World Applied Sciences Journal 5 (3): 282-289, 2008
ISSN 1818-4952

Towards Using Genetic Algorithm for Solving Nonlinear Equation Systems

[1]*Ibrahiem M.M. El-Emary* and [2]*Mona M. Abd El-Kareem*

[1]Department of Computer Engineering, Faculty of Eng., Al Ahliyya Amman Univ., Amman, Jordan
[2]Department of Mathematics and Computer Science, Faculty of Science, Suez Canal Univ., Ismailia, Egypt

Abstract: This paper proposes a new paradigm for solving systems of nonlinear equations through using Genetic Algorithm (GA) techniques. So, a great attention was presented to illustrate how genetic algorithms (GA) techniques can be used in finding the solution of a system described by nonlinear equations. To achieve this purpose, we apply Gauss–Legendre integration as a technique to solve the system of nonlinear equations then we use genetic algorithm (GA) to find the results without converting the nonlinear equations to linear equations. After that, we compare the obtained result that is achieved by using GA with the exact solution that is obtained by numerical methods. Also, in this paper, an approach to solve the system of nonlinear equations needed to define a Gauss–Legendre numerical integration is presented. The obtained results indicate that a GA is effective and represents an efficient approach to solve the systems of nonlinear equations that arise in the implementation of Gauss–Legendre numerical integration.

Key words: Gauss–Legendre Numerical Integration · Genetic Algorithms · Fitness Function and Newton type method

INTRODUCTION

Solving systems of nonlinear equations is one of the most difficult numerical computation problems. The convergences of the classical solvers such as Newton-type methods are highly sensitive to the initial guess of the solution. This paper aims to apply Genetic Algorithm as a softcomputing technique in solving this nonlinear equation system and investigating the major returned benefits as a result of using GA. First of all, GAs are computer programs which create an environment where populations of data can compete and only the fittest survive, sort of evolution on a computer. GAs is 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: -

Optimization: such as data fitting, clustering, trend spotting, path finding and ordering.

Management: distribution, scheduling, project management, courier routing, container packing, task assignment and timetables.

Financial: portfolio balancing, budgeting, forecasting, investment analysis and payment scheduling.

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).

In this paper, we try to use genetic algorithms to find the solution of a system of non-linear equations. It is difficult to describe the solution set of a non-linear system with infinitely many solutions. To avoid the disadvantages of solving large system of non-linear equations such as rounding error, inverting large matrices the genetic algorithms are introduced and discussed in details in this paper which is organized from five sections. In section two, related works are presented. Section three deals with discussing GA. Describing nonlinear equation systems is presented in section four. Section five deals with the simulated output results. Conclusion and future works are covered in the six section.

Related Works: N. Kaya [1] uses GA for machining fixture locating and clamping position optimization. In that work, the application of genetic algorithms (GAs) to the

Corresponding Author: Ibrahiem M.M. El-Emary, Department of Computer Engineering, Faculty of Eng., Al Ahliyya Amman Univ., Amman, Jordan

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. Traditionally, machining fixtures were designed and manufactured through trial-and-error, which prove to be both expensive and time consuming to the manufacturing process. But that work shows 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 is shown that fixture layout problems are multi modal, therefore heuristic rules for fixture design should be used in GA to select best design among others.

L.Li *et al*. [2] use 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. That work presented 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 optimal or near-optimal process plans based on the chosen criteria. Distributed manufacturing system as depicted in Fig. 1, 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 optimal 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. 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 a six-operation process plan. '001' is the factory ID, 'Op4' represents operation 4; M-02, T-04 and +x in the second row represent the machine, tool and TAD, respectively, that will be used to perform operation 4, so are the other columns.

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

$$PC = MC + TC + MCC + SCC \quad (1)$$

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

$$PT = MT + MCT + TCT + SCT \quad (2)$$

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.

V.Kelner, O.Lèonard [3] use GA for lubrication pump 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.

G.Renner, A.Ekàrt [4] use 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.

B.M.Kariuki *et al*. [5] use genetic algorithms for solving crystal structures from powder diffraction data. That work used GA to tackle crystal structure solution from powder diffraction data in the case of a previously unknowns structure-ortho-thymotic acid. In that structure solution calculation, the structural fragment was subjected to combine translation and rotation within the unit cell, together with variation of selected intermolecular degree of freedom under the control of a genetic algorithm, in which a population of trail structures is allowed to evolve subject to well defined procedures for mating, mutation and natural selection. Importantly, the genetic algorithm approach adopts the ‘direct-space’ philosophy for structure solution and implicitly avoids the problematic step of extracting the intensities of individual reflections from the powder diffraction data. The structure solution was found efficiently in the genetic algorithm calculation and was then used as the initial structural model in Riveted refinement calculations.

