Chapter 8

GA BASED SOFTWARE DEVELOPMENT AND APPLICATIONS

8.1 GENERAL REMARKS

GA requires random generation of binary strings to initiate the search process. The evaluation process requires the procedure for decoding of these strings to get real values of the design variables. The evaluation measure known as “fitness” is obtained after working out objective function which is to be maximized or minimized. This requires a procedure to analyze the problem for calculation of the problem constraints and penalty value for the infeasible solutions. After evaluating the solution, GA based procedures for selection of solution to evolve new offsprings and procedure mimicking mating of candidates to develop probably better offsprings are the necessary components of GA based optimization algorithm. A programming language facilitating the readymade functions for above procedures is the prime requirement of the GA based optimization program.

The structural optimization program also needs a front end tool which simplifies the data entry defining structure geometry, loading and support conditions. It also necessitates the tools showing analysis results during evaluation process of GA and the dynamic graphics indicating evolution of the structural shapes and convergence of the algorithm towards optimum. The program also requires graphical features showing the detail of the evolved structure or structural component. To facilitate all these operations three processors, namely pre-, main- and post-processors, are developed in the present chapter.

As R.C.C. is the most commonly and widely used material all over the nation, saving in the cost of R.C.C structures greatly affects the national economy. For R.C.C, being heterogeneous material, the cost is considered as the objective function as the cost of R.C.C element is the sum of costs of its ingredient including the cost of formwork. Further, in R.C.C components the design variables are continuous which can be effectively handled by Genetic Algorithm. In the present chapter, a wide variety of R.C.C structures are considered for cost optimization using GA based software.

Optimum shape design of skeletal structures has always been an active area of research in the field of search and optimization. Traditionally, various techniques based on classical optimization methods have been used to find optimal truss structures. Mathematical programming techniques to arrive at optimum solutions consider design variables as continuous which is not justifiable due to discrete nature of some of the variables which represent cross sectional areas of the steel structural members owing to their availability in standard sizes. Very few algorithms have been developed to handle the discrete nature of design variables. These algorithms find optimum designs which are practically feasible. Moreover, most of the methods use "ground structures" and are concentrated primarily on size optimization and at the most configuration optimization. The problem becomes challenging when sizing, configurational and topology optimizations are addressed simultaneously. In the present chapter capability of the GA based optimization program in solving size, configuration and topology optimization of plane and space trusses is illustrated through numerical examples after weight optimization of gantry girder and size optimization of components of steel frames.

In topology optimization of continuum structures, the structural shape is generated within a pre-defined design space. In addition, the user provides structural supports and loads. Without any further decisions and guidance of the user, the method forms the structural shape thus providing a first idea of an efficient geometry. Therefore, topology optimization is a much more flexible design tool than classical structural shape optimization, where only a selected part of the boundary is varied without any chance to generate any new material cavity, for example. In the present chapter after dealing with cost optimization of R.C.C structures and weight optimization of steel structures, the most difficult problem of topology optimization of continuum structures is attempted.

8.2 THE SELECTED ENVIRONMENT

In the present work Visual Basic is selected for the development of software for the following reasons:

- The structure of the basic programming language is very simple.
- It has a full array of mathematical, string handling and graphics functions.
- Apart from clipboard and printer access it has sequential and random access file support.
- It has one of the most powerful debugging and error-handling facilities.
- It provides ActiveX support and powerful database integration and access tools.

- *VB* is not only a language but primarily an integrated, interactive development environment ("IDE").
- The *VB*-IDE supports rapid application development ("RAD"). It is particularly easy to develop graphical user interfaces (GUI) and to connect them to handle functions provided by the application.
- The GUI of the VB-IDE provides intuitively appealing views for the management of the program structure in the large and the various types of entities (classes, modules, procedures, forms ...).
- *VB* provides a comprehensive interactive and context-sensitive online help system.
- While editing program texts, the "IntelliSense" technology informs you in a little popup window about the types of constructs that may be entered at the current cursor location.
- *VB* is a component integration language which is attuned to Microsoft's Component Object Model ("COM") which can be written in different languages and then integrated using VB.
- COM components can be embedded in / linked to the application's user interface and also into stored documents (Object Linking and Embedding "OLE", "Compound Documents").
- Packaging and deployment wizard makes distributing of application simple.
- VB has faster compiler, internet capabilities and data report designer.

8.3 GA FRIENDLY INBUILT FUNCTIONS IN *VB*

VB contains many inbuilt functions which are very much helpful for programming of GAs. The absence of these inbuilt functions would have required separate coding for them, as is the case with other programming languages. Some of these functions are discussed below:

8.3.1 *Rnd Function*

Generation of random initial population is the very first requirement while programming a GA. *VB* makes its easy to generate random numbers with just a single line coding by using the *Rnd* function. The *Rnd* function returns a pseudorandom number each time it is called. The returned value is always less than 1 and greater than or equal to 0.

8.3.2 *Mid, Left and Right Functions*

After generating a population of random initial strings, GA operators like crossover and mutation are operated on strings selected from the population. This process requires an exchange of part of one string with the other string or replacement of some bits of the string with other bits. This can be easily done by functions like *Mid*, *Left* and *Right* which extracts a substring from the middle, beginning or end of the source string.

8.3.3 User Defined Data (Variable) Types

Variables of several different types can be combined to create user-defined types. User-defined types are useful when it is required to create a single variable that records several related pieces of information. User-defined type can be created with the Type statement, which must be placed in the Declarations section of a module.

8.3.4 Important Components

❖ **ADODC:** The ADO Data control uses Microsoft ActiveX Data Objects (ADO) to quickly create connections between data-bound controls and data providers. Data-bound controls are any controls that feature a Data Source property. Data providers can be any source written to the OLE DB specification. The ADO Data control has the advantage of being a graphic control (with Back and Forward buttons) and an easy-to-use interface that allows us to create database applications with a minimum of code. In the present work ADODC is used to connect a database (prepared in MS ACCESS) of steel sections with the design module of truss topology application for checking the adequacy of the selected sections.

❖ **Data Grid:** The data grid component can be used as an input device as well as an output display device. Data grid control is the most usual way for displaying data in a database table. The data grid control is used with ADODC and MS ACCESS as back end tool. For example, it may be used to enter the coordinates for an irregular shaped geometry when used as an input device.

❖ **Flex Grid Control:** It is used as an output device, e.g. it can be used to display the results such as member end actions, reactions and nodal displacements. It can also be used for displaying various generations and populations with their corresponding fitness, weight, etc as output of the software.

❖ **Picture Box:** The picture box component can also be used as an input and output device. When considering it as an input device, it can be used for supplying structural input data like geometry, support conditions, material properties, etc on the screen itself. Line control can be used on the picture box to draw the member of the truss and point control can be used to display the nodes. Label control can be used to number the nodes and members at the run time. Shape control can be used for displaying roller supports.

- **Progress Bar:** The progress bar component is used to give a graphical display of the ongoing process in the software, e.g., it can be used for displaying progress of optimization process by graphical display of number of generations completed.

8.4 DEVELOPMENT OF GA BASED SOFTWARE

In the present work, in addition to some of the common modules based on GA, various forms, menus, subroutines and functions are developed in Visual Basic 6.0 to facilitate pre- and post-processing for each application separately. The main form directs user to select various RCC and steel structures to be optimized using GA. Upon selecting an application the sub-form related to the selected application opens and guides user to supply application oriented data. Details of Various forms developed are given application wise in the subsequent sections.

The GA based optimization application (project) is made up of:

- **Forms** which are windows that are created for user interface.
- **Controls** which are graphical features placed on the form to allow the user interaction (text boxes, labels, scroll bars, command buttons, etc.).
- Forms and control which are called **Objects**.
- **Properties** which are characteristic of a form or a control (names, captions, size, color, position and contents).
- **Methods** which are inbuilt procedures that can be invoked to impart some action to a particular object.
- **Event procedures** which are code related to some object. This is the code that is executed when a certain event occurs.
- **General procedures** which are codes not related to objects. This code must be invoked by the application.
- **Modules** which are collection of general procedures, variable declarations and constant definitions used by application.

8.4.1 The Pre-processor

The main aim of the pre-processor is to develop an environment for input of data, which is most friendly to the user. The GUI is one of the easiest and fastest ways of supplying data to the software. Other methods like numeric input of data are also useful when providing data for irregular geometries. Some of the features of GUI method of data input are listed below:

- Data input with least effort and minimum mistakes.
- Checking of input data at a simple glance.
- Easy editing and additions to supplied data.

The pre-processor comprises of various forms, menus, command buttons, toolboxes, list boxes, etc created for supplying GA related data such as number of generations, population size, length of binary string, crossover and mutation probabilities, selection method, type of crossover, type of mutation etc.

8.4.2 The Main-processor

This is the main module on which the computational time of the problem will depend. The flowchart for the same is shown in Fig. 8.1. Some of the important subroutines developed in the main processor are as follows.

- **Subroutine INITIAL:** For initial generation of random strings according to the size of population and accuracy required for each variable.
- **Subroutine APPLYGENETIC:** For separating the strings of each variable from the main string, decoding the value of each variable from its respective string and mapping each value according to the provided upper and lower limits of each variable.
- **Subroutine SELECTIT:** Based on the roulette wheel selection technique, this subroutine selects the strings with higher fitness for creation of the mating pool.
- **Subroutine STATISTICS:** This subroutine calculates vital statistics like maximum, minimum and average fitness of each generation for finding out scaled fitness and sum of fitnesses for selection procedure.
- **Subroutine SCALEFITNESS:** This subroutine scales the values of fitness of each solution.
- **Subroutine CROSSOVER_MUTATION:** For operating genetic operators like crossover and mutation on the generated random population of strings and it involves exchange of some portion of strings or replacement of some bits of the string with other bits.

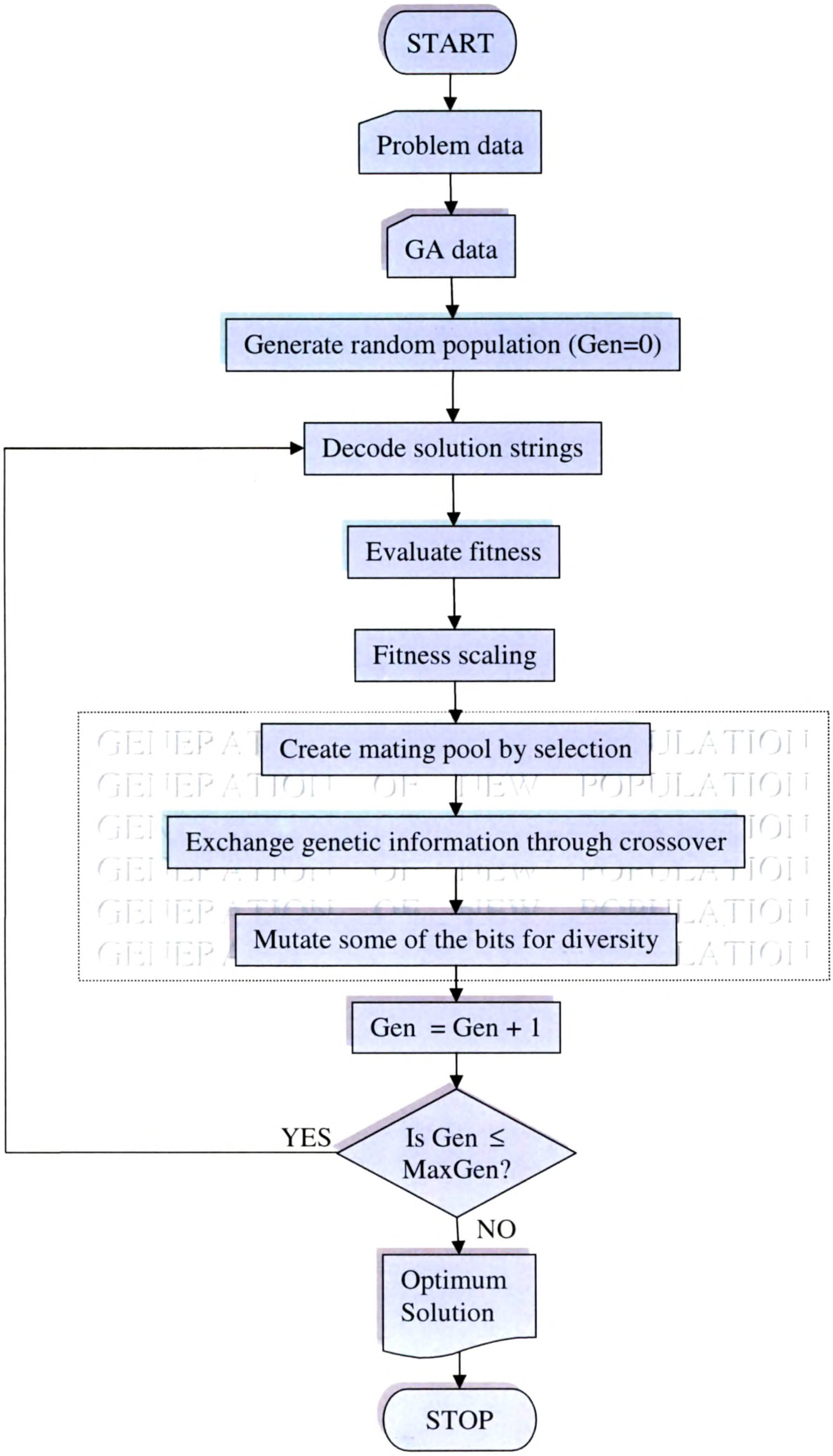

START
Problem data
GA data
Generate random population (Gen=0)
Decode solution strings
Evaluate fitness
Fitness scaling
GENERATION OF NEW POPULATION
Create mating pool by selection
Exchange genetic information through crossover
Mutate some of the bits for diversity
Gen = Gen + 1
Is Gen ≤ MaxGen?
YES
NO
Optimum Solution
STOP

Fig. 8.1 Flowchart Showing GA Procedure

8.4.3 The Post-processor

The post processor consists of facilities for displaying the results generated by the software. Separate forms have been prepared for displaying results. The following subroutines are developed in the post-processor:

- **Subroutine REPORT**: This subroutine generates a report for the respective application, giving information regarding the input data, parameter settings, final results and the design details. Another report titled "Generation History" is generated, giving information of average fitnesses at different generations. The reports are generated in two different formats namely *.txt and *.htm and can be printed directly without any formatting.
- **Subroutine DRAW_GRAPH**: This subroutine plots a graph of Generation v/s Fitness and Generation v/s Cost using the MS chart component of *VB*.

