International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 1, July 2014)

Discrete Optimization of Truss Structure Using Genetic Algorithm

Mallika Alapati[1]

Associate Professor, Dept. of Civil Engineering, VNR Vignana Jyothi Institute of Engineering & Technology/Hyderabad/ Andhra Pradesh, India

***Abstract*-- During the last few years, several methods have been developed for the optimal design of structures. However, most of them, because of their calculus-based nature, treat the search space of the problem as continuous, when it is really discrete. Sometimes this leads to unrealistic solutions and therefore, they are not used in practice, which still prefers to rely on the more traditional iterative methods. This paper describes and uses genetic algorithm approach which explains that in a certain community, only the best organisms survive from all the adverse effects, i.e.," survival of the fittest ".**

This paper describes the use of genetic algorithm (GA) in performing optimization of 2D truss structures to achieve minimum weight. The GA uses fixed length vector of the design variables which are the cross-sectional areas of the members. The objective considered here is minimizing the weight of the structure. The constraints in this problem are the stress and deflection in no member of the structure should exceed the allowable stress and deflection of the material. As a case study this method is applied to a bench mark example of 10 bar truss and the results provided show that the minimum weight obtained is further reduced by the methodology adopted.

***Keywords*-- Optimal design, optimization, Genetic algorithm**

I. INTRODUCTION

Typical structural optimization problem evolves the search for the minimum of a stated objective function subject to various constraints on such performance measures as stresses and deflections, and also restricted by practical minimum sizes, or dimensions of structural members or components. If the design variables may be continuously varied between practical extremes, the problem is termed continuous, while if the design variables represent a selection from a set of parts, the problem is considered discrete.

Many of the mathematical and numerical methods for optimization rely on the assumption of continuity of both the design variables and objective function. Under these assumptions, if the structural problem is actually discrete in nature, than the resulting optimum values of the continuous design variables must be converted to appropriate discrete values.

A conservative approach is to round to the next larger available values and to check that the constraints are still satisfied. Park and Lee (1992) have proposed an alternative method of conversion as a post-processor of structural optimization problems. But this approach seems to be limited to moderate size problems.

A. Structural Optimization

Structural optimization involves the challenge of providing the most efficient design: i.e. the least expensive design which will satisfy all the design criteria for the entire life span of the structure. The cost of the structure is reduced by optimizing the use of materials (which is related to the weight of the structure) and labor (which involves fabrication and construction time and is related to the weight and complexity – topology and configuration of the structure). Since, configuration and topological optimization are nearly impossible to conduct using existing design and optimization methods, only sizing optimization is usually attempted in structural design. Topological and configuration optimization are, thus actively researched fields in structural optimization. The introduction of GAs into the field of structural optimization has opened new avenues for research because they have been successful where traditional methods have failed.

B. Trusses and Design of trusses

A truss is a system of straight bars joined together at their ends to form a rigid framework to transfer the loads applied to the supports in the form of purely axial (compressive or tensile) forces. All trusses are actually three-dimensional structures, but most can be reduced to planar or two dimensional trusses with the loads and reactions acting in their plane. The trusses that are analyzed here are based on a mathematical model, called the ideal planar truss, which is subject to the following assumptions: (1) all external forces are applied at the nodes or joints. (2) Bars are connected by frictional less hinges. (3) Each bar is subjected to axial stress only and this stress is constant along the length.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 1, July 2014)

Traditionally trusses are designed by first a choosing a particular topology (number of nodes and connectivity between nodes) and configuration (position of the nodes). Then the cross-sectional areas of the members are assumed. The truss is analyzed to obtain the stresses in the members and deflections at the nodes using finite element computer programs or truss-analysis software. The member areas are increased if the stresses are very less than the allowable stresses and reduced if the stresses are higher than allowable stresses. The truss is again analyzed and this is carried out till the weight is "sufficiently" reduced and the stresses and deflections are within their respectable limits.

