IMPACT: Journal of Computational Sciences and
Information Technology (IMPACT:JCSIT)
ISSN(P): Applied; ISSN(E): Applied
Vol. 1, Issue 1, Dec 2016, 1-18

OPTIMIZATION OF STEEL FRAMES USING GENETIC ALGORITHM

PARIJATHA N

Assistant Professor, Department of Civil Engineering,

GSSS Institute of Engineering & Technology for Women, Mysore, Karnataka, India

ABSTRACT

One of the major purposes of optimization in civil engineering is to perform a suitable design for the structure. This goal has to fulfill technical criteria and contain the minimum economical costs. Building frames are of the most customary civil engineering structures. Genetic Algorithm which is one of the optimization methods inspired by nature, has overcome this problem. In order to solve such problems, genetic algorithm needs a multiple analyses of structures. Therefore, in this study attempts were made to introduce and embed new formulae into a newly developed program to handle new techniques for selection and mutation as genetic operations. This optimization technique can well substitute that of the deterministic one where a considerable factor of safety and therefore, a heavy structure as always is a must. For this purpose, one may take into account the behavior for load, yield stress, young modulus, etc, using parameters such as standard deviation and variance, through which safety remarks can be embedded into the design procedure by some mathematical relations, resulting to optimization technique.

KEYWORDS: Genetic Algorithm, Reproduction, Crossover, Mutation Objective Function Constraints Optimum Weight

INTRODUCTION

1.1 General

"The science of selecting best design from the acceptable ones with the aim of achieving better economy and functional performance is known as Optimization".

Optimization is the act of obtaining optimum result under given circumstances. In the design, construction and maintenance of any engineering system, engineers have to take many technological and managerial decisions at several stages. The ultimate goal of all such decisions is to either maximize the desired benefit or minimize the effort required. Since the effort required or the benefit desired in any practical situation can be expressed as a function of certain design variables, optimization can be defined as the process of finding the conditions that give the maximum or minimum value of a function. Function which is expressed in terms of design variables is called objective function. There is no single method available for solving all optimization problems efficiently. Hence a number of optimization methods have been developed for solving different types of optimization problems.

The optimum seeking methods are also known as mathematical programming techniques and are generally studied as a part of operations research. Operations research is a branch of mathematics concerned with the application of scientific methods and techniques to decision making problems and with establishing the best or optimal solutions.

Genetic Algorithms are the form of random search methods based on the mechanics of natural genetics. They combine the concept of DARWIN'S theory of 'SURVIVAL OF THE FITTEST' with genetic operators to form a robust search procedure. Its simplicity of approach and directness in discrete variable combinations make it more attractive than mathematically complex methods.

1.2 Problem Statement

The processes of obtaining the optimum design of structures are very complex to solve by hand, due to large number of design variables, objectives. Typically, the design is limited by constraints such as the choice of material, feasible strength, displacement deflect on, size constraints, load cases, support conditions, and true behavior of beam to column connection. Hence, one must decide which parameters can be modified during the optimization process. Usually, structural optimization problems involve searching for the minimum of the structural weight in steel structure. This minimum weight design is subjected to various constraints with respect to performance measures, such as stresses and displacements, and restricted by practical minimum cross-sectional areas or dimensions of the structural members or components.

1.3 Structural Optimization

1.3.1 Main Advantages of Steel Structures

Steel is a universally used material. It is used either separately or combined with another material e.g. reinforced concrete. Its popularity may be attributed to the combined effects of several factors, the most important of which are: it possesses great strength, it exhibits good ductility, its fabrication is easy, and its is relatively cheap. In addition, steel is the ultimate recyclable material. Several advantages are listed below:

Advantages of Steel Structures

Ease of Erection

No formwork needed and minimum carnage required for the erection, many parts of the structure can be prefabricated away from the site, and it is largely self-supporting during erection.

Modifications

Either extensions or strengthening is relatively straightforward. Possible reuses after a structure is disassembled or scrap value even though not reusable.

Uniformity

The properties of steel do not change appreciably with time, as do those of reinforced-concrete structures.

Low Self-Weight

Permits large clear spans without intermediate columns

Dimensional Control

Prefabrication in the workshop ensures accurate work

1.4 Problems Associated with the Analysis and Design of Steel Frame Structures

The design of structural steelwork is a process based on many contributing aspects: past experience of successful and unsuccessful construction, laboratory tests and results of research, combining to ensure structures do not fail.

Structures can therefore be used efficiently and safely but at the same time must be economically built and maintained.

From this it can be understood that the design process must satisfy two conflicting aim economy and safety. Achieving this compromise is not an easy task, consequently codes of practice have evolved to assist and guide the designer, but different national codes,

1.4.1 The Need for Design Optimization

One can address several difficulties that may face the structural designer when utilizing conventional design.

- Firstly, the skill and experience of the designer, which could lead to completely different designs.
- Secondly, the complexity of the treated structure makes the difficulty of doing several re-analyses and subsequently redesigns.
- Thirdly, there is difficulty of handling all possible loading cases.
- Fourthly, the intended usage of the structure might prevent the designer from achieving economical design.
- Fifthly, the alternative design and analysis techniques might confuse the designer in choosing the appropriate technique.

