3$, $x = 27$, $f(x) = x^3 + 1 = (27)^3 + 1 = 730$
For string $S_4$, $x = 21$, $f(x) = x^3 + 1 = (21)^3 + 1 = 442$

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

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

### 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 $f(x) = x^3$, 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 288.75 to 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 2587 this improved by $124\%$ in a single generation. The best string from initial population (randomly selected) 01100, 11001, 00101, 10011 is 11001, 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 $f(x) = x^3 + 1$, 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 289.75 to 647.75, this improved by $124\%$. During the same period the maximum fitness has improved 626 to 841, this improved by $35\%$.

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

The best string from initial population (randomly selected) 01100, 11001, 00101, 10011 is 11001, 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 $f(x) = x^3$ and $f(x) = x^3 + 1$, the best string from initial population (randomly selected) 01100, 11001, 00101, 10011 is 11001 and 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 $f(x) = x^3$ and $f(x) = x^3 + 1$. 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


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


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 Prajanananda 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.

IJCA™: www.ijcaonline.org