C. Truss Optimization

The optimization of truss structures can be classified into three categories depending on what component of the structure is used as design variable: (1) Sizing (2) Configuration and (3) topological optimization. In sizing optimization of trusses the cross-sectional areas of the members are the design variables and the coordinates of the nodes and the connectivity between various members are fixed. This can be made more practically useful by restricting the member areas to pre-specified discrete values. In configuration optimization the design variables are the nodal coordinates, and in topological optimization the number of nodes and the connectivity between nodes are the design variables. These optimization problems have been discussed separately; however the most efficient design will be obtained by considering all three simultaneously. In general, multilevel optimization methods are used in which topological optimization is first performed keeping the configuration and member sizes fixed. When an optimal topology is found, configuration and/or sizing optimization is performed on the topology found in the previous step. As mentioned earlier this method will not provide the most optimal solution as all the three problems are not linearly separable. As a result, traditional methods of optimization have failed and the use of other tools such GAs is gaining popularity in the field of structural optimization. The application of GAs to truss optimization is discussed later.

D. Related techniques

Ant colony optimization (ACO) uses many ants (or agents) to traverse the solution space and find locally productive areas. While usually inferior to genetic algorithms and other forms of local search, it is able to produce results in problems where no global or up-to-date perspective can be obtained, and thus the other methods cannot be applied.

Bacteriologic Algorithms (BA) inspired by evolutionary ecology and, more particularly, bacteriologic adaptation. Evolutionary ecology is the study of living organisms in the context of their environment, with the aim of discovering how they adapt. Its basic concept is that in a heterogeneous environment, you can't find one individual that fits the whole environment. So, you need to reason at the population level. BAs have shown better results than GAs on problems such as complex positioning problems (antennas for cell phones, urban planning, and so on) or data mining. The Cross-entropy (CE) method generates candidates solutions via a parameterized probability distribution. The parameters are updated via cross-entropy minimization, so as to generate better samples in the next iteration.

Cultural algorithm (CA) consists of the population component almost identical to that of the genetic algorithm and, in addition, a knowledge component called the belief space.

Evolution strategies (ES, see Rechenberg, 1994) evolve individuals by means of mutation and intermediate and discrete recombination. ES algorithms are designed particularly to solve problems in the real-value domain. They use self-adaptation to adjust control parameters of the search.

Evolutionary programming (EP) involves populations of solutions with primarily mutation and selection and arbitrary representations. They use self-adaptation to adjust parameters, and can include other variation operations such as combining information from multiple parents.

Extremal optimization (EO) Unlike GAs, which works with a population of candidate solutions, EO evolves a single solution and makes local modifications to the worst components. This requires that a suitable representation be selected which permits individual solution components to be assigned a quality measure ("fitness"). The governing principle behind this algorithm is that of *emergent* improvement through selectively removing low-quality components and replacing them with a randomly selected component. This is decidedly at odds with a GA that selects good solutions in an attempt to make better solutions.

Gaussian adaptation (normal or natural adaptation, abbreviated NA to avoid confusion with GA) is intended for the maximisation of manufacturing yield of signal processing systems. It may also be used for ordinary parametric optimisation.It relies on a certain theorem valid for all regions of acceptability and all Gaussian distributions. The efficiency of NA relies on information theory and a certain theorem of efficiency.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 1, July 2014)

Its efficiency is defined as information divided by the work needed to get the information. Because NA maximises mean fitness rather than the fitness of the individual, the landscape is smoothed such that valleys between peaks may disappear. Therefore it has a certain "ambition" to avoid local peaks in the fitness landscape. NA is also good at climbing sharp crests by adaptation of the moment matrix, because NA may maximise the disorder (average information) of the Gaussian simultaneously keeping the mean fitness constant.

Genetic programming (GP) is a related technique popularized by John Koza in which computer programs, rather than function parameters, are optimized. Genetic programming often uses tree-based internal data structures to represent the computer programs for adaptation instead of the list structures typical of genetic algorithms.

Grouping Genetic Algorithm (GGA) is an evolution of the GA where the focus is shifted from individual items, like in classical GAs, to groups or subset of items. The idea behind this GA evolution proposed by Emanuel Falkenauer is that solving some complex problems, a.k.a. *clustering* or *partitioning* problems where a set of items must be split into disjoint group of items in an optimal way, would better be achieved by making characteristics of the groups of items equivalent to genes. These kind of problems include Bin Packing, Line Balancing, Clustering w.r.t. a distance measure, Equal Piles, etc., on which classic GAs proved to perform poorly. Making genes equivalent to groups implies chromosomes that are in general of variable length, and special genetic operators that manipulate whole groups of items. For Bin Packing in particular, a GGA hybridized with the Dominance Criterion of Martello and Toth, is arguably the best technique to date.