Therefore, the use of computers has made reliable and accurate analysis much easier, and the speed with which alternative solutions can be analysed makes it possible to achieve more economical designs than were attainable in the past. Design optimization is therefore an interesting research topic, and recommendations for design optimization have been made by design experts among them (MacGinley, 1997 and Adeli, 1994). Design optimization is concerned with the problem of the selection of geometric parameters and mechanical strength properties for the structural elements. This selection consists of a search for the extreme solutions, which satisfy the prescribed criteria, the search being conducted in an objective and rational way that does not rely on the intuition or special abilities of the designer. Thus, design optimization takes over that part of the design process, which consists of selecting sizes and subsequently checking that the required criteria have been met. The question arises whether the design optimization field can or should fully replace traditional designing procedures, that is, whether or not the task of optimization is to embrace all structural parameters so that the solution of an optimization problem should be equivalent to obtaining a complete design of a structure. This question will be answered in this thesis.

1.5 Proposed Study

In the present study the above trend in optimization is given due importance. Hence 'SIMPLE GENETIC ALGORITHM' (SGA) is used.

A problem as opposed to an academic one with the following characteristics is selected.

The design variables are discrete.

- The material of frame is steel.
- The structural optimization is carried out using 'SIMPLE GENETIC ALGORITHM'.

Software Description

1.6.1 ANSYS

ANSYS is powerful in representing the partially restrained connections with a non-linear spring element. Also, ANSYS is used as its reason for second-order behaviour is evaluated accurately for partially restrained frames.

1.6.1.1 Modeling of Steel Frame Structures with ANSYS

Some of the basic frame analysis methods such as slope deflection, moment distribution, stiffness and flexibility methods can be modified to work with partially restrained connections but tend to be very tedious and complicated. Because most structural engineering use computers in the analysis of frames, there are several software packages designed to analyze structures such as SAP2000 and STAAD. The problem is that they cannot represent partially restrained connection behaviour with moment-rotational curve.

In this study, ANSYS software was used to model various elements and connection of steel structures. ANSYS is powerful in representing the partially restrained connections with a non-linear spring element. Also, ANSYS is used as its reason for second-order behaviour is evaluated accurately for partially restrained frames.

1.6.2 TURBO C++

TURBO C++ is a programming language: As a programming language, C is rather like Pascal or Fortran. Values are stored in variables. Programs are structured by defining and calling functions. Program flow is controlled using loops, if statements and function calls. Input and output can be directed to the terminal or to files. Related data can be stored together in arrays or structures.

Of the three languages, C allows the most precise control of input and output. C is also rather more terse than Fortran or Pascal. This can result in short efficient programs, where the programmer has made wise use of C's range of powerful operators. It also allows the programmer to produce programs which are impossible to understand.

1.7 Scope and Objective

of the Present Work

The following are the scope and objective of the present work:

- Aim of this thesis work is to do an optimal design of steel frames for minimum cost. In steel structures it is assumed that, cost and weight are proportional. Hence in the present study optimal design of frames is arrived by minimizing the weight of the frame.
- In the present study 'SIMPLE GENETIC ALGORITHMS' (SGA), which are becoming popular in engineering optimization problems in the recent past are used.
- To perform above problem the commercial available softwares like ANSYS and TURBO C++ are used.
- Presently researchers have at their disposal a number of finite element packages, which allow obtaining all mechanical characteristics of the structure: nodal displacements, stresses, reactive forces at supports, etc.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter reviews all the works that has been carried out previously related to the present work.

2.2 Review of Literature

2.2.1 By S. Pezeshk, Member, ASCE, Carried Out the Work on "Design Optimization of 2D "Steel Frame Structures"

This chapter presents a genetic algorithm for design optimization of multi–bay multi– storey steel frameworks according to BS 5950 to achieve our objectives.

The first is to ascertain that the developed GA approach can successfully be incorporated in design optimization in which framework members are required to be adopted from the available catalogue of standard steel sections. The design should satisfy a practical design situation in which the most un favorable loading cases are considere.

- The second is to understand the advantages of applying automated design approaches.
- The third is to investigate the effect of the approaches, employed for the determination of the effective buckling length of a column, on the optimum design. Here, three approaches are tackled and results are presented.
- The fourth is to demonstrate the effect of the complexity of the design problem on the developed algorithm.

This chapter starts with describing the design procedure for steel frame structures according to BS 5950, then combines this procedure with the GA to perform design optimization of the steel frame structures. The design method obtained a steel frame structure with the least weight by selecting appropriate sections for beams and columns from BS 4.

This paper concluded that:

The optimizer is successfully linked to a finite element package for a more accurate treatment of the determination of the effective buckling length that leads to achieving a more economical design. It is interesting to note that even some of the powerful computer software packages available today for the design of steel frameworks such as CSC and STAAD–III require the structural designer to input the effective buckling length factor as a parameter. In this study, computation of the effective buckling length is automated and included in the developed algorithm.