8.5 COST OPTIMIZATION OF RCC PLANE FRAME

Optimization of RCC plane frame involves optimization of size of beam and column elements that is nothing but minimization of the quantity of their constituents i.e. concrete and steel. Thus optimization of RCC structure is an optimization of multi-parameter function. Usually in such cases there is a need of exploring the effect of number of design variables on the computational time required. With two or more than two design variables it becomes difficult to predict their interactions and the designer's task becomes more complicated. The design variables, constraints and objective function chosen for RCC plane frame optimization are discussed below.

8.5.1 Design Variables

The depth and width of the beam or column of the plane frame are considered here as two variables. The binary string representation scheme is used. The length of string is selected so as to get variation in size up to 2 mm for columns and beams. Hence a combined string for beam and column of 16 bits (8 bits for each) is selected. Thus number of substrings in the solution string for a frame having n members will be n*2.

8.5.2 Objective Function

The RCC structure comprises of two materials viz. steel and concrete, hence the objective function for RCC structure can not be the weight because unit costs of theses two materials are different. Thus cost is considered as the objective function here which is given by

$$O(x) = V_C\, C_C + V_S\, C_S + A_f\ C_f \qquad \ldots (8.1)$$

where O(x) = objective function which is total cost of an element, V_C = volume of concrete and steel, W_S = weight of steel reinforcements, A_f = area of formwork and C_C, C_S, C_f = unit cost of concrete, steel and formwork respectively.

It is necessary to map the natural objective function to fitness function form in order to generate nonnegative values in all the cases and to reflect the relative fitness of the individual string.

8.5.3 Constraints

The design of beam and column is carried out based on IS: 456 (2000) [83] provisions for limit state design of structural components. The main constraints formulated are as follows.

I For Beams

- **Moment of resistance constraint**

  Constraint : $M_u \leq M_{ulim}$ ... (8.2)

  where M_u = Ultimate moment due to factored load obtained from elastic analysis, and M_{ulim} = Ultimate flexural strength of a section

  Constraint function: $g(x) = \max (Mu / M_{ulim} -1, 0)$ ... (8.3)

- **Maximum reinforcement area**

  Constraint : $A_s \leq A_{stmax}$ ... (8.4)

  where A_{stmax} = the maximum permissible tensile steel area and A_s = the steel area of the section.

  Constraint function: $g(x) = \max (A_s / A_{stmax} - 1, 0)$ ... (8.5)

- **Miminum reinforcement area**

  Constraint : $As \geq A_{stmin}$ (0.2 x (b x d)/100) ... (8.6)

  where A_{stmin} = the minimum permissible area of steel for a section and b and d = breath and depth of a beam section respectively.

  Constraint function: $g(x) = \max (A_{stmin} / A_s - 1, 0)$ ... (8.7)

II For Columns

- **Maximum and minimum reinforcement area**

  $A_{stmin} = 0.8\ (D_{XX}\ \text{x}\ D_{YY})/100$ ... (8.8)

$A_{stmax} = 6.0\ (D_{XX} \times D_{YY})/100$ ... (8.9)

Constraints : $A_s \leq A_{stmax}$ and $A_s \geq A_{stmin}$ ... (8.10)

Constraint functions: $g(x) = \max\ (A_s / A_{stmax} - 1, 0)$ and

$\max\ (A_{stmin} / A_s - 1, 0)$ ... (8.11)

where A_{stmin} = minimum area of steel, A_{stmax} = maximum area of steel, D_{XX} = size of column in x-x direction and D_{YY} = size of column in y-y direction. These constraints are handled by selecting proper values of upper bound and lower bound.

Constraint functions thus evaluated are added up to get overall constraint violation (C) which is then used to find the penalty function given as

$$PF = (1 + K\ C) \qquad \text{... (8.12)}$$

This penalty function is multiplied with the objective function (Eq. (8.1)) to get penalized objective function $O_p(x)$ and employed to find the fitness function which is formulated as

$$F(x) = 1 / (1 + O_p(x)\ ) \qquad \text{... (8.13)}$$

8.5.4 Example of RCC Plane Frame

An example of 2 bay 3 storey plane frame is considered here for the loading and support condition as shown in Fig. 8.2 for the following input data: Storey height = 3.5 m, bay width = 4 m, grade of steel = M15 and grade of concrete = Fe415. The Genetic data considered is as follows: population size = 20, maximum generations = 30, cross over probability = 0.78, mutation probability = 0.033 and length of substring for width and depth = 8. Number of design variables are 30 (i.e. 15 members x 2 variables for each member). The optimization is carried out with one point cross over for each substring, constant rate mutation and roulette wheel selection scheme. After supplying the input data, sub menu *Concrete Design* in the *Design* menu as shown in Fig. 8.2, is selected to start the design. Upon clicking this submenu the preliminary design for the given loading is carried out to decide the upper bound and lower bound values for dimensions. After this the actual search process starts and the program finds the optimum set of design variables. The final results obtained after 5 GA runs are displayed in Fig. 8.3 for beams and columns.

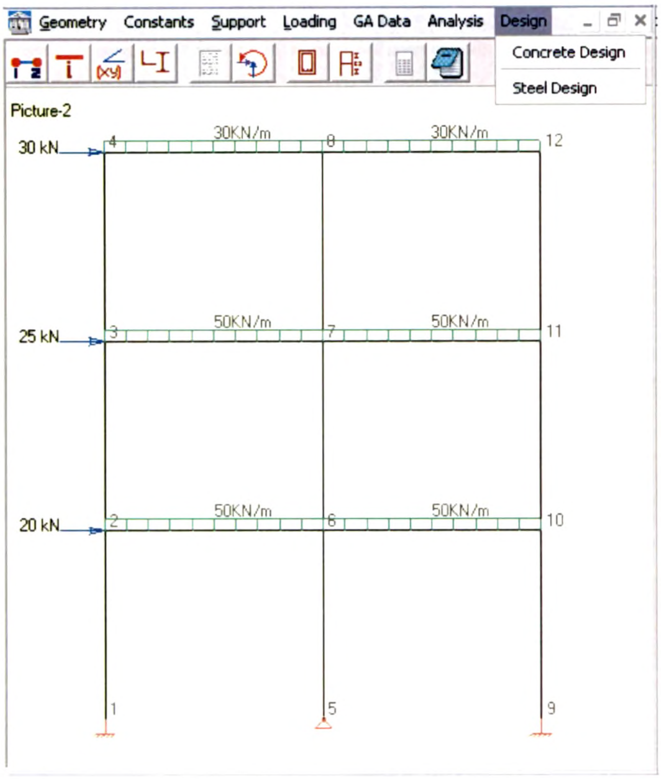
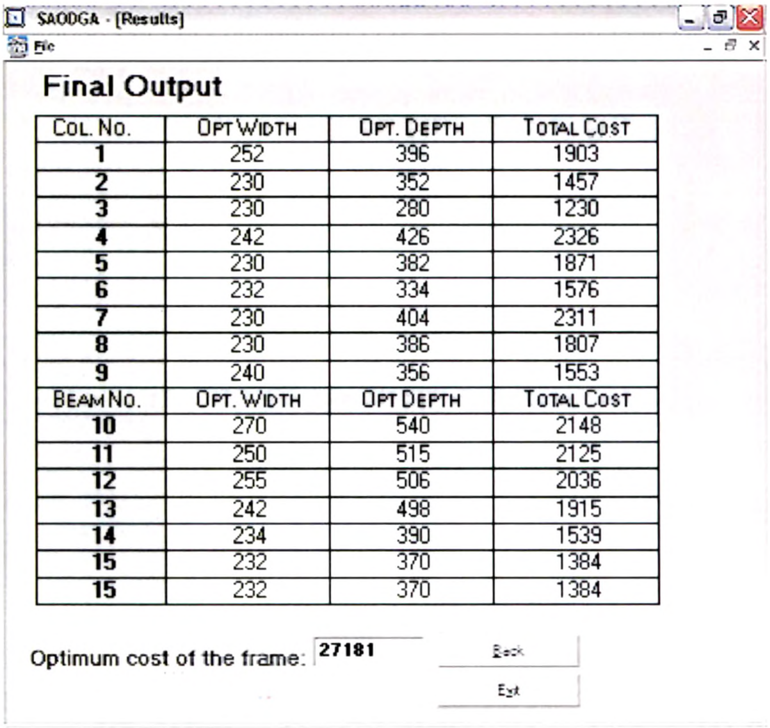


Fig. 8.2 Geometry and Loading



Final Output

COL. NO.	OPT WIDTH	OPT. DEPTH	TOTAL COST
1	252	396	1903
2	230	352	1457
3	230	280	1230
4	242	426	2326
5	230	382	1871
6	232	334	1576
7	230	404	2311
8	230	386	1807
9	240	356	1553
BEAM NO.	OPT. WIDTH	OPT DEPTH	TOTAL COST
10	270	540	2148
11	250	515	2125
12	255	506	2036
13	242	498	1915
14	234	390	1539
15	232	370	1384
15	232	370	1384

Optimum cost of the frame: 27181

Fig. 8.3 Dimensions and Cost after 5 GA Runs

8.6 Cost Optimization of RCC Water Tank

Circular water tanks are commonly used in water supply station and sewage treatment pumping stations, chemical processing and large underground sumps. The perimeter of a circular shape for a given area is the least than any other shapes. This makes it economical than any other shape. In the present section, an attempt is made to arrive at such a combination of height, diameter, wall thickness and reinforcement that the overall cost of the tank is minimum for a given capacity.

8.6.1 Objective Function

As the water tank is of R.C.C., the objective is to minimize the cost of the tank. Total cost comprises of the cost of concrete and cost of reinforcing steel. For small capacity tanks the formwork for circular tanks becomes costly and therefore it is also included in the objective function which can be written as

$$G(C) = Vc \times Cc + Ws \times Cs + Af \times Cf \qquad \text{... (8.14)}$$

where Vc = total volume of concrete in m^3, Cc = unit cost of concrete in Rs/m^3, Ws = total weight of reinforcing steel in Kg, Cs = unit cost of reinforcing steel in Rs/Kg, Af = total area of formwork and Cf = unit cost of formwork in Rs/m^2.

8.6.2 Constraint

The structural design of water tank includes only one major constraint i.e. the wall thickness of the side wall of the tank. As the water tanks are designed on the basis of theory of uncracked sections in concrete the wall thickness will be governed by the fact that the wall of the tank should be free from cracks and should be watertight. The constraint is formulated as follows:

$$t_{prov} \geq t_{req} \qquad \text{... (8.15)}$$

where t_{prov} = provided wall thickness in mm and t_{req} = required wall thickness in mm.

8.6.3 Penalty

If $t_{prov} < t_{req}$, a penalty is applied to the objective function for violating the constraint. The penalty (ψ_t) term may be formulated as below:

$$\psi_t = t_{req} \; / \; t_{prov} \qquad \text{... (8.16)}$$

The penalized objective function G1(C) is formulated as under:

$$G1(C) = G(C) \times \psi_t \qquad \ldots (8.17)$$

The above equation makes it clear that Ψ_t will always have a value greater than 1 and when multiplied with G(C), the numerical value of the objective function (cost) will increase. This will prevent that particular string from getting selected in the next generation.

8.6.4 Design Aspects of Circular R.C.C. Water Tanks

The present case deals with a circular R.C.C. water tank with a flexible joint at base i.e. the side wall is free to expand or contract as the joint between floor and side wall is flexible. A deflection curve of the side walls and pressure distribution is shown in Fig 8.4. The pressure at the base is $\gamma_\omega H$.

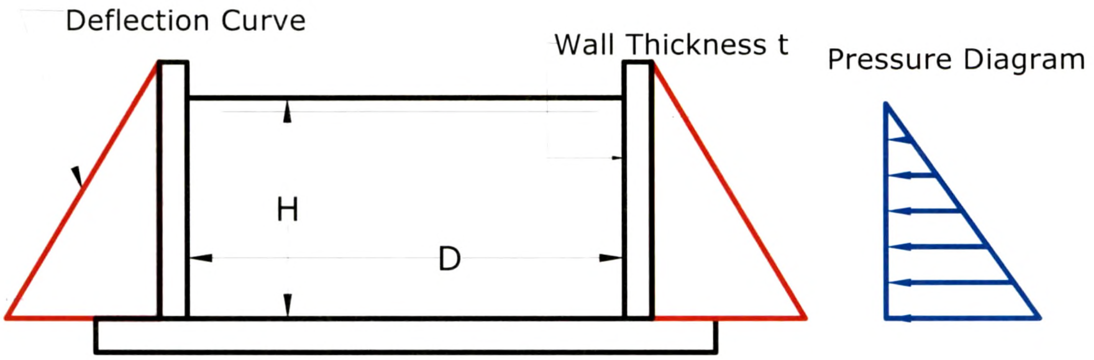
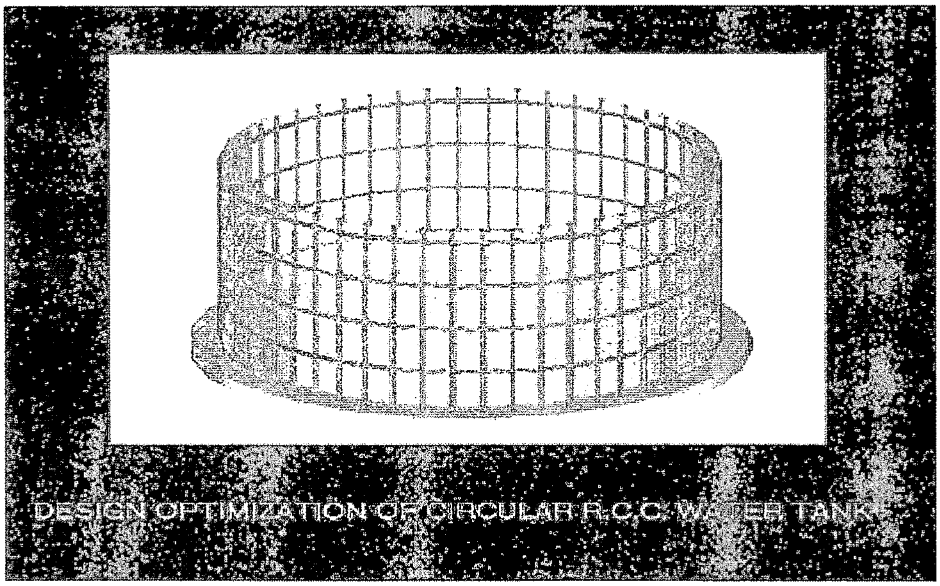


Fig 8.4 Deflection Curve and Pressure Distribution on Tank

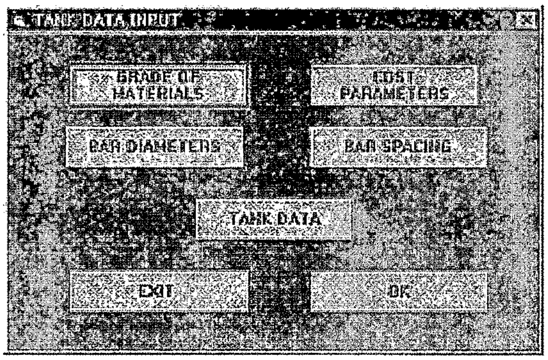
The section of the wall should be designed for hoop tension. As the hoop tension decreases from the base towards the top, the reinforcement may be reduced accordingly. Hoop steel is the main steel (horizontal) and is circular in shape. Vertical steel is the distribution steel. Guide lines for fixing the reinforcement, lapping of the bars and other for design of uncracked section are as per IS: 3370 [84]. The permissible stresses in steel for strength calculation, permissible stresses in concrete for resistance to cracking and minimum percentage of reinforcement for liquid retaining structures are taken from this standard code.