Harmony search (HS) is an algorithm mimicking musicians behaviors in improvisation process.

Interactive evolutionary algorithms are evolutionary algorithms that use human evaluation. They are usually applied to domains where it is hard to design a computational fitness function, for example, evolving images, music, artistic designs and forms to fit users' aesthetic preference.

Memetic algorithm (MA), also called *hybrid genetic algorithm* among others, is a relatively new evolutionary method where local search is applied during the evolutionary cycle. The idea of memetic algorithms comes from memes, which unlike genes, can adapt themselves. In some problem areas they are shown to be more efficient than traditional evolutionary algorithms.

Simulated annealing (SA) is a related global optimization technique that traverses the search space by testing random mutations on an individual solution. A mutation that increases fitness is always accepted. A mutation that lowers fitness is accepted probabilistically based on the difference in fitness and a decreasing temperature parameter. In SA parlance, one speaks of seeking the lowest energy instead of the maximum fitness. SA can also be used within a standard GA algorithm by starting with a relatively high rate of mutation and decreasing it over time along a given schedule.

Stochastic optimization is an umbrella set of methods that includes GAs and numerous other approaches.

Tabu search (TS) is similar to Simulated Annealing in that both traverse the solution space by testing mutations of an individual solution. While simulated annealing generates only one mutated solution, tabu search generates many mutated solutions and moves to the solution with the lowest energy of those generated. In order to prevent cycling and encourage greater movement through the solution space, a tabu list is maintained of partial or complete solutions. It is forbidden to move to a solution that contains elements of the tabu list, which is updated as the solution traverses the solution space.

II. History Of GA's

"Genetic algorithms are inspired by Darwin's theory about evolution. Simply said, solution to a problem solved by genetic algorithms is evolved".

Genetic algorithms in particular became popular through the work of John Holland in the early 1970s, and particularly his book Adaptation in Natural and Artificial Systems (1975). His work originated with studies of cellular automata, conducted by Holland and his students at the University of Michigan. Holland introduced a formalized framework for predicting the quality of the next generation, known as Holland's Schema Theorem. Research in GAs remained largely theoretical until the mid-1980s, when The First International Conference on Genetic Algorithms was held in Pittsburgh, Pennsylvania.

As academic interest grew, the dramatic increase desktop computational power allowed for practical application of the new technique.

What are GA's?

Genetic algorithm use a vocabulary borrowed from natural genetics. We would talk about *individuals* (or genotypes, structures) in a population; quite often these individuals are called also *strings* or *chromosomes.*

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 1, July 2014)

Each genotype (in this book a single chromosome) would re[resent a potential solution to a problem (the meaning of a particular chromosome, i.e., its *phenotype*, is defined externally by the user); an evolution process run on a population of chromosomes corresponds to a search through a space of potential solutions. Such a search requires balancing two (apparently conflicting) objectives: exploiting the best solutions and exploring the search space.

Hill climbing is an example of strategy which exploits the best solution for possible improvement; on the other hand, it neglects exploration of the search space. Random search is a typical example of strategy which explores the search space ignoring the exploitations of the promising regions of the space. Genetic algorithms are a class of general purpose (domain independent) search methods which strike a remarkable balance between exploration and exploitation of the search space.

GAs have been quite successfully applied to optimization problems like wire routing, scheduling, adaptive control, game playing, cognitive modeling, transportation problems, traveling salesman problems, optimal control problems, database query optimization, etc.