2.2.2 S. Y. MAHFOUZ University of Bradford, UK, 1999, Carried Out Work on, "Design Optimization of Structural Steel Work".

This thesis deals with the design optimization of 2D and 3D steel frame structures.

A computer code based on the direct method for the stability analysis of 2D steel frame structures has been developed and verified. This code is then used to compare the values of the effective buckling length of columns with those determined from the charts presented in BS 5950.

The versatility of GAs in dealing with discrete design optimization is demonstrated. Modifications to GA are implemented to improve its performance. It is shown that the choice of parameters in a GA can considerably affect its robustness and speed of convergence. A technique is developed to deal with a case when the number of catalogue cross sections does not fit into a string keeping the probability of selection equal for all sections.

Applications to design optimization of 2D and 3D steel structures are presented. In order to consider realistic steelwork design problems, a GA has been linked to a system of structural design rules (British Standards BS 5950 and BS 6399), interacting with a finite element analysis package ANSYS. A steelwork optimization problem is considered as a selection of the optimum set of practical cross sections from a catalogue (British Standard BS 4, BS 4848). The design criteria of the codes of practice and other practical designer's considerations are reflected in the formulation of the optimization problem.

2.2.3 By S. Pezeshk, 1Member, ASCE, C. V. Camp, Associate Member, ASCE, Andd. Chen, Member, ASCE Carried Out the Work on "Design of Nonlinear Framed Structure Using Genetic Optimization".

In this paper we present a genetic algorithm (GA)-based optimization procedure for the design of 2D, geometrical, nonlinear steel-framed structures. The approach presented uses GAs as a tool to achieve discrete nonlinear optimal or near-optimal designs. Frames are designed in accordance with the requirements of the AISC-LRFD specification. In this paper, He employs a group selection mechanism, discuss an improved adapting crossover operator, and provide recommendations on the penalty function selection. He compares the differences between optimized designs obtained by linear and geometrically nonlinear analyses. Through two examples, we will illustrate that the optimal designs are not affected significantly by the *P*-D effects.

CHAPTER 3

GENENTIC ALGORITHM

3.1 Introduction

Genetic algorithm is a method for solving optimization problems that is based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution. We can apply the genetic algorithm to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear.

3.2 Basic Concepts

GAs are good at taking larger, potentially huge, search spaces and navigating them looking for optimal combinations of things and solutions which we might not find in a life time.

GAs is very different from most of the traditional optimization methods. Genetic algorithms need design space to be converted into genetic space. So, Genetic algorithms work with a coding of variables. The advantage of working with a coding of variable space is that coding discreteness the search space even though the function may be continuous. A more striking difference between GAs and most of the traditional optimization method is that GA uses a population of points at one time in contrast to the single point approach by traditional optimization methods. This means that GA processes a number of designs at the same time.

3.3Working Principle

The GA is an iterative optimization procedure. Instead of working with a single solution in each iteration, a GA works with a number of solutions (collectively known as population) in each iteration. A flowchart of the working principle of a simple GA is shown in Figure 1.

In the absence of any knowledge of the problem domain, a GA begins its search from a random population of solutions. We shall discuss about the detail of coding procedure a little later. But now notice how a GA processes strings in iteration. If a termination criterion is not satisfied, three different operators – reproduction, crossover and mutation – are applied to update the population of strings. One iteration of these three operators is known as a generation in the parlance of GAs. Since the representation of a solution in a GA is similar to a natural chromosome and GA operators

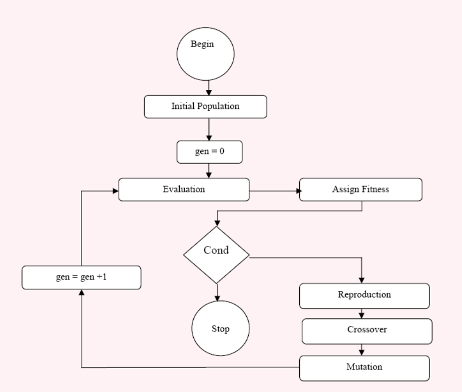

Fig. 2.1: Flowchart of working principle of a genetic algorithm

Figure 1

3.3.1Genetic Operators

Algorithm derives their power from genetic operators. A Simple Genetic Algorithm that yields good results in many practical problems composed of three operators.

- Reproduction
- Crossover
- Mutation

Other than these, low level operators like dominance, inversion, deletion, translocation, segregation, duplication etc., can also be incorporated. Depending on the nature of the problem and on the requirements for performance SGAs can be improved by applying more and more of these operators.

Reproduction

Reproduction (or selection) is usually the first operator applied to a population. Reproduction selects good strings in a population and forms a mating pool. The essential idea is that above-average strings are picked from the current population and duplicates of them are inserted in the mating pool. The commonly used reproduction operator is the proportionate selection operator, where a string in the current population is selected with probability proportional to the string's fitness. Thus, the ith string in the population is selected with probability proportional to . Since the population size is usually kept fixed in a simple GA, the cumulative probability for all string in the population must be one.