R.K. Sahoo *et al*. [6] use genetic algorithms for multi-component aromatic extraction. That work applied (GA) which leads to globally optimal binary interaction parameters from multi-component liquid-liquid equilibrium data, has been recently demonstrated for some ternary, Quaternary and quinary systems. The binary interaction parameters are related to each other through the closure equations. In that work, the binary interaction parameters based on non-random two liquid (NRTL) activity coefficient model have been estimated using GA, without and with closure equations for 65 multi-component aromatic extraction systems: 53 ternary, 9 quaternary and 3 quinary systems. Parameters that satisfy the closure equations exhibit better root mean square deviations than those that do not satisfy the closure equations. Root mean square deviation value without implementation of closure equations is 0–80% better than literature as compared to 0–90% better with implementation of closure equations. Aromatics, such as benzene, toluene and xylene are considered essential in the chemical industry because they are the source of many organic chemicals. These aromatics are presented in naphtha. High purity aromatics are difficult to be separated using ordinary distillation operation, since they form several binary azeotropes with non-aromatics. Extraction is, therefore, a better choice to separate the aromatics from naphtha, as they are preferentially soluble in a variety of solvents. To predict the separation, it is necessary to know the liquid–liquid equilibrium (LLE) data for a particular system. Various activity coefficient models, such as the universal quasi chemical (UNIQUAC) and the non-random two liquid (NRTL) can be used to predict the LLE. Each of these models requires proper binary interaction parameters that can represent LLE for highly non-ideal liquid mixtures usually encountered in aromatic extraction. These parameters are usually estimated from the known experimental LLE data via optimization of a suitable objective function.

Genetic Algorithms: Based on the theory of genetics, the GA encodes each individual in the population with a chromosome [7,8]. This encoding represents the parameters for the objective function being optimized. There are several different techniques for encoding parameters, performing the selection and the alteration stages of the algorithm. The alteration stage is separated into Crossover and Mutation. The method used in this paper selects a random sample of parents from the population with a specified probability. An arithmetic crossover is then performed on these individuals which creates children based on a linear interpolation of the two parents. This is shown in Fig. 3.

A multi-non-uniform mutation is performed which modifies the parent parameters with a binomial distribution which narrows as the number of generations gets larger. More details of the individual techniques are given in [8].

Background: Genetic algorithm (GöA) 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 the given problem, [9]. GAs 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 [10] 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, [10]. 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 is heuristic (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.

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 define a phenotype (the individual) with certain fitness. During reproduction first occurs recombination (or crossover). Genes from parents form in some way the whole new chromosome. The new 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", [11, 12].

Systems of Nonlinear Equations: Solving systems of nonlinear equations is one of the most difficult problems in numerical computations. The difficulties associated with solving systems of nonlinear equations are magnified as the number of equations increases, but it is not unheard of for even five equations to be very difficult to solve. Since the problem can be exacting for even small systems and since there are numerous real life applications in which systems of nonlinear equations must be solved. This is an area in which potentially vast amounts of computational time can be saved. There are a myriad of real life applications for these problems from such diverse areas as distributions, multi-objective optimization and trajectory/path planning and many other's applications in which large systems of nonlinear equations must be solved efficiently. The problem of solving a system of nonlinear equations is to select a vector of solution 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:

$$\begin{aligned} f(x,y) &= 0 \\ g(x,y) &= 0 \end{aligned} \tag{3}$$

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 below in Fig. 4. 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. 4. 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

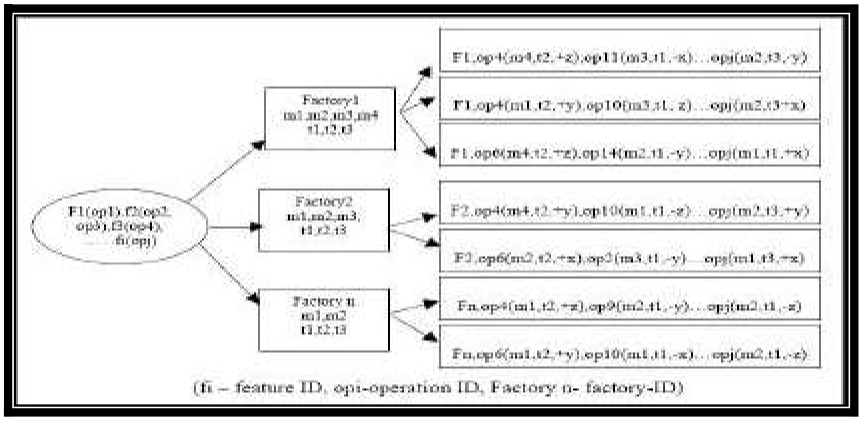

Fig. 1: Description of a Distributed Manufacturing System

	Op4	Op1	Op5	Op6	Op3	Op2
	m-02	m-03	m-02	m-01	m-03	m-01
001	t-04	t-02	t-03	t-02	t-01	t-04
	+x	-y	+z	-x	+x	-y