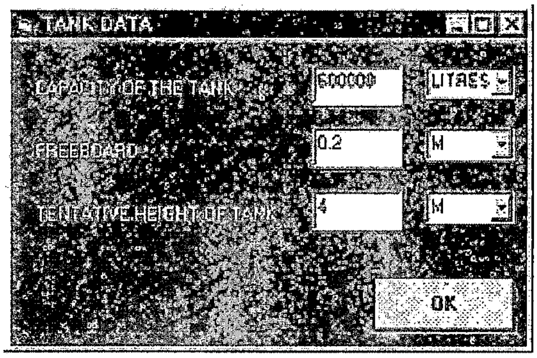
The step by step procedure for the use of the software is demonstrated here through screen shots.

1) Start up screen



2) Input the water tank data



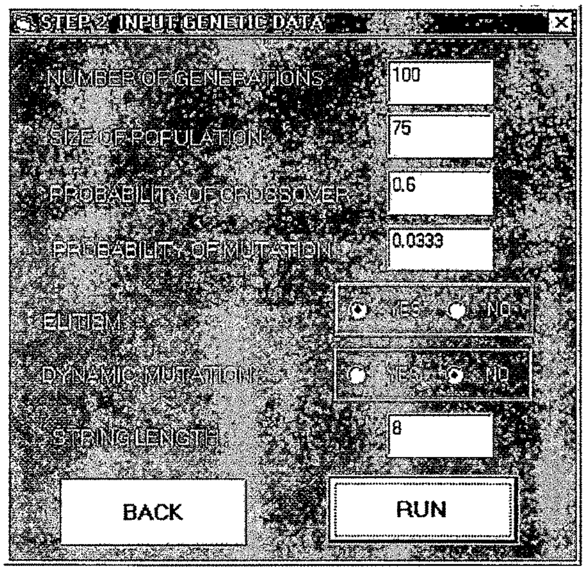


CAPACITY OF THE TANK 600000 LITRES

FREEBOARD 0.2 M

TENTATIVE HEIGHT OF TANK 4 M

3) Input the genetic parameter



NUMBER OF GENERATIONS 100

SIZE OF POPULATION 75

PROBABILITY OF CROSSOVER 0.6

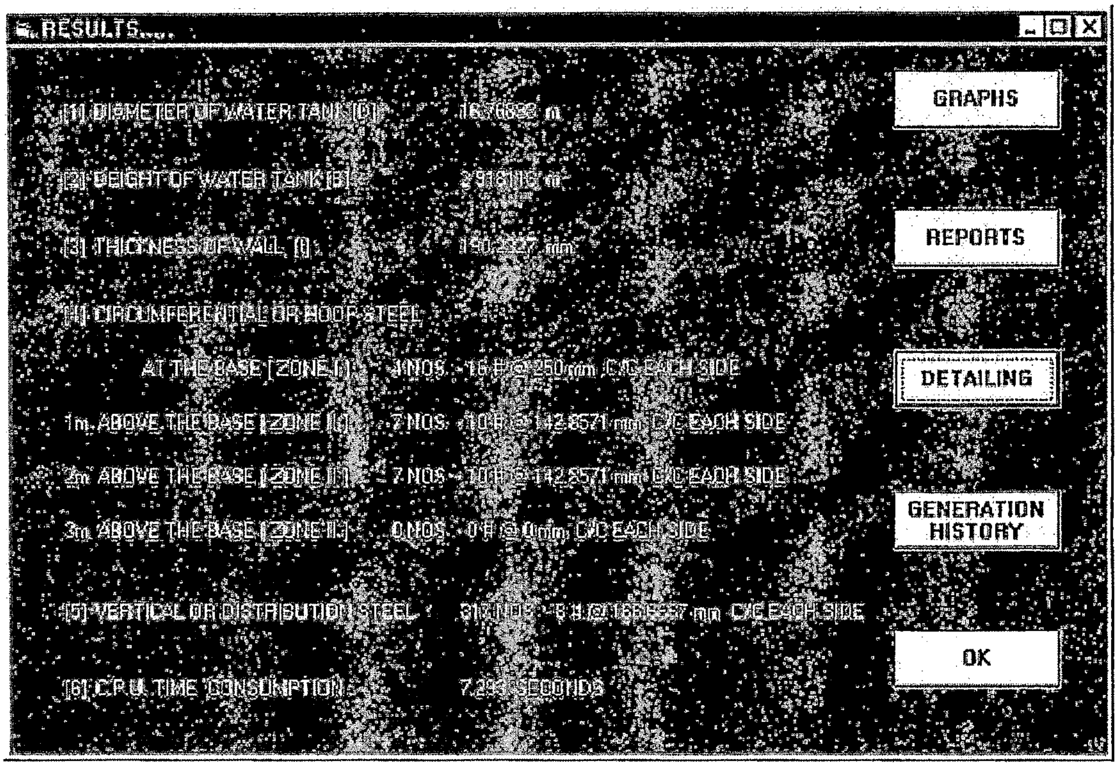
PROBABILITY OF MUTATION 0.0333

ELITISM ● YES ○ NO

DYNAMIC MUTATION ○ YES ● NO

STRING LENGTH 8

4) Run the software and view the results



RESULTS....

[1] DIAMETER OF WATER TANK [D] 16.76882 m

[2] HEIGHT OF WATER TANK [H] 2.918116 m

[3] THICKNESS OF WALL [t] 190.2327 mm

[4] CIRCUMFERENTIAL OR HOOP STEEL

AT THE BASE [ZONE I] 4 NOS - 16 # @ 250mm C/C EACH SIDE

1m ABOVE THE BASE [ZONE I] 7 NOS - 10 # @ 142.8571 mm C/C EACH SIDE

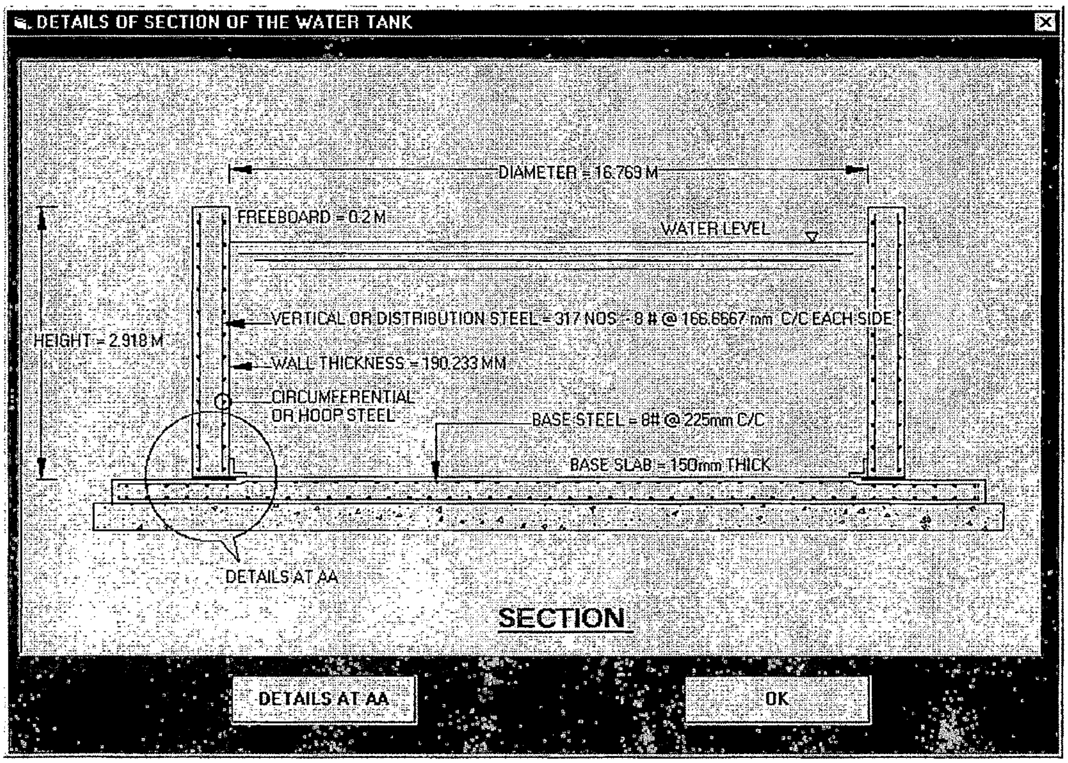
2m ABOVE THE BASE [ZONE II] 7 NOS - 10 # @ 142.8571 mm C/C EACH SIDE

3m ABOVE THE BASE [ZONE III] 0 NOS - 0 # @ 0mm C/C EACH SIDE

[5] VERTICAL OR DISTRIBUTION STEEL 317 NOS - 8 # @ 166.6667 mm C/C EACH SIDE

[6] C.P.U. TIME CONSUMPTION 7.293 SECONDS

5) View the reinforcement detailing of water tank



DETAILS OF SECTION OF THE WATER TANK

DIAMETER = 16.769 M
FREEBOARD = 0.2 M
WATER LEVEL
VERTICAL OR DISTRIBUTION STEEL = 317 NOS - 8 # @ 166.6667 mm C/C EACH SIDE
HEIGHT = 2.918 M
WALL THICKNESS = 190.233 MM
CIRCUMFERENTIAL OR HOOP STEEL
BASE STEEL = 8# @ 225mm C/C
BASE SLAB = 150mm THICK
DETAILS AT AA
SECTION
DETAILS AT AA
OK

8.6.5 Design Example

For the following data: Capacity of tank = 600,000 litres, Grade of concrete = M25, Grade of steel = Fe415 and tank is resting on ground, the lower and upper bound values for thickness are taken as 150 and 300 mm and those for height of the tank are 2.5 m and 5.5 m. Printable output along with other output generated by the software are given below:

```
  INPUT DATA
 ========
   # GENETIC DATA
     ------------------
     [1] GENERATIONS =  100
     [2] POPULATION SIZE = 75
     [3] PROBABILITY OF CROSSOVER = 0.6
     [4] PROBABILITY OF MUTATION = 0.0333
     [5] STRING LENGTH = 8

   # WATER TANK DATA
     ----------------------
     [1] CAPACITY = 600  m^3
     [2] FREEBOARD = 0.2  m
     [3] GRADE OF CONCRETE = M-25
     [4] GRADE OF STEEL = Fe-415
     [5] BAR DIAMETERS = 8 # TO  32 #
     [6] BARSPACING = 100  mm  TO  250  mm
     [7] TENTATIVE HEIGHT OF TANK = 4 m
     [8] CONCRETE COVER = 50 mm
     [9] UNIT COST OF CONCRETE = 1750 Rs/m^3
    [10] UNIT COST OF STEEL = 20 Rs/kg
    [11] UNIT COST OF FORMWORK = 30 Rs/m^2

 OUTPUT RESULTS
 ===========
     [1] DIAMETER OF THE TANK = 16.76898  m
     [2] HEIGHT OF THE TANK = 2.918118  m
     [3] WALL THICKNESS OF THE TANK =190.2327  mm
     [4] CIRCUMFERENTIAL OR HOOP STEEL
              AT BASE OF THE TANK [ZONE I]  : 4 NOS  - 16 # @ 250 mm  C/C EACH SIDE
              AT 1m ABOVE THE BASE [ZONE II] : 7 NOS  - 10 # @ 142.8571 mm  C/C EACH SIDE
              AT 2m ABOVE THE BASE [ZONE III] : 7 NOS  - 10 # @ 142.8571 mm  C/C EACH SIDE
              AT 3m ABOVE THE BASE [ZONE IV] : 0 NOS  - 0 # @ 0 mm  C/C EACH SIDE
     [5] VERTICAL STEEL OR DISTRIBUTION STEEL : 315 NOS  - 8 # @ 166.6667 mm  C/C EACH SIDE
     [6] TOTAL COST OF THE WATER TANK = 165089.8 Rs
     [7] C.P.U. TIME CONSUMPTION = 7.098 SECONDS
```

Figure 8.5 shows convergence of the developed algorithm towards optimum solution.