Crossover

The crossover operator is applied next to the string of the mating pool. In crossover operator, two strings are picked from the mating pool at random and some portion of the strings is exchanged between the strings. In a single-point crossover operator, both strings are cut at an arbitrary place and right-side portion of both strings are swapped among themselves to create two new strings, as illustrated in the following:

Crossover is the recombination operator. After strings for mating are selected, a crosstie is selected at random and bits are swapped between the strings following the cross site. Crossover is performed at a fairly high probability (i.e.) 0.6 to 0.8.

Parent 1	0 0 0 0 0	0 0 1 1 1	Child 1
	→		
Parent 2	1 1 1 1 1	1 1 0 0 0	Child 2

In the single-point crossover operator search is not extensive, but the maximum information is preserved from parent to children. On the other hand, in the uniform crossover, the search is very extensive but minimum information is preserved between parent and children strings. If a crossover probability of PCis used then 100PC% strings in the population are used in the crossover operation and100 (1- PC) %of the population are simply copied to the new population.

Mutation

Crossover operator is mainly responsible for the search aspect of genetic algorithms, even though the mutation operator is also used for this purpose sparingly. The mutation operator changes a 1 to a 0 and vice versa with a small mutation probability: Pm

0 0 0 **0** 0

↓

0 0 0 **1** 0

In the above example, fourth gene has changed its value, thereby creating a new solution. The need for mutation is to maintain diversity in population. For example, if in a particular position along the string length all strings in the population have a value 0, and a 1 is needed in that position to obtain optimum or a near-optimum solution, then mutation operator described above will be able to create a 1 in that position. The inclusion of mutation introduces some probability of turning that 0 into 1. Furthermore, for local improvement of a solution, mutation is useful.

3.4 Encoding

Various encoding methods have been created for particular problems to provide effective implementation of genetic algorithms. According to what kind of symbol is used as the alleles of a gene, the encoding methods can be classified as follows:

- Binary encoding
- Real-number encoding

- Integer or literal permutation encoding

3.4.1 Binary Encoding

Binary encoding (i.e., the bit strings) are the most common encoding used for several of reasons. One is historical: in their earlier work, Holland and his students concentrated on such encodings and genetic algorithms practices have tended to follow this lead. Another reason for that was because much of existing GAs theories is based on the assumption of using binary encoding.

3.4.1. a Crossover

The single point, multi point, and uniform crossover can implement in binary encoding. The effect of each one of these crossover type is shown in the figure (2)

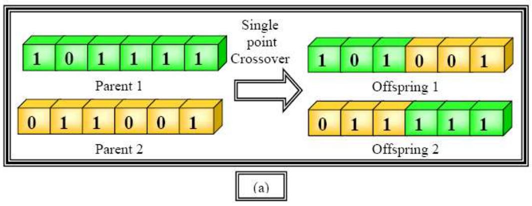

Figure 2: Explanation of Crossover Effect on Binary String. Single Point Crossover

3.4.1. b. Mutation

The single point and multi point can implement in binary encoding, the effect of mutation is shown in the figure (3)

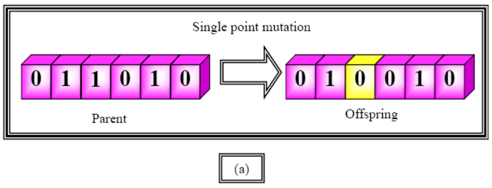

Figure 3: Explanation of Mutation Effect on Binary String. (a) Single Point Mutation

3.5 General Steps Followed By a Genetic Algorithm

The general steps followed by a Genetic Algorithm process can be summarized as:

- Initialize the population
- Evaluate initial population
- Perform competitive selection
- Apply genetic operators to generate new solutions
- Evaluate solutions in the population
- Repeat steps 3 through 5 until some convergence criteria are satisfied.

3.6 Application of Genetic Algorithms

Genetic algorithms (GAs) are adaptive methods which may be used to solve search and optimization problems. The power of GAs comes from the fact that the technique is robust and can deal successfully with a wide range of problem areas, including those which are difficult for other methods to solve. Therefore, the main ground for GAs is in difficult areas where no such solving techniques exist. Even where existing techniques work well, improvements can be made by mixing them with GAs.

GAs in various forms is implemented to wide range of problems including the following:

- Optimization: GAs has been used in a wide variety of optimization tasks, including numerical optimization and combinatorial optimization problems such as circuit design and job shop scheduling.
- Automatic Programming: GAs has been used to evolve computer programs for special tasks and to design other computational structures cellular automata and sorting networks.
- Machine and robot learning: GAs have been used for many machine learning applications, including classification and prediction tasks such as the prediction of dynamical systems, weather prediction , and prediction of protein structure. GAs have also been used to design neural networks, and to evolve rules for learning classifier systems or symbolic production systems and to design and control robots.

3.7 Advantages and Disadvantage of Genetic Algorithms

Advantages

- Optimizes with continuous or discrete variables.
- Doesn't require derivative information.
- Simultaneously searches from a wide sampling of the cost surface.
- Deals with a large number of variables.

