INTRODUCTION
This paper aims to apply Genetic Algorithm as a nonlinear technique in solving linear and nonlinear equation system and to investigate the major benefits obtained as a result of using GA. Normally, GAs are simulation programs which create an environment where populations of data can compute and only the fittest survive, sort of evolution on a computer. GAs are excellent for all tasks requiring optimization and highly effective in any situation where many inputs (variables) interact to produce a large number of possible outputs (solutions). Some example situations are:

1. Optimization:
   Such as data fitting, clustering, trend spotting, path finding and ordering.
2. Management:
   Distribution, scheduling, project management, courier routing, container packing, task assignment and timetables.
3. Financial:
   Portfolio balancing, budgeting, forecasting, investment analysis and payment scheduling.
4. Engineering:
   Structural design (e.g. beam sizes), electrical design (e.g. circuit boards), mechanical design (e.g. optimize weight, size & cost), process control, network design (e.g. computer networks).

A genetic algorithm is a problem solving method that is inspired by its genetics as its model of problem solving. A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. GA handles a variety of possible solutions. Individual solution is represented through a chromosome, which is just a summarization of representation. To find the best solution, it is necessary to perform certain operations of (on) each optimum solution. The process starts by generating an initial population of chromosomes. This first population must offer a wide diversity of genetic materials. The gene pool should be as large as possible so that any solution of the search space can be propagated. Then, the GA loops over an iteration process to make the population evolve. Each iteration consists of selection, crossover, mutation and replacement.

Selection
Selection is the process of choosing two parents from the population for crossing. After deciding on an encoding, the next step is to decide how to perform selection. Darwin stated in his theory of evolution that best ones survive to create new offspring. Selection is a method that randomly picks chromosomes out of the population according to their evaluation function. The higher the fitness function, the more chance an individual will be selected. Some of the selection methods are:

1. Roulette Wheel Selection:
   Roulette wheel selection is one of the traditional GA selection techniques. Roulette wheel selection is easier to implement but is noisy. The rate of evolution depends on the variance of fitness’s in the population.

2. Tournament Selection:
   To fine-tune (Fine-tuning) GA search performances depends upon adoptable selective pressure and population diversity, which is an ideal procedure for selection. A tournament competition among individuals gives a tournament strategy for selective pressure.

3. Elitism:
   The first best chromosome or the few best chromosomes are copied to the new population. The rest is done in a classical way. Such individuals can be lost if they are not selected to reproduce or if crossover or mutation destroys them. This significantly improves the GA’s performance.

Crossover (Recombination)
Crossover is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). 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.

Mutation
After crossover, the strings are subjected to mutation. Mutation is performed to one individual to produce a new version of it, where some of the original genetic material has been randomly changed. Mutation prevents the algorithm to be trapped in a local minimum. Mutation plays the role of recovering the lost genetic materials as well as for randomly disturbing genetic information.
It is an insurance policy against the irreversible loss of genetic material. Mutation has been traditionally considered as a simple search operator. If crossover is supposed to exploit the current solution to find better ones, mutation is supposed to help for the exploration of the whole search space. Mutation is viewed as a background operator to maintain genetic diversity in the population. It introduces new genetic structures in the population by randomly modifying some of its building blocks. Mutation helps escape from local minima’s trapped and maintains diversity in the population.

**RELATED WORKS**

GA for machining fixture locating and clamping position optimization. In this work, the application of genetic algorithms (GAs) to the fixture layout optimization is presented to handle fixture layout optimization problem. Fixtures are used to locate and constrain a work piece during a machining operation, minimizing work piece and fixture tooling deflections due to clamping and cutting forces are critical to ensuring accuracy of the machining operation. Conventionally, fixtures machining were designed and manufactured through trial-and-error, which is more costly and time consuming to the manufacturing process. But it proves that the fixture layout optimization problems are multi modal problems. Optimized designs do not have any apparent similarities although they provide very similar performance. It observed that fixture layout problems are multi modal, therefore it investigate rules for fixture design should be used in GA to select best design among others [1][10].