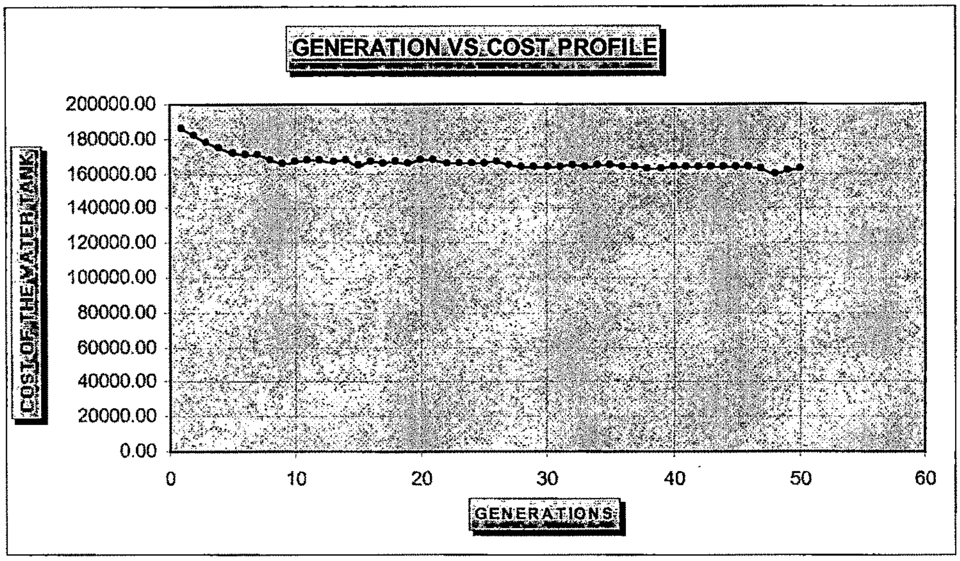
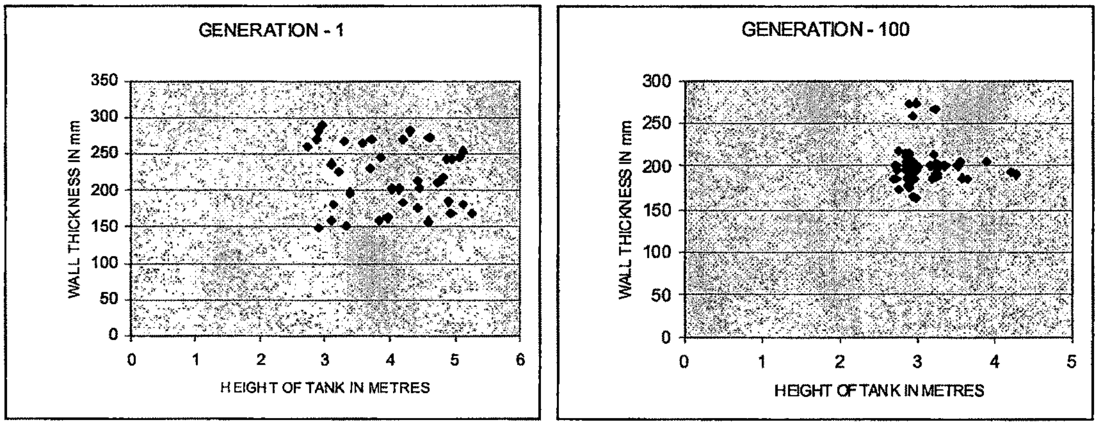


Fig. 8.5 Minimization of Cost during Search



(a) Status of Generation - 1 **(b) Status of Generation - 100**

Fig. 8.6 Graphs Showing Progressive Improvement in the Solution

Figures 8.6 (a) and (b) displays concentration of the solutions near the region of optimal solution. A comparison of result is given in Table 8.1.

Table 8.1 Result Comparison of Water Tank

Parameter	Results By Present Software	Result By R.C.C. Text book [85]
Diameter of tank	16.77 m	12.62 m
Height of tank	2.92 m	5 m
Wall thickness	190.23 mm	235.2 mm
Total cost of the tank	Rs 1,65,090 /-	Rs 1,92,606 /-

8.7 Cost Optimization of Isolated Footing

The importance of a footing in any structure is due to the fact that performance of the whole structure is influenced by the strength of the foundation. Looking to the importance of a foundation it is evident that it is one of the major contributors in overall cost of the structure. Thus, even a small amount of saving in the cost of a single footing can cause a heavy impact on the total cost of the structure. The major cost of the footing involves cost of concrete and cost of reinforcing steel.

8.7.1 Objective Function

The objective function for the isolated R.C.C. footing can be formulated as follows:

$$CT(x) = Wrs \times Crs + Vc \times Cc + Af \times Cf \qquad \dots (8.18)$$

where CT(x) = objective function i.e. total cost of the footing in Rs, Wrs = total weight of the reinforcing steel in Kg, Crs = cost of the reinforcing steel per unit weight in Rs/Kg, Vc = total volume of concrete in the footing in m^3, Cc = cost of concrete per unit volume in Rs/m^3, Af = area of formwork in m^2 and Cf = cost of the formwork per unit area in Rs/m^2.

8.7.2 Variables

The design variables for optimization of an isolated footing are the length, width and depth of the footing. The idea is to arrive at such a combination of length, width, depth and reinforcement area that the overall cost of the footing is minimum and at the same time, the footing is safe from structural design point of view. The binary string representation scheme is used for all the variables continuous in nature. The user can select any string length depending on the accuracy required.

Each potential solution is represented by a single binary string called the “Main string”, which is then divided into three smaller strings each representing a design variable i.e. “length string”, “width string” and the “depth string”. The binary strings are then converted into their decimal equivalents and are mapped between upper and lower bounds to obtain the values of the variables. The procedure is illustrated in Fig. 8.7.

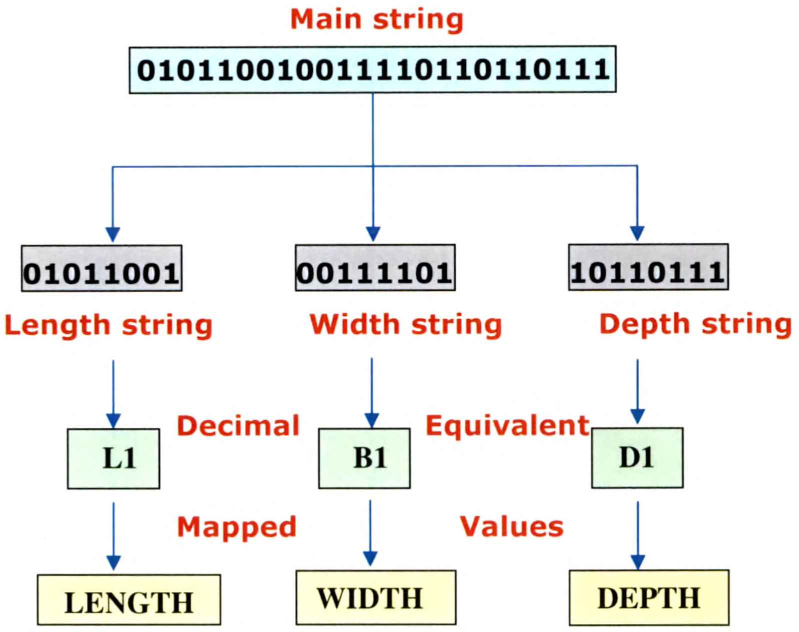


Fig. 8.7 Binary Representation of the Variables

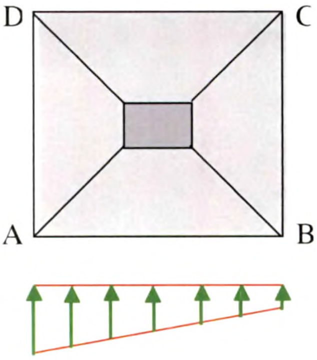
8.7.3 Constraints

Safety is of prime importance in any structural design. Thus, while optimizing any structural component there should be no compromise with safety. This requires fulfillment of certain condition and constraints, violation of which would make the structure unsafe. In the present case, a penalty approach is used for solutions that violate a constraint. The objective function of these solutions is penalized suitably to prevent occurrence of this solution string in the further generations. The following constraints and their corresponding penalties have been used in the design.

❖ **S.B.C. of soil:** The net upward pressure below the base of the foundation should be less than or equal to S.B.C. of soil to prevent excessive settlement and failure. Thus, pressure below any corner of footing should not be greater than the S.B.C. of soil. This may be represented as follows:

Constraints :

$$\begin{aligned} P_A &\leq \text{S.B.C.} \\ P_B &\leq \text{S.B.C} \\ P_C &\leq \text{S.B.C.} \\ P_D &\leq \text{S.B.C.} \end{aligned} \qquad \ldots (8.19)$$



Penalties : $\Phi_A = (P_A - (SBC / 2)) / (SBC / 2)$

$\Phi_B = (P_B - (SBC / 2)) / (SBC / 2)$

$\Phi_C = (P_C - (SBC / 2)) / (SBC / 2)$

$\Phi_D = (P_D - (SBC / 2)) / (SBC / 2)$ … (8.20)

where P_A, P_B, P_C and P_D = the pressures below corners A, B, C and D of the footing respectively, S.B.C. = safe bearing capacity of soil and Φ_A, Φ_B, Φ_C, Φ_D = penalties for pressures below corner A, B, C and D of the footing respectively.

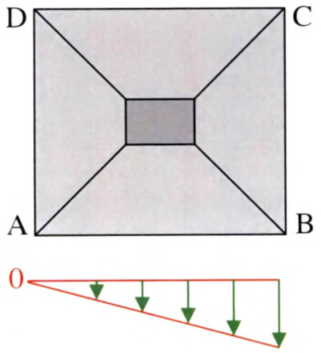
❖ **No tension condition below the base of footing:** Ideally, footings are designed for no-tension condition, which requires that pressure below any corner of the footing should not be less than zero (i.e. no negative pressure below the base). This may be represented as follows:

Constraints : $P_A \geq 0$

$P_B \geq 0$

$P_C \geq 0$

$P_D \geq 0$ … (8.21)



Penalties : $\Psi_A = (\text{Abs}(P_A) + (S.B.C. / 2)) / (S.B.C. / 2)$

$\Psi_B = (\text{Abs}(P_B) + (S.B.C. / 2)) / (S.B.C. / 2)$

$\Psi_C = (\text{Abs}(P_C) + (S.B.C. / 2)) / (S.B.C. / 2)$

$\Psi_D = (\text{Abs}(P_D) + (S.B.C. / 2)) / (S.B.C. / 2)$ … (8.22)

where Ψ_A, $\Psi_{B,}$ $\Psi_{C,}$ and Ψ_D = penalties for negative pressure below the base of the footing for corners A, B, C and D respectively.

❖ **Flexure check:** The depth selected for the footing must satisfy the flexure check i.e. the depth should be greater than,

$$d_{req} = \sqrt{(M_{max} / Q.b)}$$ … (8.23)

Constraint : $d_{req} \leq$ dprovided … (8.24)

Penalty : $d_{req} / d_{provided}$ ... (8.25)

where M_{max} = maximum moment from Mux and Muy, Q = material constant, b = unit width of footing, d_{req} = depth required for safety in flexure and $d_{provided}$ = provided depth of the footing.

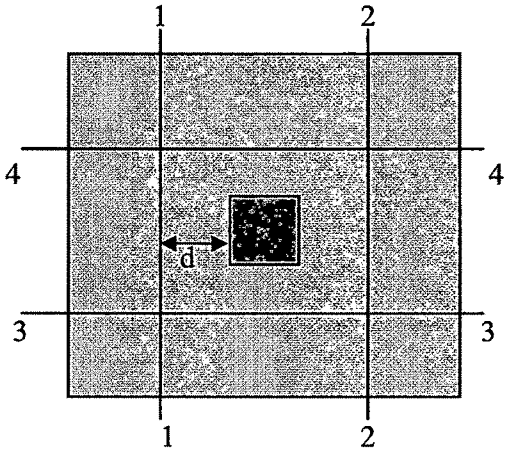
❖ **One way shear check:** The footing should be safe in one way shear check applied at a distance "d" from the face of the column where "d" is effective depth of footing. As the pressure distribution below the base is non-uniform, one way shear checks are applied at section 1-1, 2-2, 3-3 and 4-4 i.e. on all the four sides of the column.

Constraints : $\zeta v_1 \leq \zeta c$

$\zeta v_2 \leq \zeta c$

$\zeta v_3 \leq \zeta c$

$\zeta v_4 \leq \zeta c$ ... (8.26)



Penalties : $\epsilon_1 = \zeta v_1 / \zeta c$

$\epsilon_2 = \zeta v_2 / \zeta c$

$\epsilon_3 = \zeta v_3 / \zeta c$

$\epsilon_4 = \zeta v_4 / \zeta c$ ... (8.27)

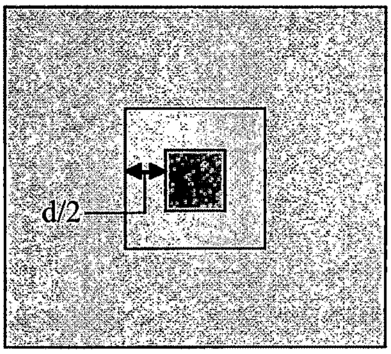
where ϵ_1, ϵ_2, ϵ_3 and ϵ_4 = penalties for 1-way shear check at sections 1-1, 2-2, 3-3 and 4-4 respectively, ζv_1, ζv_2, ζv_3 and ζv_4 = vertical shear stress at sections 1-1, 2-2, 3-3 and 4-4 respectively and ζc = shear strength of concrete depending on percentage reinforcing steel provided.

❖ **Two Way Shear Check**

The footing should be safe in two way shear check applied at a distance d/2 from the column face, where d= effective depth of footing.