Disadvantages

Optimization algorithms have the disadvantage that some kind of initial guess is required and this may bias the final result. GAs on the other hand only require a search range, which need only be constrained by prior knowledge of the physical properties of the system. Effectively they search the whole of the solution space, without calculating the fitness function at every point. This can help avoid a anger in any optimization problem which is being trapped in local maxima or minima. There are two main reasons for this:

- The initial population, being randomly generated, will sample the whole of the solution space, and not just a small area.
- Variation inducing tactics, i.e. crossover and mutation, prevent the algorithm being trapped in one part of the solution space.

4.4 Flow chart of genetic algorithm for steel frame optimization:

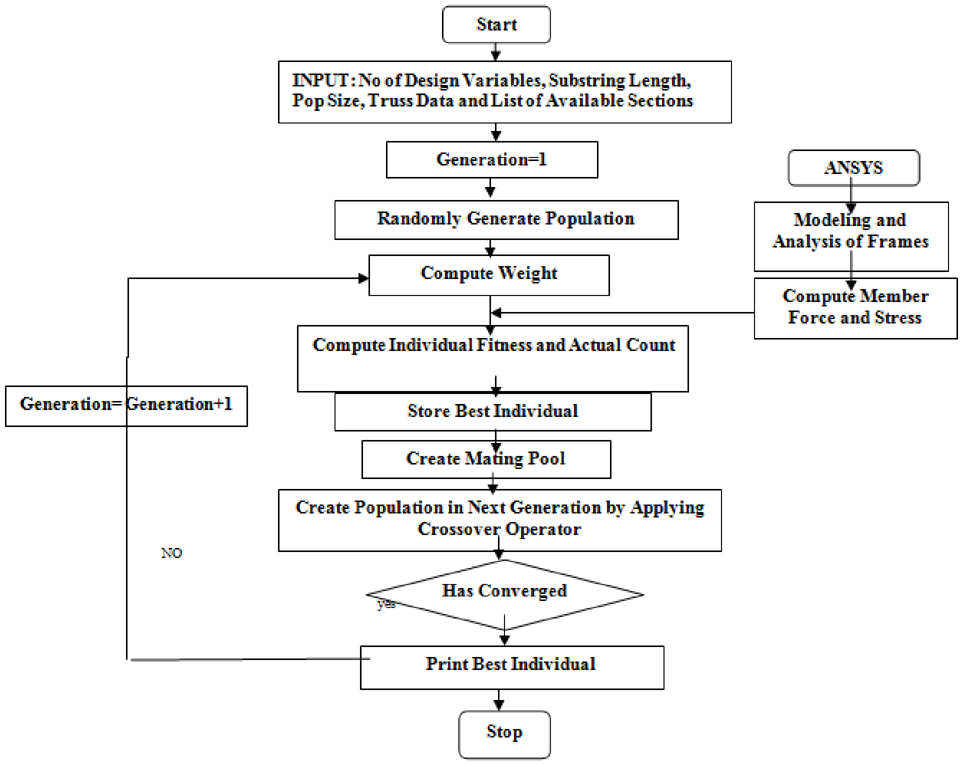

Figure 4

4.5. Frame Details

Span = 16 m

Height = 7.2 m

Support = fixed

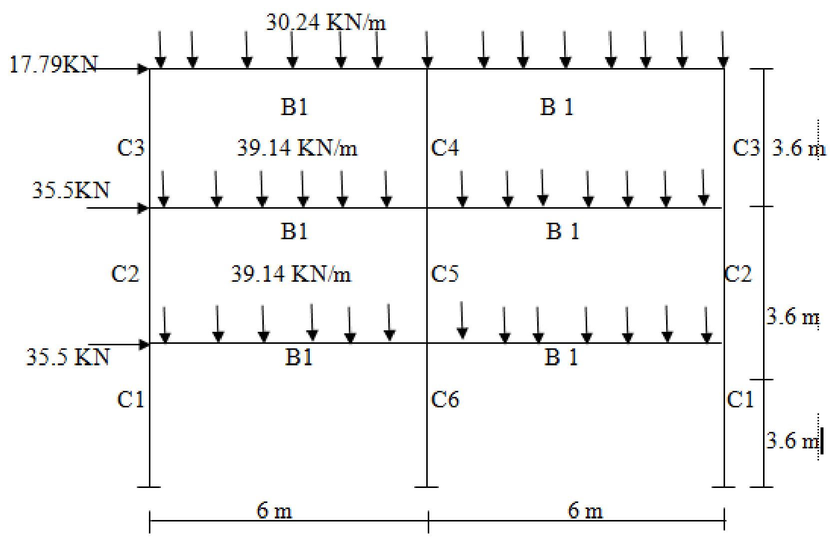

Figure 5: Simple Steel Frame

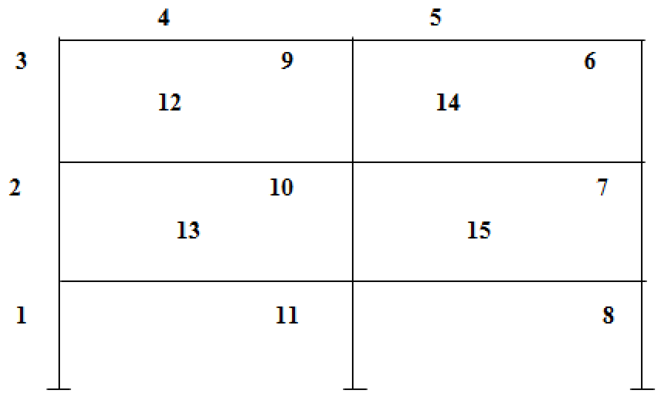

Figure 6: Frame with Members

Table 1: Section Properties of Three-Storey, Two-Bay Frame

Type	Section	Area(cm2)	I(mm4)	r (mm)
C1	W 12 x 35	25.47	1.1 x 10^4	21.3
C2	W 12 x 26	19.43	0.7 x 10^4	16.6
C3	W 8 x 24	17.98	3.2 x 10^4	17.5
C4	W 14 x 43	32.00	1.67 x 10^4	21.5
C5	W 12 x 30	22.33	0.9 x 10^4	23.8
C6	W 10 x 20	16.48	0.46 x 10^4	16.2
B1	W 16 x 25	19.51	1.17 x 10^4	16.6

4.8. Input

- **For Genetic Algorithms**
  - Population size
  - Chromosome length
  - Number of parameters
  - Length of parameter
  - Maximum number of generations
  - Crossover probability
  - Mutation probability
  - Set of areas.
- **For Analysis**
  - Number of members
  - Young’s modulus
  - Nodal loads
  - Member areas from GA.

4.9. Result

Optimum weight = 23.026 KN

Details of computations for case-1 are shown in the following tables. Program is run for three times with different population sizes, crossover probability and generations (refer table-2). With the values obtained graph is drawn between generation and weight.

Table-2: Program Variables

Variables	Case-1	Case-2	Case-3
Pop size	16	20	16
Max.Gen	100	150	150
Crossover prob	0.8	0.8	0.75
Mutation prob	0.01	0.01	0.01

Max Gen=100

Pcross =0.8

Pmutn =0.01

Table 3: Generation -1