Fig. 2: Representation of a Process Plan

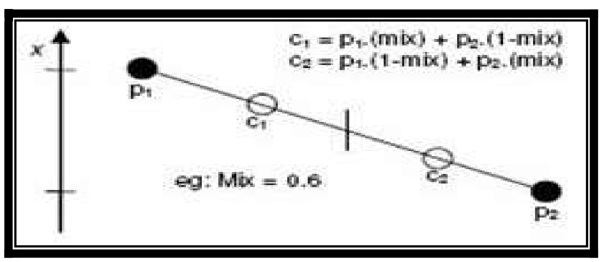

Fig. 3: One Dimensional Arithmetic Crossover Operator

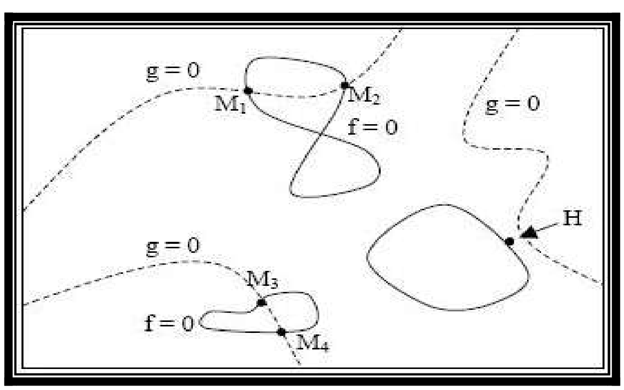

Fig. 4: The solution to the problem of interest occurs at points M1-M4 where the zero-contour lines intersect. Point H misleads because the zero-contour lines approach one another yet do not

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. However, the majority of these methods are not always effective in the minimization of the master function stemming from the problem of solving a system of nonlinear equations. Most are incapable of locating multiple roots, or they tend to converge to local minima. Optimization techniques such as variable metric methods, Brent's method and simplex search prove in adequate in the general N-dimensional nonlinear equation problem. Genetic algorithms consider multiple solutions to search problems simultaneously due to their population approach. Thus, they are effective in optimization problems with more than one optimum solution. Further, since genetic algorithms do not use derivative information, then they do not tend to get caught in local optima. GAs allows overcoming some of the shortcomings that prevent more traditional search techniques from effectively solving systems of nonlinear equations. However, genetic algorithms do not always converge to the true minimum in a search problem. Thus, they are not inviting stand–alone tools for minimizing the master function f(x) resulting from the problem of solving a system of nonlinear equations. The strength of genetic algorithms is that they rapidly converge to near-optimal solutions.

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 = (2N-1), [13]. The general N–point formula results in a system of nonlinear equations to be solved to compute the value of the quadrature nodes (x_j) and their associated weights (w_j). The system of nonlinear equations that results from the general N–point formula is

$$f_n(w_1, w_2, ..., w_n, x_1, x_2, ..., x_n) = \sum_{j=1}^{N} w_j x_j^{n-1} - \int_{-1}^{1} x^{n-1} = 0 \tag{4}$$

For n = 1,2,…,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

$$\begin{aligned} & w_1 + w_2 - 2 = 0 \\ & w_1 x_1 + w_2 x_2 = 0 \\ & w_1 x_1^2 + w_2 x_2^2 - \frac{2}{3} = 0 \\ & w_1 x_1^3 + w_2 x_2^3 = 0 \end{aligned} \tag{5}$$

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_1,x_2,w_1 and w_2 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.

The Proposed Methodology of Solution

Coding Scheme: The coding scheme for this problem is accomplished using a relatively standard coding scheme. In this scheme, the parameters of the search problem are represented as bit strings. For example, in the two-point Gauss-Legendre problem, there exist four equations in four unknowns. The equations are presented collectively as equation (5) and the unknowns are w_1, w_2, x_1 and x_2. The four parameters are each coded as binary strings according to the formula

$$p = p_{\min} + \frac{b}{(2^i - 1)}(p_{\max} - p_{\min}) \tag{6}$$

Where:

P is the parameter being coded. P_{min} is the minimum value of the parameter (in the case of x_j, the minimum value is –1, while for w_j, it is 0); P_{max} is the maximum value of the parameter (in the case of both x_j and w_j, the maximum values are 1). And b is the decimal value associated with a binary string of length.