Constraint : $\zeta v \leq \zeta c$ ... (8.28)

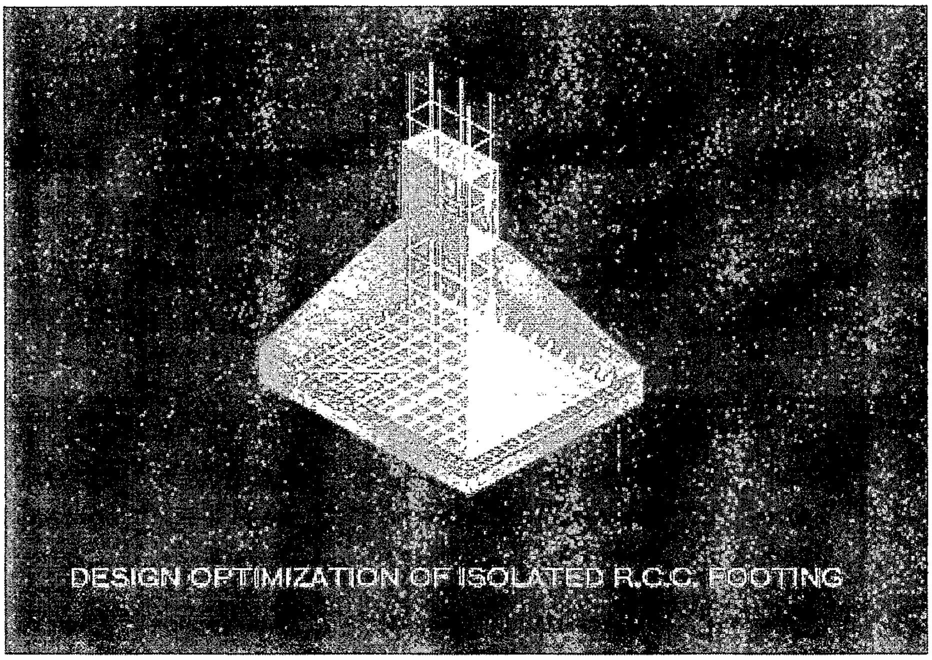
Penalty : $\epsilon = \zeta v / \zeta c$ ... (8.29)



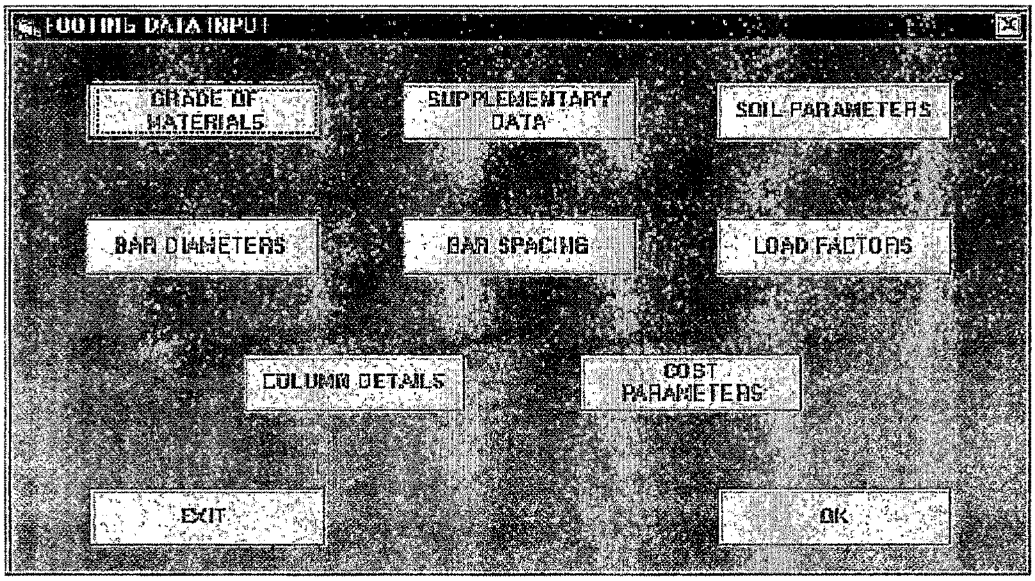
where ζv = vertical shear stress at a section at a distance of d/2 from the face of the column and Є = penalty for two way shear check.

8.7.4 Steps to Use the Software and Screenshots

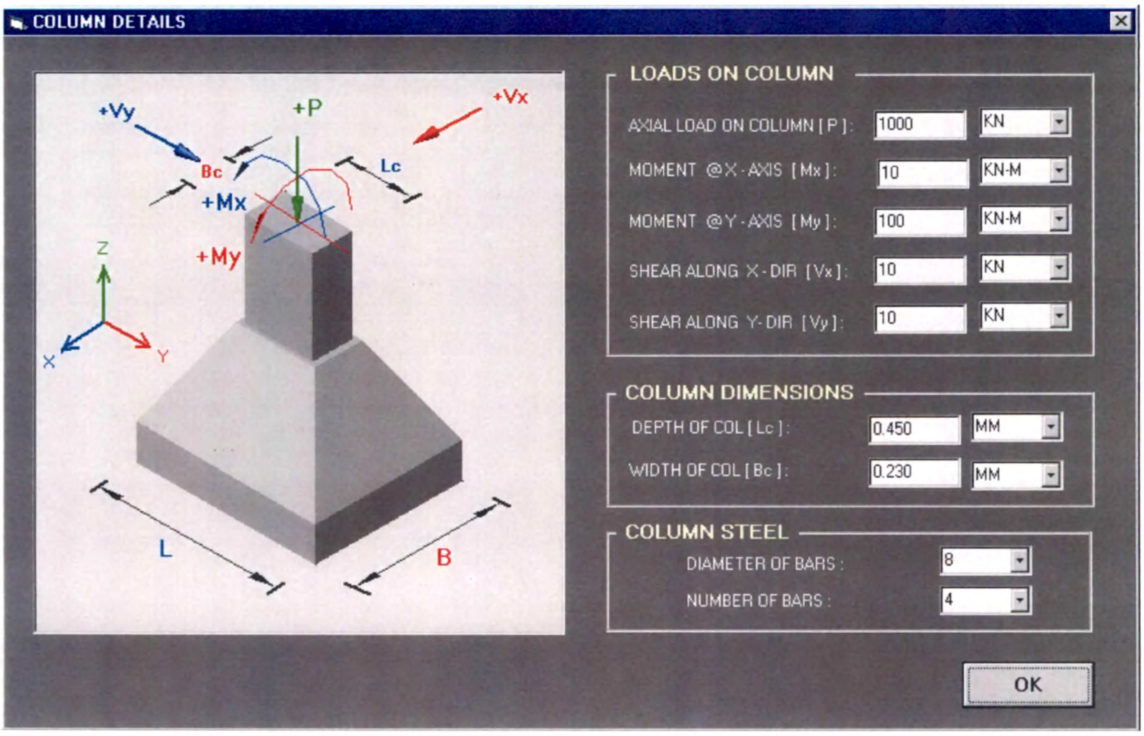
1) Start up screen



2) Supply the footing data



3) Enter loading data



COLUMN DETAILS

LOADS ON COLUMN

AXIAL LOAD ON COLUMN [P]: 1000 KN

MOMENT @X-AXIS [Mx]: 10 KN-M

MOMENT @Y-AXIS [My]: 100 KN-M

SHEAR ALONG X-DIR [Vx]: 10 KN

SHEAR ALONG Y-DIR [Vy]: 10 KN

COLUMN DIMENSIONS

DEPTH OF COL [Lc]: 0.450 MM

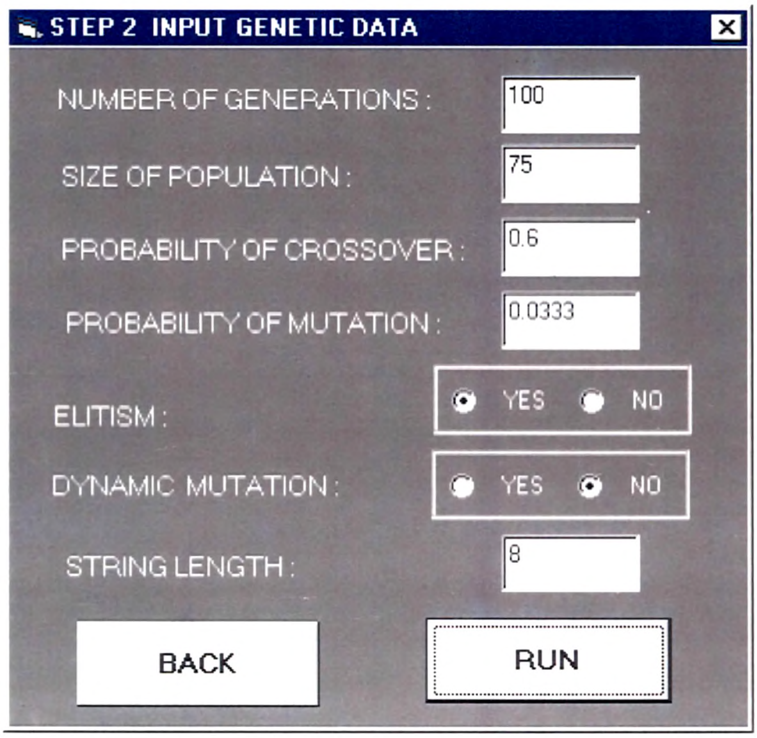
WIDTH OF COL [Bc]: 0.230 MM

COLUMN STEEL

DIAMETER OF BARS : 8

NUMBER OF BARS : 4

4) Supply the genetic data



STEP 2 INPUT GENETIC DATA

NUMBER OF GENERATIONS : 100

SIZE OF POPULATION : 75

PROBABILITY OF CROSSOVER : 0.6

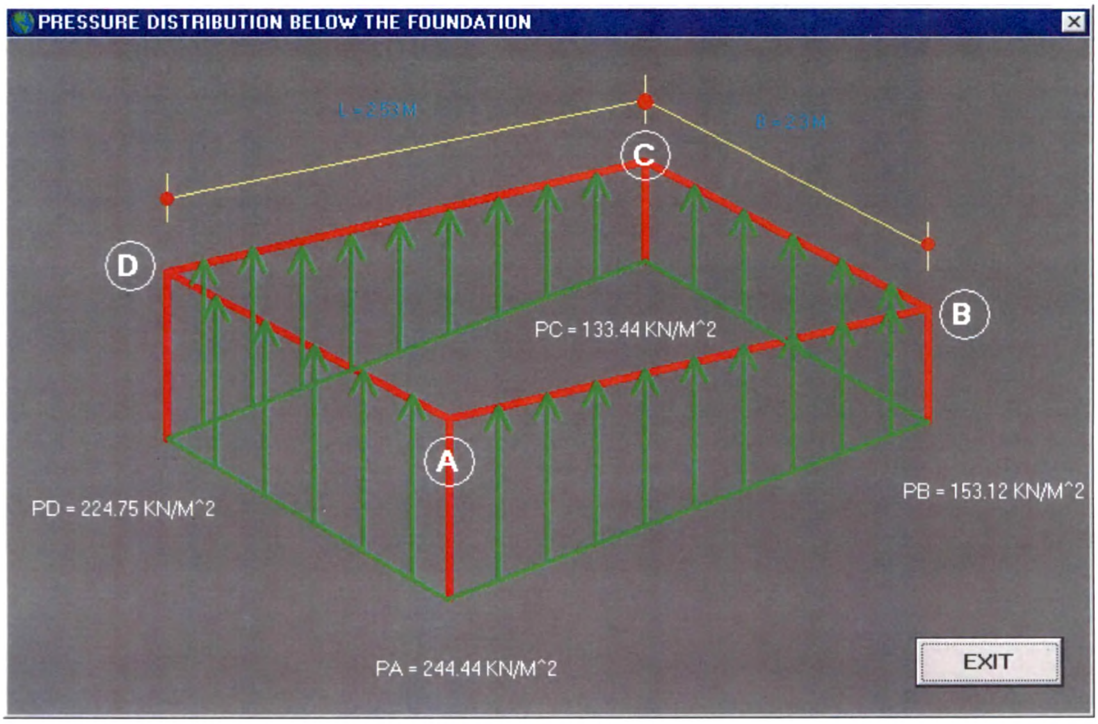
PROBABILITY OF MUTATION : 0.0333

ELITISM : ● YES ○ NO

DYNAMIC MUTATION : ○ YES ● NO

STRING LENGTH : 8

7) View the base pressures



8.7.5 Design Example

Data:

(1) Column size (mm)	: 250 × 750	(5) Column reinforcement	: 10 # 25 mm
(2) Axial load (kN)	: 2000	(6) Grade of steel	: Fe-415
(3) Column concrete grade	: M-30	(7) Footing concrete grade	: M-15
(4) SBC at 2.0 m (kN/m^2)	: 220		

<u>Printable Output Generated by the Software</u>

```
 INPUT DATA
 ========

   # GENETIC DATA
  ---------------------
[1] GENERATIONS = 100                      [4] PROBABILITY OF MUTATION = 0.0333
[2] POPULATION SIZE = 75                   [5] STRING LENGTH = 8
[3] PROBABILITY OF CROSSOVER = 0.9

 # FOOTING DATA
  --------------------
[1] LOAD ON COLUMN = 2000 KN               [14] MAXIMUM SPACING OF BARS = 250 mm
[2] MOMENT ABOUT X-AXIS =0 KN-m            [15] MINIMUM SPACING OF BARS = 100 mm
```

```
[3] MOMENT ABOUT Y-AXIS =0 KN-m
[4] SHEAR ALONG X-AXIS =0 KN
[5] SHEAR ALONG Y-AXIS =0 KN
[6] LENGTH OF COLUMN = 0.75 m
[7] WIDTH OF COLUMN = 0.25 m
[8] DIAMETER OF COLUMN BARS =25 mm
[9] NUMBER OF COLUMN BARS = 10 NOS
[10] S.B.C. OF SOIL =220 KN/M2
[11] DEPTH BELOW GROUND LEVEL = 2 m
[12] MINIMUM DIAMETER OF BAR = 8# mm
[13] MAXIMUM DIAMETER OF BAR =32# mm
```

```
[16] GRADE OF CONCRETE = M-15
[17] GRADE OF STEEL = FE-415
[18] EDGE DEPTH = 0.23 m
[19] BASE DEPTH =150 cm
[20] BOTTOM OFFSET  =150 cm
[21] BENCHING = 50 mm
[22] CONCRETE COVER = 50 MM
[23] UNIT COST OF CONCRETE = 1750 RS/M3
[24] UNIT COST OF STEEL = 20 Rs/ kg
[25] UNIT COST OF FORMWORK = 30 Rs/m2
```

```
OUTPUT RESULTS
===========
[1]  LENGTH OF FOOTING = 3.173102 m
[2]  WIDTH OF FOOTING = 3.151745 m
[3]  DEPTH OF FOOTING = 1.308422 m
[4]  STEEL PARALLEL TO LONGER DIRECTION
       = 17 NOS - 12 # @ 185.7354 mm
```

```
[5]  STEEL PARALLEL TO SHORTER DIRECTION
       = 24 NOS - 12 # @ 151.4047 mm
[6]  TOTAL COST OF FOOTING = 18466.09 Rs
[7]  C.P.U. TIME = 8.844 SECOND
```

Graphs showing progressive improvement in the solution

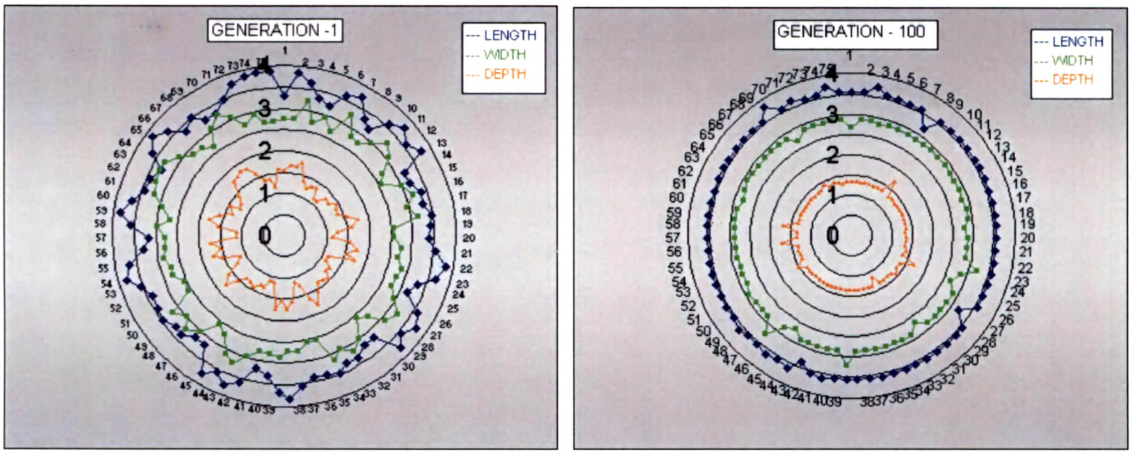


Table 8.2 Comparison of Result

Parameter	Present Software	R.C.C. Textbook [85]
Length of the footing	3.17 m	3.45 m
Width of the footing	3.15 m	2.95 m
Depth of the footing	1.31 m	1.2 m
Steel parallel to longer direction	17 nos –12 #	20 nos -12 #
Steel parallel to shorter direction	24 nos – 12 #	24 nos -12 #
Total cost of the footing	Rs. 18,467 /-	Rs. 19,789 /-

8.8 COST OPTIMIZATION OF COMBINED FOOTING

8.8.1 Design Variables

As the cost of foundation depends on volume of concrete and weight of reinforcing bars, seven design variables considered are length, width and depth of foundation block, longitudinal reinforcement areas for negative moments under two columns and positive moment between two columns and reinforcement area parallel to width for transverse bending as depicted in Fig. 8.8.