No	POPULATION	A1	A2	A3	A4	f(x)	F	FACT	AC	POOL	PAIR	CS	NEW POPULATION
1	0000101011101111	24.47	72.50	22.56	24.42	45.395	1026.24	0.9799	1	0000101011101111	(3 2)	9	1011000001110111
2	1010000010001010	72.50	24.47	80.25	72.50	84.337	272.73	0.2595	0	0001000001110111			0010010001110001
3	0010010001110001	17.98	22.33	26.20	19.43	27.988	1249.67	1.1892	1	0010010001110001	(4 2)	0	1110000001110001
4	1011000001110111	17.92	24.47	26.20	26.22	31.779	1221.75	1.1626	1	1011000001110111			0010010011111110
5	1110000011111110	22.76	24.47	24.42	22.76	31.008	1195.99	1.1381	1	1110000011111110	(2 6)	5	1100100011111110
6	0110101101101100	19.51	17.92	19.51	19.52	25.317	1108.64	1.0550	1	0110101101101100			1110000100010101
7	1100100100010101	19.52	85.91	19.43	16.48	43.102	989.53	0.9415	1	1100100100010101	(0 9)	15	1110000110100001
8	1110000110100000	22.76	19.43	72.50	24.47	43.993	884.55	0.8417	1	1110000110100000			0001000000111010
9	0001000000111011	19.43	24.47	32.00	17.92	29.928	1268.44	1.2070	1	0001000000111011	(2 1)	16	0110101101010101
10	0001000001110111	19.43	24.47	26.22	26.22	32.206	1290.63	1.2282	2	0001000001110111			1100100100101100

Avg fit = 1050.817

Max fit = 1290.63

Min wt. = 25.31

Table 4: Generation -30

No	POPULATION	A1	A2	A3	A4	f(x)	F	FACT	AC	POOL	PAIR	CS	NEW POPULATION
1	0110101101100001	19.51	17.92	19.51	19.43	25.274	706.149	1.0950	1	0110101101100001	(7 2)	11	1110001101010101
2	1110000100100001	22.76	19.43	19.98	19.43	26.18	604.133	0.9372	1	1110000100100001			0110101101010101
3	0110101101010101	19.51	17.92	16.48	16.48	25.20	747.120	1.1585	1	0110101101010101	(7 4)	11	1110001101100001
4	0110101100100001	19.51	17.92	17.98	19.43	24.84	738.160	1.1446	1	0110101100100001			1110001101010101
5	1110001101100001	22.76	32.00	19.51	19.43	30.17	647.649	1.0043	1	1110001101100001	(2 2)	4	1110001101100001
6	0110100100100001	19.51	85.91	17.98	19.43	44.08	213.471	0.3310	0	1110000100010101			1110000100010101
7	1110000100010101	22.76	19.43	19.43	16.48	25.20	669.605	1.0383	1	1110000100010101	(3 6)	4	1110001101010101
8	1110001101010101	11.76	32.00	16.48	16.48	27.93	574.240	0.8904	1	1110001101010101			0110101101100001
9	0110101101100001	19.51	17.92	19.51	19.43	25.27	756.128	1.1725	1	0110101101100001	(4 1)	16	1110000100010101
10	1110000100010101	22.76	19.43	19.43	16.48	25.20	791.736	1.2277	2	1110000100010101			1110000100010101