GA Terminology

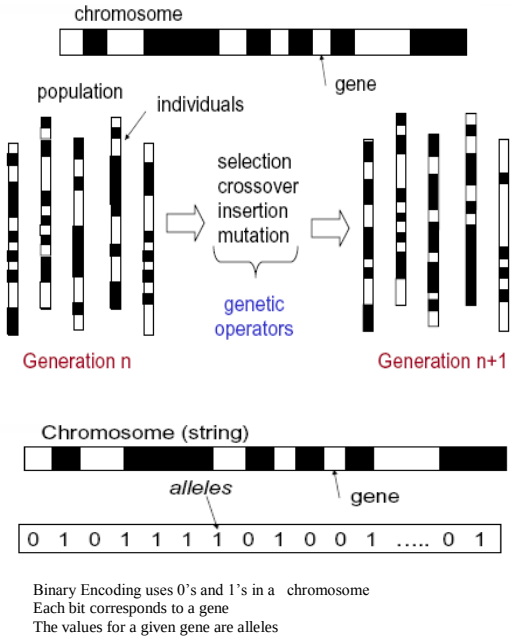

Figure 1: GA Terminology

Individual - Any possible solution
Population - Group of all individuals
Search Space - All possible solutions to the problem
Chromosome - Blueprint for an individual
Trait - Possible aspect of an individual
Locus - The position of a gene on the chromosome
Genome - Collection of all chromosomes for an individual
Selection - Individuals chosen for later breeding.
Crossover- Selecting genes from parent chromosomes and creating new off springs.
Mutation- Randomly changes the resulting offspring from crossover.

A. Methodology

The evolution usually starts from a population of randomly generated individuals.

B. Initialization

In each generation, the fitness of every individual in the population is evaluated, multiple individuals are selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population.

The new population is then used in the next iteration of the algorithm.

Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

Fitness

Fitness (often denoted *w* in population genetics models) is a central concept in evolutionary theory. It describes the capability of an individual of certain genotype to reproduce, and usually is equal to the proportion of the individual's genes in all the genes of the next generation. If differences in individual genotypes affect fitness, then the frequencies of the genotypes will change over generations; the genotypes with higher fitness become more common. This process is called natural selection.

An individual's fitness is manifested through its phenotype. As phenotype is affected by both genes and environment, the fitnesses of different individuals with the same genotype are not necessarily equal, but depend on the environment in which the individuals live. However, since the fitness of the genotype is an averaged quantity, it will reflect the reproductive outcomes of all individuals with that genotype.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 1, July 2014)

As fitness measures the quantity of the *copies* of the genes of an individual in the next generation, it doesn't really matter how the genes arrive in the next generation. That is, for an individual it is equally "beneficial" to reproduce itself, or to help relatives with similar genes to reproduce, as long as similar amount of copies of individual's genes get passed on to the next generation. Selection which promotes this kind of helper behaviour is called kin selection.

The **fitness function** is evaluated for each individual, providing fitness values, which are then normalized. Normalization means multiplying the fitness value of each individual by a fixed number, so that the sum of all fitness values equals.

- The population is sorted by descending fitness values.
- Accumulated normalized fitness values are computed (the accumulated fitness value of an individual is the sum of its own fitness value plus the fitness values of all the previous individuals). The accumulated fitness of the last individual should of course be 1 (otherwise something went wrong in the normalization step!).
- A random number *R* between 0 and 1 is chosen. The selected individual is the first one whose accumulated normalized value is greater than *R*.

C. Selection (genetic algorithm)

Selection is the stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding (recombination or crossover).

D. Selection schemes

- Roulette wheel selection with scaling
- Stochastic tournament selection with a tournament size of two
- Remainder stochastic sampling without replacement
- Remainder stochastic sampling with replacement
- Elitism

E. Crossover (or) Recombination operator

Crossover and mutation are two basic operators of GA. Performance of GA depends on them. Type and implementation of operators depends on a problem. There are many ways how to do crossover and mutation. The considered crossover methods are

Single point crossover
Two point crossover

F. Mutation

This operator occurs much less frequently both in nature and GA.

The basic idea of using this operator is to introduce some diversity into the population. In other words, to delay the situation in which all the population becomes so homogeneous that no further improvement is possible. Bit inversion - selected bits are inverted as shown in following figure:

III. Present Problem

G. The 10bar truss optimization

The discrete weight optimization of a 10-bar plane truss shown in Figure has been solved using a Genetic Algorithm concept and STAAD-Pro and MATLAB softwares have been used. Twenty Five angle sections have been taken from the standard Indian sections available and are given in SI units.