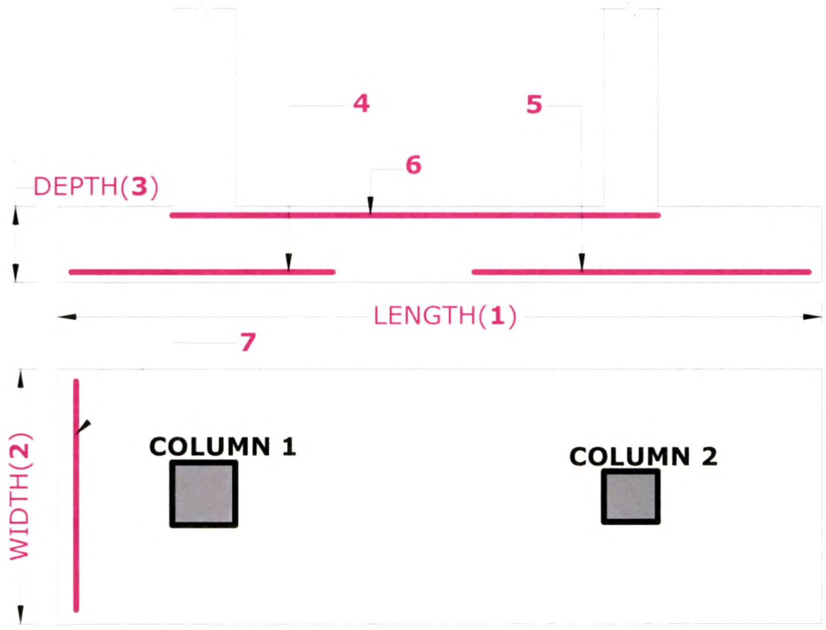


Fig. 8.8 Combined footing showing design variables

Genetic search space is defined by selecting the upper and lower bound values of these variables. The developed algorithm selects these values by performing preliminary design based on data supplied by the user.

8.8.2 Constraints

All the structural engineering optimization problems are constrained optimization problems. Following constraints that govern the optimization process are imposed on a solution as per IS 456 specifications for limit state method of design [83].

(i) $P_s \leq SBC$ (ii) $P_s \geq 0$ (iii) $p_t \geq p_{tr}$

(iv) $\tau_{avg,ss} \leq \tau_{c,ss}$ (v) $\tau_{avg,ds} \leq \tau_{c,ds}$ … (8.30) where P_s is the soil pressure under the footing, SBC is the safe bearing capacity of soil below foundation, p_t is percentage reinforcement in the solution string, p_{tr} is the reinforcement percentage required for given bending moment and dimensions of footing in the solution string selected by GA, $\tau_{avg,ss}$ and $\tau_{c,ss}$ are average and permissible shear stresses for one way shear and $\tau_{avg,ds}$ and $\tau_{c,ds}$ are for two way shear.

The corresponding constraint functions to evaluate penalty are:

(i) $g_i(x) = \max((P_s / SBC - 1), 0)$ (ii) $g_i(x) = ABS(\min((P_s / SBC), 0))$

(iii) $g_i(x) = \max((P_{tr} / P_t - 1), 0)$ (iv) $g_i(x) = \max((\tau_{avg,ss} / \tau_{c,ss} - 1), 0)$

(iii) $g_i(x) = \max((\tau_{avg,ds} / \tau_{c,ds} - 1), 0)$ ... (8.31)

Total constraint violation, $C = \sum_{i=1}^{N_c} g_i(x)$, where N_c is number of constraints. ... (8.32)

The penalty function can be obtained as: $P(x) = 1 + KC$, which is then multiplied with objective function, $O(x)$ to get penalized objective function $O_p(x)$. This objective function is used in the calculation of fitness function f(x) given in Eq. 8.34.

8.8.3 Objective function

Objective function, O(x) is the function of design variables which is to be minimized or maximized. Cost of the combined footing is taken as objective function in the work presented here and is expressed as:

$$O(x) = V_c U_c + W_s U_s + A_f U_f \quad \text{... (8.33)}$$

where V_c is volume of concrete in m^3, U_c is unit cost of concrete per m^3, W_s is weight of steel reinforcement in kg, U_s is unit cost of steel per kg, A_f is area of formwork in m^2 and U_f is unit cost of formwork per m^2.

8.8.4 Fitness function

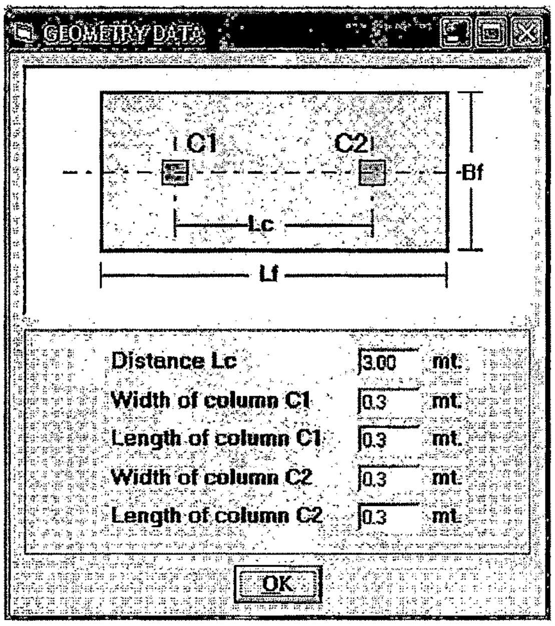
The fitness function chosen for this example is

$$f(x) = \frac{C_{max} - O_P(x)}{C_{max}} \quad \text{... (8.34)}$$

where C_{max} is sufficiently large value of objective function which is selected judiciously.

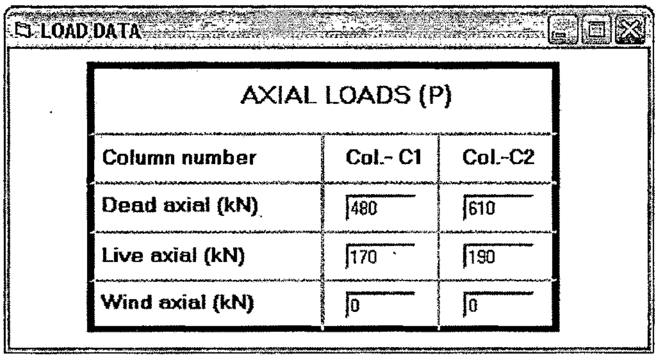
8.8.5 Software development

In the pre-processor various forms provide entry of input data required for the design of footing such as distance between two columns, dimensions of columns, etc as shown in Fig. 8.9. The load data is supplied through form displayed in Fig. 8.10. It also provides separate forms for entering GA parameters, material property, unit cost of materials and soil property. The main processor generates initial random solutions, decodes them to find actual variables, analyses the footing for calculation of maximum shear force and bending moment for three load cases given below and carries out various GA related calculations to arrive at the optimum solution.



Distance Lc	3.00	mt.
Width of column C1	0.3	mt.
Length of column C1	0.3	mt.
Width of column C2	0.3	mt.
Length of column C2	0.3	mt.

Fig. 8.9 Columns Data

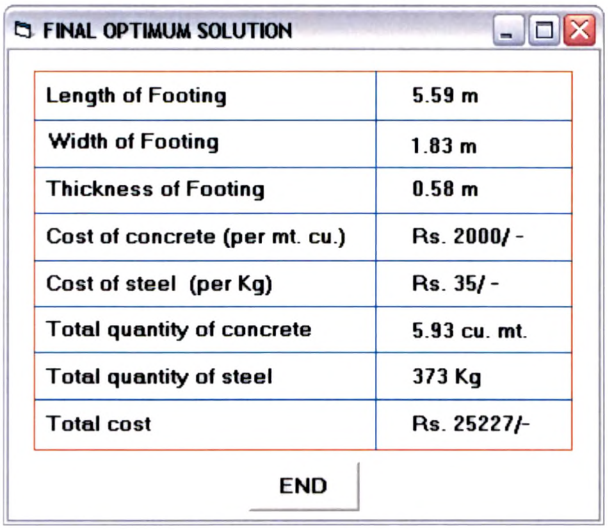


AXIAL LOADS (P)

Column number	Col.- C1	Col.-C2
Dead axial (kN)	480	610
Live axial (kN)	170	190
Wind axial (kN)	0	0

Fig. 8.10 Form for Entering Load

The post processor shows the final solution in a tabular form indicating the cost of the concrete, cost of the reinforcement and total cost of the footing as in Fig. 8.11. It also shows the reinforcement detail as shown in Fig. 8.12.



FINAL OPTIMUM SOLUTION

Length of Footing	5.59 m
Width of Footing	1.83 m
Thickness of Footing	0.58 m
Cost of concrete (per mt. cu.)	Rs. 2000/ -
Cost of steel (per Kg)	Rs. 35/ -
Total quantity of concrete	5.93 cu. mt.
Total quantity of steel	373 Kg
Total cost	Rs. 25227/-

END

Fig. 8.11 Optimum Design Results for Combined Footing

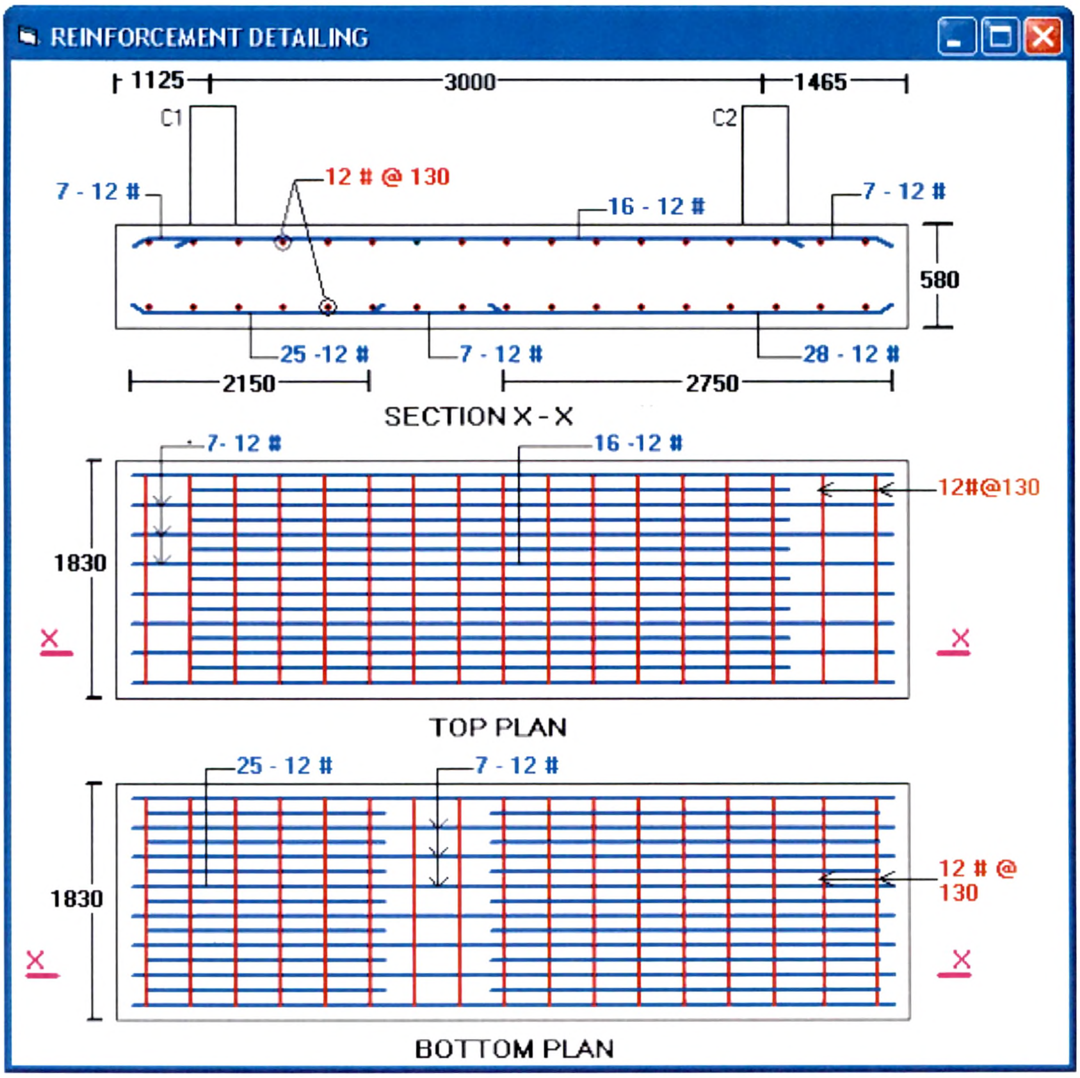


Fig. 8.12 Reinforcement Detailing for Combined Footing

8.8.6 Example of Combined footing

Data [85]: C/C distance between columns = 3.0 m, Size of column C1 = 0.3 × 0.3 m, Size of column C2 = 0.3 × 0.3 m, Grade of concrete = M20, Grade of steel = Fe415, DL + LL on column C1 = 480 +170 kN, DL + LL on column C2 = 610 + 190 kN and S.B.C. of soil = 175 kN/m^2.