Table 1: Simulated Output Results

Generation No.	Fitness function	Nodes	Weights
5	0.0812178122	0.5843137255	0.9882352941
		- 0.4980392157	1.0000000000
10	0.0784045726	0.5921568627	0.9725490196
		- 0.4980392157	1.0000000000
15	0.0196078431	0.5921568627	0.9803921569
		- 0.5843137255	1.0000000000
18	0.0101038062	0.5764705882	0.9960784314
		- 0.5843137255	1.0000000000
20	0.0039215686	0.5764705882	0.9960784314
		- 0.5764705882	1.0000000000
22	0.0020299885	0.5764705882	1.0000000000
		- 0.5764705882	1.0000000000
25	0.0020299885	0.5764705882	1.0000000000
		- 0.5764705882	1.0000000000

The exact solution of this system is given as:

Value of function	Value of nodes	Value of weights
0	0.5773502692	1.0000000000
	- 0.5773502692	1.0000000000

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

$$g = \max [\text{abs} (fi)] \text{ for } i = 1, 2, 3\ldots N \tag{7}$$

Where:

g is the fitness function to be minimized, max [abs (fi)] is the maximum absolute value of individual equations in the system f(x) = 0 and N is the number of equations in the system [13].

Simulated Output Results: In this section, we solve the linear system of the equation (3) using the following parameters:

- The number of quadrate nodes = 2
- The number of associated weights = 2
- String length of each parameter is = 8
- The population size of each generation =1000
- The crossover probability = 0.5
- The mutation probability = 0.05
- The number of generation = 25

From the results of the Table 1 which indicates G.A. in solving linear system, we see that the solution is almost equal to the analytical one. Nevertheless on using the G.A. to solve linear system, we obtain the result very swiftly.

CONCLUSION AND FUTURE WORK

Results of solving nonlinear equation systems indicate that a GA is effective and represents an efficient approach that arises in the implementation of Gauss–Legendre numerical integration. 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 training and designing artificial intelligence systems such as artificial neural networks, prediction of three dimensional protein structure. Also, the same technique can be used and applied to more complex non-linear systems.

REFERENCES

1. Kaya, N., 2006. Machining Fixture Locating and Damping Position Optimization Using Genetic Algorithms, Computers in Industry, 57: 112-120.
2. Li, L.J., Y.H. Fuh, Y.F. Zhang and A.Y.C. Nee, 2005. Application of Genetic Algorithm to Computer-Aided Process Planning In Distributed Manufacturing Environments", Robotics and Computer – Integrated Manufacturing, 21: 568-578.
3. Kelner, V. and O. Léonard, 2004. Application Of Genetic Algorithms To Lubrication Pump Stacking Design, J. Computational and applied Mathematics 168: 255-265.
4. Renner, G. and A. Ekárt, 2003. Genetic Algorithms In Computer Aided Design ", Computer Aided Design, 35: 709-726.
5. Kariuki, B.M., H.S. González, R.L. Johnston and K.D.M. Harris, 1997. The Application Of A Genetic Algorithm For Solving Crystal Structures From Powder Diffraction Data, Chemical Physics Letters 280: 189-195.
6. Sahoo, R.K., T. Banerjee, S.A. Ahmed and A. Khanna, 2006. Improved Binary Parameters Using GA for Multi-Component Aromatic Extraction: NRTL Model Without And with Closure Equations, Fluid phase Equilibria, 239: 107-119.
7. Bajcsy, R. and S. Kovacic, 1989. Multiresolution elastic matching, Computer Vision, Graphics and Image Processing, 46: 1-21.
8. Michalewicz., Z., 1999. Genetic Algorithms + Data Structures = Evolution Programs 3rd Ed."; Springer; New York.

9. McCall, J., 2005. Genetic Algorithms for Modeling and Optimization, J. Computational and Appl. Mathematics, 184: 205-222.
10. Jähne, B., 2002. Digital Image Processing", Springer – Verlag Berlin, Heidelberg, 2002.
11. "Genetic Algorithms". http://www.tjhsst.edu/~ai/AI2001/GA.htm
12. Sandikci, B., Genetic Algorithms .www.ie.bilkent.edu.tr/~Lors/ie572/barhaneddin.pdf
13. Ilgin, M.A., 2006. Joint Optimization of Spare Parts Inventory And Maintenance Policies Using Hybrid Genetic Algorithms 2006.http://www.fbe.deu.edu.tr/tezler/2006/YL_t2021.pdf