The assumed data :-

Modulus of elasticity E=2.05e008 kN/m^2
Density of the material =76.8195kN/m^3
Allowable stress=±172.25 MPa
Allowable displacement=±55.385mm

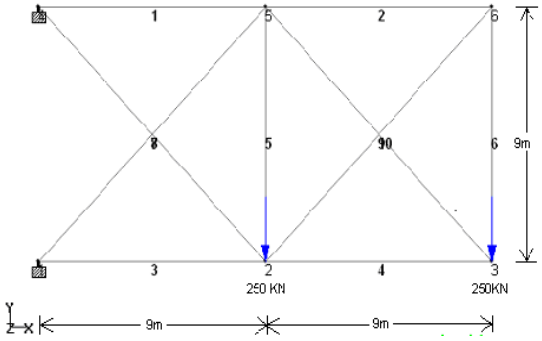

Figure 2: Truss Geometry

The mathematical Formulation of the given truss with the constraints has been given below:-

Mathematical definition of Objective Function

The optimization problem is the minimization the weight of the structure subject to stress, displacement and minimum member size constrains.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 1, July 2014)

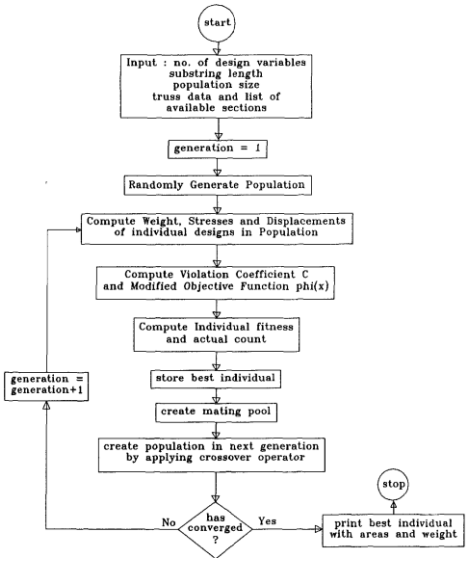

Figure 3: Flowchart

The objective function is:

$$W = \sum_{i=1}^{nel} \rho_i L_i A_i$$

Where *Li* is the length, ρ_i is the material specific weight, and *Ai* is the cross sectional of the *i*-th bar. Note that the objective function is linear on the design variables *Ai*.

The problem is subject to tensile and compressive stress constraints, bounds on displacements, and side constraints on the areas, as follows:

$$\sigma_i^c \le \sigma_i \le \sigma_i^t \qquad i = 1, \ldots nel$$

$$U_i^{min} \le U_i \le U_i^{max} \qquad i = 1, \ldots nel$$

$$A_i^{min} \le A_i \le A_i^{max} \qquad i = 1, \ldots nel$$

where,

$\sigma_i = i^{th}$ tensile stress.

$\sigma_i^C = i^{th}$ bar limit compressive stress(lower bound);

$\sigma_i^T = i^{th}$ bar limit tensile stress(upperbound);

U_i = displacement of i^{th} degree of freedom.

$U_i^{min} = i^{th}$ degree of freedom minimum displacement(lower bound);

$U_i^{max} = i^{th}$ degree of freedom maximum displacement(upperbound);

$A_i^{min} = i^{th}$ bar minimum cross-sectional area.

$A_i^{max} = i^{th}$ bar maximum cross sectional area.

Genetic algorithm cannot be applied to constrained problems so we need to change the constrained problem into unconstrained one and also to make the variables discrete and not continuous as in other methods.

For changing the constrained problem into unconstrained one we introduce a penalty function which is given below.

Min W= p*ΣA(i)*L(i) +P[Σδi +Σδk]
δi=IσiI-σall<=0, i=1,2,…10.
δk=IΔkI-Δall<=0, k=1,2,…12.
δ=1 for g>0.
δ=0 otherwise.

IV. Conclusions

A. Results and discussions

After the first generation the truss is analyzed in software based on stiffness matrix method of analysis and fitness value of genes are calculated by writing a program code in MATLAB and then cumulative probabilities are calculated and the best fitness value is retained

For the generation of the next population a crossover percentage of 25% is considered i.e. 20 numbers between 0 and 1 is generated and Cumulative probability of the genes less than 0.25 are selected and crossover operator is applied.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 1, July 2014)

Similarly for mutation a mutation percentage of 1% is considered and the gene of a particular chromosome is mutated. Then the fitness values of the genes are calculated and the worst fitness values are replaced by the best fitness value of the previous generation and this process is repeated till the fitness values of the 20 genes converges and the stresses and displacements are within allowable limits. After this process we get the optimized weight of the structure which is the main objective of this project. The results are tabulated and appropriate graphs are plotted to show the variations between the fitness values and the number of generations and also between the fitness value of the chromosomes and the number of chromosomes.