GA for computer aided process planning in a distributed manufacturing environment, factories possessing various machines and tools at different geographical locations are often combined to achieve the highest production efficiency. When jobs requiring several operations are received, feasible plans may vary due to different resource constraints. Therefore, obtaining an optimal or near-optimal process plan becomes important. Their work presents a genetic algorithm (GA) which, according to prescribed criteria such as minimizing processing time, could swiftly search for the optimal process plan for a single manufacturing system as well as distributed manufacturing systems. By applying the GA, the computer-aided process planning (CAPP) system can generate desirable or near-optimal process plans based on the chosen criteria. In a distributed manufacturing environment, factories possessing various machines and tools are at different geographical locations and different manufacturing capabilities are often selected to achieve the highest production efficiency. When jobs requiring several operations are received, available factories according to the precedence relationships of those operations produce feasible process plans. Manufacturing operations can be performed by different machines and tools located at different locations. The final desirable or near-optimal process plan will emerge after comparison of all the feasible process plans. When dealing with a distributed manufacturing system, a chromosome not only represents the sequence of the operations but also indicates which factory this process plan comes from[1][10]. Therefore, the identity number of the factory will be placed as the first gene of each chromosome no matter how the other genes are randomly arranged. Every other gene comprises operation ID and corresponding machine, tool and tool access direction (TAD), which will be used to accomplish this operation. As a result, a process plan will be represented by a random combination of genes. Fig.2 shows the representation of six-operation process plans. Evaluations in that work contain two criteria; a criterion 1 is the minimum production cost and criteria 2 is the minimum processing time.

Total production cost (Pc) equation is

\[
Pc = MC + TC + MCC + SCC + TCT + SCT + SPT \ldots \quad (1)
\]

Where: MC is the machine cost, TC is the tool cost, MCC is the machine change cost, TCC is the tool change cost and SCC is the set up change cost.

Total processing time (Pt) equation is

\[
Pt = MT + MCT + TCT + SCT + SPT \ldots \quad (2)
\]

Where: MT is the total machining time, MCT is the total machine change time, TCT is the total tool change time and SCT is the total set up change time.

GA for lubrication pumps stacking design. Sizing a pump stacking used in an aircraft lubrication system is a challenging task. The combination of several pumps, in parallel and in a single casing, must deliver specified oil flow rates, on a variable number of circuits and under given flight conditions. The optimal assembly has to minimize overall dimensions, weight and cost. This optimization problem involves a large space search, continuous and discrete variables and multi objectives. Genetic algorithm is well suited to solve that kind of problems [3][10].

Genetic algorithms in computer aided design. Design is a complex engineering activity, in which computers are more and more involved. The design task can often be seen as an optimization problem in which the parameters or the structure describing the best quality design algorithms and several ways in which they can solve difficult design problems They discussed several advanced genetic algorithms that have proved to be efficient in solving difficult design problems[4][10].

**BIOLOGICAL BACKGROUND**

All living organisms consist of cells. In each cell there is the same set of chromosomes. Chromosomes are strings of DNA and serve as a model for the whole organism. The genes determine a chromosome’s characteristic. Each gene has several forms or alternatives, which are called alleles, producing differences in the set of characteristics associated with that gene. The set of chromosome is called the genotype, which defines a phenotype (the individual) with certain fitness. During reproduction first occurs crossover. Genes from parents form in some way the whole new chromosome. The newly created offspring can then be mutated. Mutation means that the elements of DNA are bit-wise changed. The fitness of an organism is measured by success of the organism in its life. According to Darwinian Theory, the highly fit individuals are given opportunities to “reproduce” whereas the least fit members of the population are less likely to get selected for reproduction, and so “die out”, [12][13].

IJAET/Vol.III/ Issue II/April-June, 2012/114-118
GENETIC ALGORITHMS
A genetic algorithm is a search technique used in finding true or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms form particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination). Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype or the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals. Genetic algorithms form one of the best ways to solve a problem for which a little is known. They are very general algorithms that work well in any search space. A genetic algorithm is able to create a high quality solution.

Background
Genetic algorithm (GAs) is a heuristic solution search or an optimization technique, originally motivated by the Darwinian principle of evolution through (genetic) selection. A GA uses a highly abstract version of evolutionary processes to evolve solutions to some given problems. Each GA operates on a population of artificial chromosomes. These are strings in a finite alphabet (usually binary). Each chromosome represents a solution to a problem and has fitness, a real number, which is a measure of the quality of the solution; to the particular problem. Starting with a randomly generated population of chromosomes, a GA carries out a process of fitness-based selection and recombination, to produce a successor population- the next generation. During recombination, selecting parent chromosomes and their genetic material is recombined to produce child chromosomes. These then pass into the successor population. As this process is iterated, a sequence of successive generations evolves and the average fitness of the chromosomes tends to increase until some stopping criterion is reached.