Avg fit = 644.839

Max fit =791.736

Min wt. = 24.84

Table 5: Generation -100

No	POPULATION	A1	A2	A3	A4	f(x)	F	FACT	AC	POOL	PAIR	CS	NEW POPULATION
1	1110000100010101	22.76	19.43	19.43	16.48	25.207	439.913	1.0990	1	1110000100010101	(6 9)	0	0110101101010101
2	1110001101100001	22.76	32.001	19.51	19.43	30.178	276.348	0.6903	1	1110001101100001			1110001101010101
3	111000100010101	22.76	9.43	19.43	16.48	25.207	455.639	1.1382	1	111000100010101	(7 3)	8	1110000100100001
4	1110000100100001	22.76	19.43	17.98	19.43	26.188	456.652	1.1408	1	1110000100100001			1110000100100001
5	1110001101100001	22.76	32.00	19.51	19.43	30.178	380.621	0.9508	1	1110001101100001	(9 2)	8	0110101101100001
6	1110001101010101	22.76	32.00	16.48	16.48.	27.930	236.534	0.5908	0	1110001101010101			1110001101010101
7	0110101101010101	19.51	17.92	16.48	16.48	23.026	431.621	1.0782	1	0110101101010101	(6 4)	12	1110000100100001
8	1110000100100001	22.76	19.43	17.98	19.43	26.188	334.178	0.8346	1	1110000100100001			1110000100010101
9	1110000100010101	22.76	19.43	19.43	16.48	25.207	489.759	1.2235	1	1110000100010101	(3 5)	11	1110001101010101
10	1110001101010101	22.76	32.00	16.48	16.48	27.930	501.656	1.2532	2	1110001101010101			0110101101010101

Avg fit = 400.292

Max fit =501.656

Min wt. = 23.026

GRAPH: GENERATION V_S WEIGHT

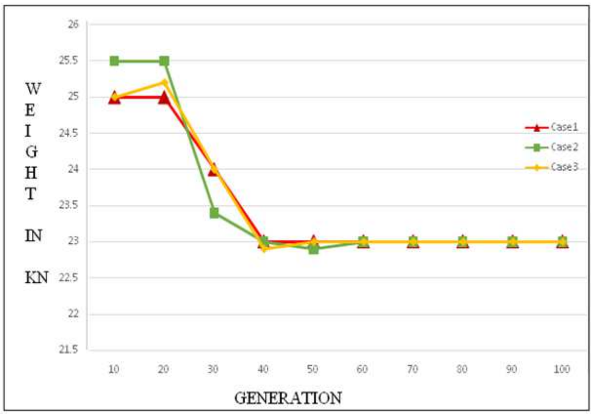

Figure 7

4.10 The Deformed Shape and Bending Moment for Non-Linear Analysis, is Shown in Figure 8 and Figure 9 Respectively. A Basic ANSYS Input File as in Appendix-A

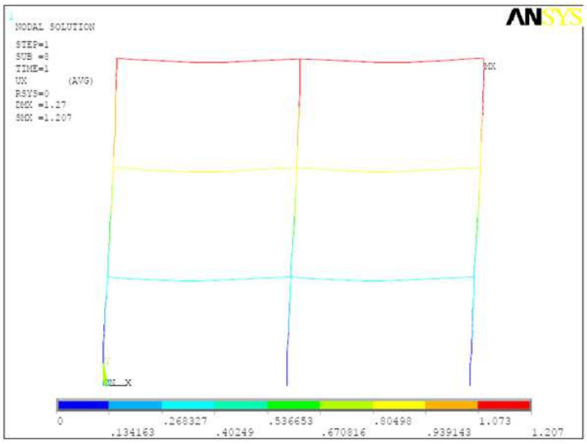

Figure 8: Three Storeys, Two Bay Deformed Shapes

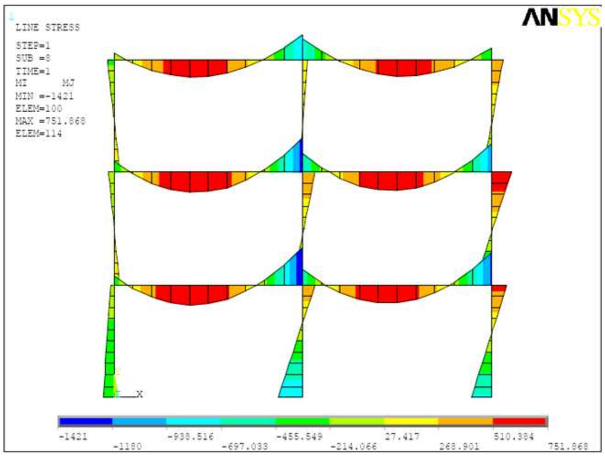

Figure 9: Three Storey Two Bay Bending Moment

Table 6: Tabulates the Comparison of Horizontal Displacements at the Upper Left Corner and Max Bending Moment at the Column Base with Semi-Rigid Frame by ANSYS