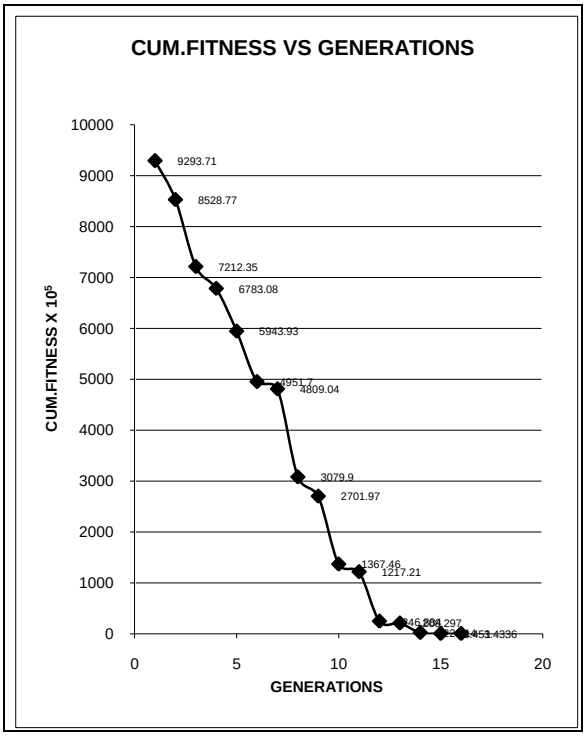

Figure 4: Graph showing Fitness Vs Generation no

As we can observe from the graph the fitness value decreases from the first generation (92.9371 X 10^7) to the sixteenth generation(3.1434 X 10^4). Final weight of the structure is 1548.83 kg

From the above tables we can observe that stresses (165N/mm^2) and the displacements(55.385mm) are with in allowable limits and also the optimized weight of the truss structure has been obtained.

Conclusion

- A larger initial population will lead to a better convergence and a much better optimized truss.
- Mutation rate should not be set too high otherwise passing on the good genes becomes difficult.
- Selection process is based on retaining the best from the previous generation (Selection based on Elitism).
- The cumulative fitness value decreases from the first generation to the last generation as the objective is minimization and the optimized convergence is achieved.
- For trusses with higher configurations higher number of generations are required for optimization.

Scope of Future Work

- The study can be extended by considering different selection procedures.
- The same Genetic Algorithms can be applied to trusses of different configurations.
- Apart from trusses it can also be applied to various other problems in different fields**.**
- A **C-programming** or a **Matlab** code can be generated for retaining the best individual from the previous generation which will make the work much simpler and also leads to a better solution.

REFERENCES

[1] David.E.Goldberg, 1989,"Genetic Algorithms in Search, Optimization and machine learning",, Addison-Wesley Publishing Co.

[2] S.Rajeev and C.S.Krishnamoorthy, 1992, "Discrete optimization of Structures Using Genetic Algorithms", ASCE Journal of Structural Engineering, Vol.118, No.5.pp.1233-1250

[3] Carlos Artemio Coello ,1991, "Discrete Optimization of Trusses using Genetic Algorithms", Ph.D Thesis,Tulane University, New Orleans

[4] Noyan Turkkan, "Discrete optimization of Structures Using a Floating Point Genetic Algorithms, Moncton, N.B., Canada

[5] C.A. Coello,A.D. Christiansen, 2000, "Multi Objective Optimization of Truss Using Genetic Algorithms", Volume 75, Issue 6, 647–660

[6] Deb .K and Gulati .S, 2001, "Design of Truss Structure for Minimum Weight Using Genetic Algorithms", Finite Elements in Analysis and Design, Vol.37, 447- 465.

[7] Rajah.S, 1995, "Sizing Shape and Topological Optimization of Trusses Using Genetic Algorithms", Journal of Structural Engineering, 1480-1487