Initial data required to carry out preliminary design of foundation is supplied through various forms developed in the software. Every solution string, which is composed of seven design variables, is checked for the following three loading conditions as per IS: 456 [83]:

(i) C1 – DL +LL and C2 – DL + LL (ii) C1 – DL and C2 – DL + LL
(iii) C1 – DL +LL and C2 – DL

Maximum values of shear force and bending moments induced due to three loading conditions are calculated and the foundation is designed for the maximum values among the three cases.

Population size and maximum number of generation of 30 and 20 are selected respectively for GA search. Length of string for every variable is 8. The probabilities of crossover and mutation adopted are 0.9 and 0.05 respectively. Final results obtained after three GA runs are shown in Figures 8.11 and 8.12 in tabular form and graphical form respectively. Table 8.3 shows the economy achieved through the program in comparison with the available results.

Table 8.3 Comparison of Results for Combined Footing

Item	Reference [85]	FL [§ 9.8.8]	Obtained results
Length of footing	5.1 m	5.1 m	5.59 m
Width of footing	2.0 m	2.0 m	1.83 m
Depth of footing	0.67 m	0.62 m	0.58 m
Total quantity of concrete	6.83 m^3	6.32 m^3	5.93 m^3
Total quantity of steel	460.40 kg	421.00 kg	373.00 kg
Total cost of concrete	Rs. 13660/-	Rs. 12640/-	Rs. 11860/-
Total cost of steel	Rs. 16114/-	Rs. 14735/-	Rs. 13055/-
Total Cost	Rs. 29774/-	Rs. 27375/-	Rs. 24915/-

The convergence achieved in one of the GA runs using tournament selection scheme is depicted in Fig. 8.13.

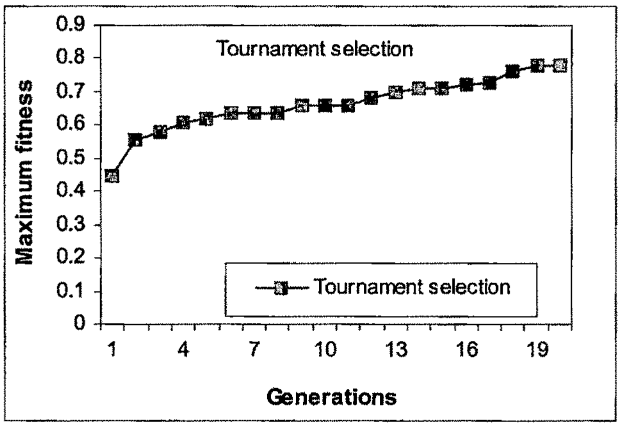


Fig. 8.13 Generation History of Maximum Generation

8.9 COST OPTIMIZATION OF RETAINING WALL

Design optimization of retaining wall is attempted here. Four possible modes of failure considered are: (i) overturning about the toe, (ii) sliding on the base of the footing, (iii) maximum soil pressure under the base exceeding SBC of soil and (iv) minimum soil pressure being tensile in nature. To prevent structural failure of component parts, they are designed to have sufficient strength against anticipated loads.

8.9.1 Design Variables

Given a height of the earth to be retained the design variables that affect the cost of retaining wall are: footing width, stem thickness at the base, toe width and footing thickness. These variables are allowed to vary continuously over a specified range which decides the extent of the search space. The proposed ranges are: footing width - 0.48 h_w to 0.6 $h_{w,}$ stem thickness (at the top) – 0.06 h_w to 0.12 $h_{w,}$ toe width – 0.15 h_w to 0.2 h_w and footing thickness – 0.06 h_w to 0.1 h_w, where h_w is height of the wall. All variables are represented using a binary string of 8 bits each. The solution string thus becomes 32 bits long.

8.9.2 Constraints and Penalties

- **Overturning about the toe**

  To prevent the over turning the stabilizing moment must be sufficiently in excess of the overturning moment so that the factor of safety against overturning is never less than 1.5 and should preferably be 2.0 or more.

  Constraint : $FOS_{ot(act)} \geq 1.5$ ... (8.35)

Penalty : $\Phi_1 = (1 + (FOS_{ot(per)} - FOS_{ot(act)}) / FOS_{ot(per)})$ ... (8.36)

- **Sliding of the retaining wall**

  The factor of safety against sliding should never be less than 1.5 and should preferably be 2.0 or more.

  Constraint : $FOS_{s(act)} \geq 1.5$ ... (8.37)

  Penalty : $\Phi_2 = (1 + (FOS_{s(per)} - FOS_{s(act)}) / FOS_{s(per)})$ ... (8.38)

- **Maximum soil pressure**

  The maximum pressure due to all possible loading on the wall should be less than safe bearing capacity of soil to prevent the failure of soil resulting in settlement of wall.

  Constraint : $p_{max} \leq SBC$ ... (8.39)

  Penalty : $\Phi_3 = (1 + ((p_{max} - SBC) / SBC))$ ... (8.40)

- **Negative soil pressure below the footing**

  This constraint is imposed to prevent negative soil pressure belw the wall resulting in over turning of the wall.

  Constraint : $p_{min} \geq 0$ ... (8.41)

  Penalty : $\Phi_4 = (1 + abs\ (p_{min}))$ ... (8.42)

Besides, the design considerations as per IS: 456 [83] have been complied with. Though GA is most suitable for unconstrained optimization problems, these can be easily handled using the penalty approach. In this approach a penalty is applied to infeasible solutions in proportion to the degree of constraint violation.

The general form of the penalty function is:

$P_k = \Pi\ (1 + vkj / vk)^{nk}$ ... (8.43)

Where P_k = Penalty incurred for violating constraint k, Vkj = Measure of violation of constraint k, Vk = A reference value that is assigned a value greater than the expected maximum value of vkj and nk = Exponential penalty weight factor for constraint k

8.9.3 Objective Function

Objective function is the phenotypic representation of the fitness function. For the present case, objective function is

$$O(x) = Cc * Vc + Cs * Ws \quad \text{... (8.44)}$$

where Cc = Cost of concrete per M^3, Vc = Volume of concrete in M^3, Cs = Cost of steel per Kg and Ws = Weight of steel in Kg.

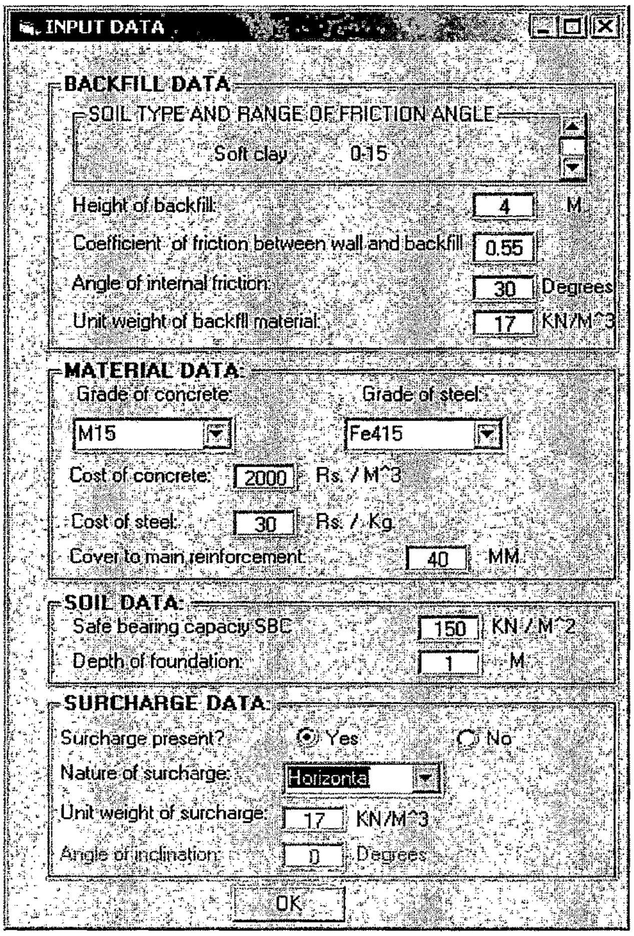
The fitness function is formulated from the objective function as follows:

$$F(x) = 1 / (O(x) \times P_k) \qquad \ldots (8.45)$$

where P_k – total penalty applied to the solution as shown above.

8.9.4 Processors Developed

- **Preprocessor:** The preprocessor is aimed to provide a user-friendly input by the use of forms, menus, option buttons, dialogue boxes etc. The form for input of data is as shown in Fig. 8.14 whereas Fig. 8.15 shows the form for entering the permissible factors of safety and spacing details. The form shown in Fig. 8.16 is developed to supply genetic parameters, penalty factors and geometric constraints.



INPUT DATA

BACKFILL DATA:

SOIL TYPE AND RANGE OF FRICTION ANGLE

Soft clay 0-15

Height of backfill: 4 M

Coefficient of friction between wall and backfill 0.55

Angle of internal friction: 30 Degrees

Unit weight of backfll material: 17 KN/M^3

MATERIAL DATA:

Grade of concrete: M15

Grade of steel: Fe415

Cost of concrete: 2000 Rs. / M^3

Cost of steel: 30 Rs. / Kg

Cover to main reinforcement: 40 MM

SOIL DATA:

Safe bearing capaciy SBC 150 KN / M^2

Depth of foundation: 1 M

SURCHARGE DATA:

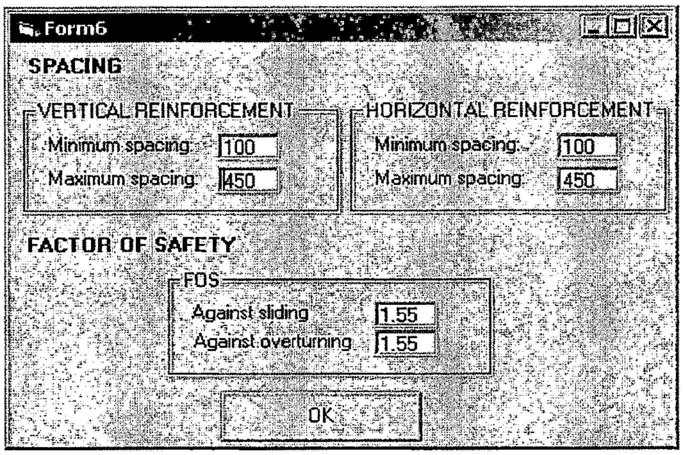
Surcharge present? ● Yes ○ No

Nature of surcharge: Horizonta

Unit weight of surcharge: 17 KN/M^3

Angle of inclination: 0 Degrees

Fig. 8.14 Input Data Form for Retaining Wall



SPACING

VERTICAL REINFORCEMENT

Minimum spacing: 100

Maximum spacing: 450

HORIZONTAL REINFORCEMENT

Minimum spacing: 100

Maximum spacing: 450

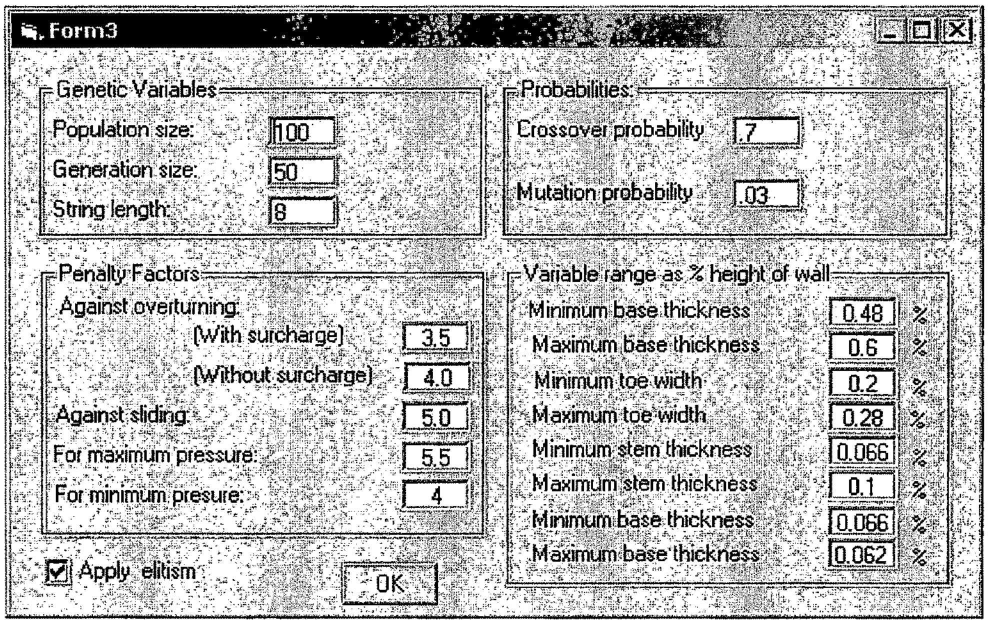
FACTOR OF SAFETY

FOS

Against sliding 1.55

Against overturning 1.55

Fig. 8.15 Form for Input of Factors of Safety and Permissible Spacing



Genetic Variables

Population size: 100

Generation size: 50

String length: 8

Probabilities:

Crossover probability .7

Mutation probability .03

Penalty Factors

Against overturning:

(With surcharge) 3.5

(Without surcharge) 4.0

Against sliding: 5.0

For maximum pressure: 5.5

For minimum presure: 4

☑ Apply elitism

Variable range as % height of wall

Minimum base thickness 0.48 %

Maximum base thickness 0.6 %

Minimum toe width 0.2 %

Maximum toe width 0.28 %

Minimum stem thickness 0.066 %

Maximum stem thickness 0.1 %

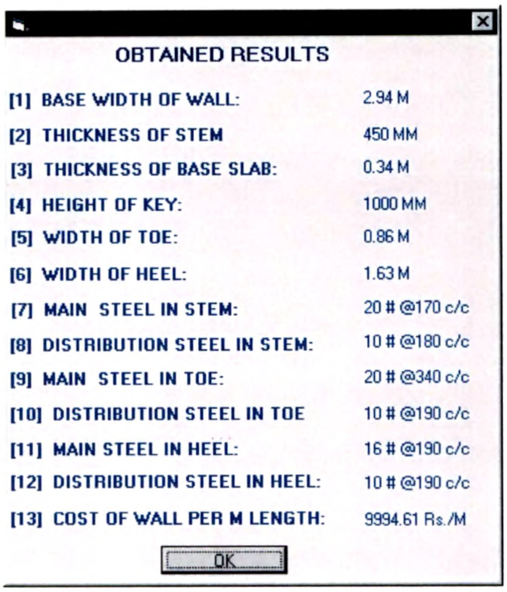
Minimum base thickness 0.066 %

Maximum base thickness 0.062 %

Fig. 8.16 Input of Genetic Parameters

❖ **Mainprocessor:** In the main processor various subroutines are developed. The subroutines developed for GA are the same. A separate subroutine developed for retaining wall is ***DESIGN*** subroutine which carries out the design calculations constraint evaluation and objective function evaluation which is then used to find the fitness of the solution. Elitism is used to improve the efficiency of the algorithm. In elitism, the best individual in every generation is retained until a further improvement is obtained. Thus, the intermediate results need not be stored as the last solution is programmed to be the best-so-far solution.