In this way, a GA “evolves” a best solution to any problem [7][11]. GA was first proposed by John Holland’s, as a means to find good solutions to problems that were otherwise computationally intractable. Holland’s schema theorem [8][11], and the related building block hypothesis, provided a theoretical and conceptual basis for the design of efficient GAs. It also proved straightforward to implement GAs due to their highly modular nature. As a consequence, the field grew quickly and the technique was successfully applied to a wide range of practical problems in science, engineering and industry. GA theory is an active and growing area, with a range of approaches being used to describe and explain phenomena not anticipated by earlier theory. In tandem with this, more sophisticated approaches for directing the evolution of a GA population are aimed at improving performance on classes of problem known to be difficult for Gas. [8]

The development and success of GAs contributed greatly to a wider interest in computational approaches based on natural phenomena. It is now a major stand of the wider field of computational intelligence, which encompasses techniques such as neural networks, and artificial immunology. Genetic algorithms are search methods that can be used for both solving problems and modeling evolutionary systems. Since it estimates a solution, GAs differs from other heuristic methods in several ways. The most important difference is that it works on a population of possible solutions, while other heuristic methods use a single solution in their iterations. Another important difference is that GAs is not a deterministic but a probabilistic one.

SYSTEMS OF LINEAR EQUATIONS
A system of equations is a collection of two or more equations with the same set of unknowns. In solving a system of equations, we try to find values for each of the unknowns that will satisfy every equation in the system. A system of linear equations (or linear system) is a collection of linear equations involving the same set of variables.

General form
A general system of m linear equations with n unknowns can be written as:

\[ a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\
\vdots \quad \vdots \\
\]

\[ am_1x_1 + \cdots + amnx_n = bm \]

Here \( x_1, \ldots, x_n \) are the unknowns \( a_{11}, \ldots, a_{mn} \) are the coefficients of the system, and \( b_1, \ldots, b_m \) are the constant terms.

Vector equation
One extremely helpful view is that each unknown is a weight for column vector in linear combination.

\[ X = \begin{bmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{1n} \\ a_{m1} \\ a_{m2} \\ \vdots \\ a_{mn} \end{bmatrix}, X_1 = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, b_1 = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \]

This allows all the language and theory of vector spaces (or more generally, modules) to be brought to bear. For example, the collection of all possible linear combinations of the vectors on the left-hand side is called their span, and the equations have a solution just when the right-hand vector is within that span. If every vector within that span has exactly one expression as a linear combination of the given left-hand vectors, then any solution is unique. In any event, the span has a basis of linearly independent vectors that do guarantee exactly one expression; and the number of vectors in that basis (its dimension) cannot be larger than \( m \) or \( n \), but it can be smaller. This is important because if we have \( m \) independent vectors, a solution is guaranteed regardless of the right-hand side, and otherwise not guaranteed.

Matrix equation
The vector equation is equivalent to a matrix equation of the form \( Ax=b \), where \( A \) is an \( m \times n \) matrix, \( x \) is a column vector with \( n \) entries, and \( b \) is a column vector with \( m \) entries.

\[ \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \]

The number of vectors in a basis for the span is now expressed as the rank of the matrix.
Solution set
A solution of a linear system is an assignment of values to the variables $x_1, x_2, \ldots, x_n$ such that each of the equations is satisfied. The set of all possible solutions is called the solution set. A linear system may behave in any one of three possible ways:
1. The system has infinitely many solutions.
2. The system has a single unique solution.
3. The system has no solutions.

System of linear equations in two unknowns
First, a linear equation in two unknowns: $x$ and $y$ is an equation of the form:
$$Ax + by \quad \ldots \ldots \quad (5)$$
Where $a$, $b$, and $c$ are numbers, and where $a$ and $b$ are not both zero. Second, a system of linear equations is just a collection of these beasts. To solve a system of linear equations means to find a solution (or solutions) $(x, y)$ that simultaneously satisfies all of the equations in the system.