Column Base with Semi-Rigid Frame	Ansys
Upper left corner displacement (m)	**0.03**
Max bending moment(KN-m)	**23.16**

CHAPTER 5

DISCUSSION AND CONCLUSIONS

5.1. General

In the present study genetic algorithms which work on discrete variables are implemented for the optimization of truss and proved to be robust. The optimum solution generated is feasible both from a mathematical and practical point of view. GAs is used for solving problems where gradient computations are difficult. Though gradient computations are absent, GAs is slower compared to traditional algorithms. This is not a limitation in the present day computing environment, with fast computers and large amount of resources. The program developed for the present study can be used to optimize any other engineering structure by simply modifying the analysis part.

Genetic algorithm program is developed in TURBO C++ software and modelling and analysis of truss is carried out in ANSYS software.

- From graph it is clear that even when the parameters like population size, crossover probability and number of generations are changed, for same constraints specified by the user, the optimum weight is the same (i.e., 23.207 KN). This proves the robustness of GA.
- Constraints chosen are design strength in compression member and design strength in tension member. These are chosen according to IS 800-2007 provisions.
- From table-6 we can see that 10 individuals have the same weight. This emphasizes that the convergence criteria has been achieved.
- Also from graph it is evident that the optimum weight is arrived

In case1 during 40^{th} generation

In case2 during 39th generation

In case3 during 37th generation

Hence it is proved that the genetic operators have robustness in search process.

- In present work, simple genetic operators viz., reproduction, single site crossover and bitwise mutation are applied. Crossover and mutation are taking place at the probabilities specified by the user. Hence they are under the control of the user. In the present study higher crossover probability and lower mutation probability are adopted.

5.2. Suggestions for Further Work

- Multiple point crossovers can be implemented and the effect can be compared in future works.
- Other genetic operators like inversion, special, crowding, segregation etc., can be studied and their influence in the search can be tested.
- Method of fitness scaling may be changed. In future works linear scaling may be adopted.

REFERENCES

1. AsgharVataniOskouei, SahandSarioletlaghFard, OrhanAksogan,Carried out the work on ,"Using genetic algorithm for the optimization of seismic behavior of steel planar frames with semi-rigid connections".
2. AsgharvataniOskovei, ShahidRajaeeuniversity, Iran, SahandSarioletlaghFard&kaveshKouchmeshki, Islamic Azad University,Iran. Carried out work on, "Optimization of seismic behaviour of steel frames with semi-rigid connection using genetic algorithm".
3. Camp, Charles:Pereshk, shahram:Coa,Guozhong.(1997) carried out work on "Optimized steel frame Design using a genetic Algorithm.
4. Frye, M.J., Morris, G.A.: Analysis of flexibly connected steel frames. Can J Civil Engng 2(3), 280–91 (1975).
5. Guo-Qiang Li, Jin-Jun Li.: Advanced analysis and design of steel frames,
6. M.R. Ghasemi*a and M. Yousefib, Department of Civil Engineering, University of Sistan and Baluchistan, Iran and Chabahar Maritime University, Iran."Reliability-Based Optimization Steel Frame Structures Modified Genetic Algorithm".
7. Huang Jizhuo Wang Zhan (Department of Civil Engineering, Shantou University, Shantou 515063, China) (Department of Civil Engineering, South China University of Technology,Guangzhou 510641, China),carried out work on ,Multiobjective optimization design of aseismic steel frames using genetic algorithm".
8. Jenkins, W.M.: Towards structural optimization via the genetic algorithm. Computers and Structures 40, 1321–1327 (1991).
9. Jenkins, W.M.: Plane frame optimum design environment based on genetic algorithm. Journal of Structural Engineering ASCE 118, 3103–3112 (1992).

10. Kavehand B. DadfarDepartment of Civil Engineering, Iran University of Science and Technology,Narmak, Tehran-16, Iran, carried out work on,”Optimun seismic design of steel moment resisting frames by genetic algorithm”.
11. S. Y. MAHFOUZ University of Bradford, UK, 1999, carried out work on, “Design Optimization of structural Steel work”.
12. S. Pezeshk, Member,ASCE, carried out the work on “Design Optimization of 2D Steel Frame Structures”.
13. S. Pezeshk, 1 Member, ASCE, C. V. Camp, Associate Member, ASCE, and D. Chen, Member, ASCE carried out the work on "Design of Nonlinear Framed Structure Using Genetic Optimization".
14. PetrHradil, MattiMielonen and LudovicFülöp, carried out work on,” Advanced design and optimization of steel portal frames”.