SYSTEMS OF NONLINEAR EQUATIONS
Solving systems of nonlinear equations is one of the most difficult problems in numerical computations. The problem of solving a system of nonlinear equations is to select a vector of solutions $x$, such that a vector of functions $f$ is driven simultaneously to zero. To gain some insight into the problem, consider a most simple example in which the goal is to simultaneously drive two functions $f(x, y)$ and $g(x, y)$ to zeros:
$$f(x, y) = 0 \quad g(x, y) = 0, \ldots, (6)$$
(Write in separate lines)
The functions $f$ and $g$ are arbitrary functions, each of which has zero-contour lines that divide the $x$-$y$ plane. Points are sought at which the zero-contour lines of the two functions intersect. To help gain an appreciation for the difficulty of this problem, consider the zero-contour lines of two sample functions shown in Fig. 1. Here, the zero-contour lines of the two functions intersect at four points ($M_1$ through $M_4$). Thus, there are four $(x, y)$ pairs for which both functions are simultaneously driven to zero. There are at least two difficulties associated with the two-dimensional problem represented by Fig. First, there are four solutions to the problem often in nonlinear systems of equations, there are multiple roots. Second and generally more difficult to overcome.
The functions $f$ and $g$ are not necessarily related to one another. There is nothing special about the common points of zero-contour lines from either $f$’s or $g$’s perspective. Thus, in order to solve this problem completely, the entire zero-contour lines of each function involved must be mapped out. There is a number of efficient algorithms for minimizing a function of many variables.

Gauss–Legendre Integration
Numerical integration (quadrature) is a tool used by researchers and scientists to approximate definite integrals that cannot be efficiently solved analytically. Quadrature methods rely on the selection of quadrature nodes (locations at which the integrand function is to be evaluated) and weights (values used in the approximation of the integral) to approximate integrals. In most quadrature methods, the weights are selected to give quality results while the quadrature nodes are distributed uniformly over the limits of the integral. In Gauss–Legendre quadrature, both the quadrature nodes and the weights are selected, there is, however, a problem with this approach: a system of nonlinear equations is perhaps the most difficult problem in all of numerical computation. A general Gauss–Legendre N-point formula can be developed that is exact for polynomial functions of degree $\leq (2N-1)$, [9]. The general N-point formula results in a system of nonlinear equations to be solved to compute the value of the quadrature nodes $(x)$ and their associated weights $(w)$. The system of nonlinear equations that results from the general N-point formula is:
$$f_n (w_1, w_2, \ldots, w_m, x_1, x_2, \ldots, x_m) = \sum_{j=1}^{N} w_j x_j^n - \tau_{1-1}^{1} x^{n-1} = 0 \ldots (7)$$
For $n=1, 2, \ldots, 2(N-1), 2N$ As an example, the system for the two-point formula $(N=2)$ results in the following system of four nonlinear equations
$$w_1 + w_2 = 0$$
$$w_1 x_1 + w_2 x_2 = 0$$
$$w_1 x_1^2 + w_2 x_2^2 = 0$$
$$w_1 x_1^3 + w_2 x_2^3 = 0$$
This system of four nonlinear equations is difficult to solve using traditional search techniques as a Newton
method for two reasons. First, the value of \( x, w \) and \( w \) should be determined to several decimal places, second, the ability of a method to converge to the solution in this problem is highly sensitive to the quality of the first guess supplied.

**Fitness function**

The fitness function to be minimized by the genetic algorithms is:

\[
g = \max \{|f_i|\} \quad \text{for } i = 1, 2, 3, \ldots, N \quad (8)
\]

Where:

- \( g \) is the fitness function to be minimized,
- \( \max \{|f_i|\} \) is the maximum absolute value of individual equations in the system \( f(x) = 0 \)
- \( N \) is the number of equations in the system [9]

**CONCLUSION:**

We observe that the Gauss-Legendre numerical integration is more effective than fundamental method of Newton’s in system of nonlinear equation. The result of using GA compared to the exact solutions obtained by some different numerical methods turned to be confirming. As a future work we plan to apply genetic algorithms to educational time tabled training and designing artificial intelligence systems such as artificial neural networks. Also, the same technique can be used and applied to more complex non-linear system.

**ACKNOWLEDGEMENT**

Success is a manifestation, inspiration, motivation, diligence and motivation.

Working on this paper has been one of the most wonderful and exiting experience of my life. This paper not only bears the testimony of extensive effort but also reflect the cooperation, help and guidance which I received from time to time. It is obvious that I acknowledge them for their tender help without which would never have been completed.

I owe this moment of satisfaction with a sense of gratitude to my friend “Prof.Miss.G.J Abhayankar” for his valuable guidance, suggestions, active involvement with my paper from time to time and without whose help this paper would have been impossible.

I have great pleasure in offering my sincere thanks to all colleagues of my college for their kind help.

**REFERENCE**