❖ **Postprocessor:** To have the output in graphical form a post-processor is developed. The post-processor gives the obtained results as shown in Fig. 8.17 and the reinforcement detailing of the wall section as shown in Fig. 8.18.



OBTAINED RESULTS

[1] BASE WIDTH OF WALL:	2.94 M
[2] THICKNESS OF STEM	450 MM
[3] THICKNESS OF BASE SLAB:	0.34 M
[4] HEIGHT OF KEY:	1000 MM
[5] WIDTH OF TOE:	0.86 M
[6] WIDTH OF HEEL:	1.63 M
[7] MAIN STEEL IN STEM:	20 # @170 c/c
[8] DISTRIBUTION STEEL IN STEM:	10 # @180 c/c
[9] MAIN STEEL IN TOE:	20 # @340 c/c
[10] DISTRIBUTION STEEL IN TOE	10 # @190 c/c
[11] MAIN STEEL IN HEEL:	16 # @190 c/c
[12] DISTRIBUTION STEEL IN HEEL:	10 # @190 c/c
[13] COST OF WALL PER M LENGTH:	9994.61 Rs./M

Fig. 8.17 Form Showing Results for Retaining Wall

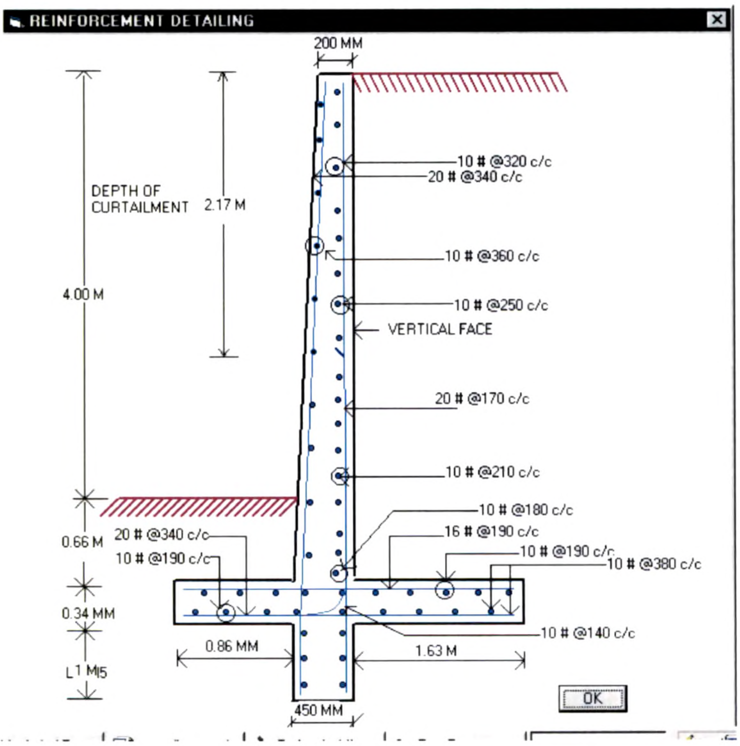
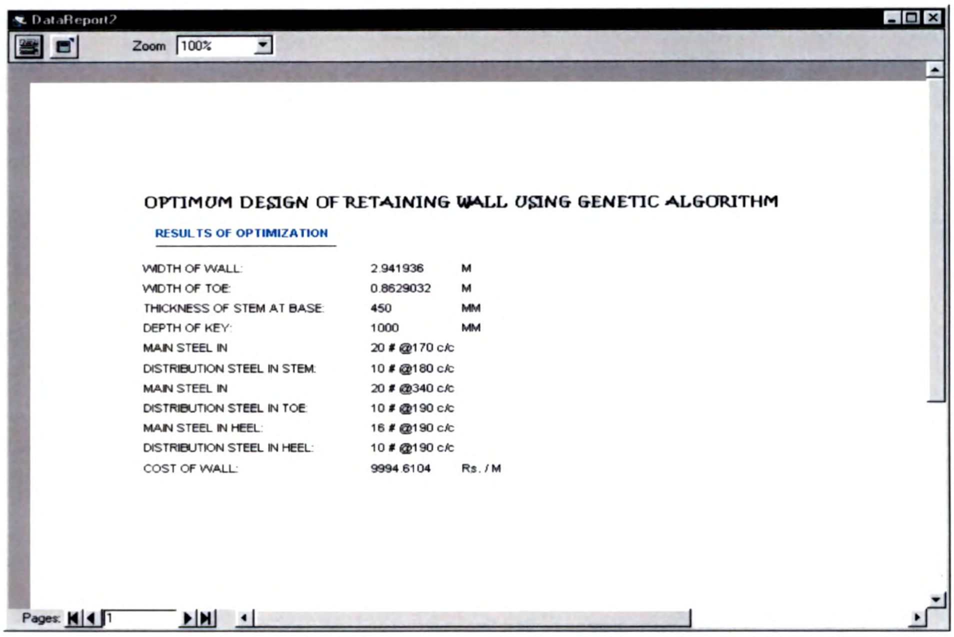


Fig. 8.18 Form Showing Reinforcement Detailing

A report is generated giving details of the input parameters, the genetic parameters and finally the results obtained as shown in Fig. 8.19.



OPTIMUM DESIGN OF RETAINING WALL USING GENETIC ALGORITHM

RESULTS OF OPTIMIZATION

WIDTH OF WALL:	2.941936	M
WIDTH OF TOE:	0.8629032	M
THICKNESS OF STEM AT BASE:	450	MM
DEPTH OF KEY:	1000	MM
MAIN STEEL IN	20 # @170 c/c	
DISTRIBUTION STEEL IN STEM:	10 # @180 c/c	
MAIN STEEL IN	20 # @340 c/c	
DISTRIBUTION STEEL IN TOE:	10 # @190 c/c	
MAIN STEEL IN HEEL:	16 # @190 c/c	
DISTRIBUTION STEEL IN HEEL:	10 # @190 c/c	
COST OF WALL:	9994.6104	Rs. / M

Fig. 8.19 Report Generated by the Software

The software also plots the graphs for Cost Vs. Generations and Fitness Vs Generations as shown in Fig. 8.20

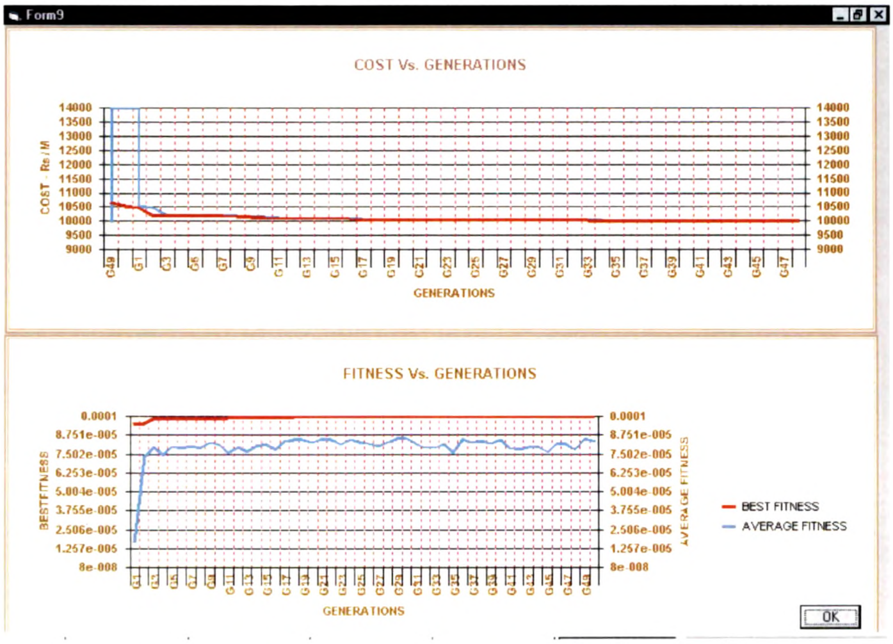


Fig. 8.20 Graph of Cost Vs Generations and Fitness Vs Generations

8.9.5 Example of Cantilever Retaining Wall

A retaining wall to retain the earth 4 m high is to be designed. The top surface is horizontal behind the wall. The soil behind the wall is a well-drained medium dense sand with unit weight = 17 kN / M^3 and angle of internal friction $\Phi = 30°$ and subjected to a surcharge load of 17 kN / M^3. The material under the wall base is same as above with a safe bearing capacity of 150 kN / M^3. The coefficient of friction between base and soil is 0.55. Materials used are M15 grade concrete and Fe415 steel.

All three genetic operators i.e. selection, crossover and mutation are applied to the design problem. Here, it is important to determine the appropriate values of crossover and mutation probabilities. The other genetic parameters to be determined at this stage are the generation numbers, the population size and the string length. The obtained results are compared with those obtained by conventional design method as shown in Table 8.4.

Table 8.4 Comparison of Results for Retaining Wall

Parameter	**Reference [85]**	**Obtained results**
Footing width	3.2 M	2.94 M
Stem thickness at top of footing	400 MM	450 MM
Toe width	1 M	0.86 M
Footing thickness	450 MM	340 MM
Cost per m length	Rs. 10800 /-	Rs. 9995 /-

8.10 OPTIMUM DESIGN OF SILOS

Materials such as coal, cement, broken stone, gravel, clinker, wheat and barley may be stored in bulk in a number of ways. However, where fully automatic feed and extraction are desired, these materials are stored in tall containers provided with outlet at the bottom. The containers are elevated and generally provided with sloping bottoms. Elevated hopper bottoms must be designed to provide an angle that ensures free flow of the material and this depends on the angle of friction between the material and bottom surface. A module is developed in the software to optimize the height to diameter ratio and opening of the silo based on GA taking into account the IS code recommendations. Provision is made in the software to design the silo with large variety of storing materials like coal, cement, food grains, ash, ore etc.

8.10.1 Design Variables

The different parts of a silo are: The vertical sidewall, Hopper bottom, Edge beams and Columns. Height of the silo, its diameter and opening of the hopper bottom are considered as the design variables. Certain other properties like the density of the filling material, angle of repose of the fill material, coefficient of friction between wall and material, grade of concrete, grade of steel and weight of material for which the silo is to be designed must be given as input data.

8.10.2 Constraints

For the design of silo, the following constraints as per IS: 4995 [86] are considered:

1. As there is no general agreement on the correct evaluation of the lateral pressure IS: 4995 [86] recommends the use of Janssen's theory.
2. Silo walls are usually thin and to ensure resistance to buckling the compressive stress on the net concrete section deducting all openings should not exceed 0.25fck.
3. The thickness should not be less than 100 MM.
4. Minimum circumferential reinforcement = 0.25 % and minimum vertical reinforcement = 0.12 %.
5. Minimum cover = 30 MM.
6. Maximum spacing for reinforcement = 300 MM and minimum spacing for reinforcement = 100 MM.
7. Angle made by the hopper bottom with the horizontal should be 15° more than the angle of repose of the fill material.

Some of the constraints are handled in the design module while some are handled by specifying proper range for design values to satisfy the constraints. Only for the following two constraints penalty method is employed. A penalty is also applied if the tensile stress in concrete exceeds the permissible value.

- **Height of the silo**

  Height of the silo is not allowed to exceed 30 M

  Constraint : $H \leq 30\,\mathrm{m}$ ... (8.46)

  Penalty : $P_{height} = 1 + (H - MaxH) / MaxH$ ... (8.47)

- **Tensile stress in the concrete**

  Constraint : $Ftc \leq sct$ ... (8.48)

  Penalty : $P_{tensile} = 1 + (Abs(Ftc)\text{-}sct) / sct$ ... (8.49) where H – Actual height of the silo, MaxH – Maximum permissible height of the silo, Ftc – Actual tensile stress in concrete and Sct – Permissible tensile stress in concrete.

8.10.3 Objective Function

For the present case, objective function is

$$O(x) = Cc * Vc + Cs * Ws + C_f * A_f \qquad \dots (8.50)$$

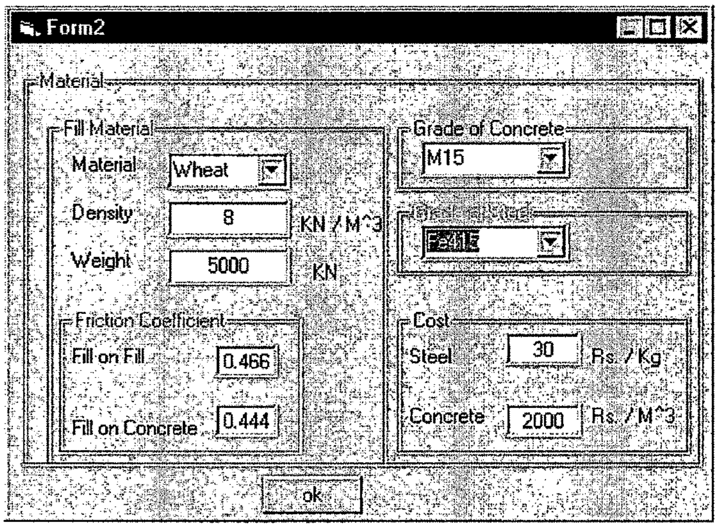
where Cc - Cost of concrete per M^3, Vc - Volume of concrete, Cs - Cost of steel per Kg., Ws -Weight of steel, C_f - Cost of form work per m^2 and A_f - Area of form work.

Fitness function is then calculated as

$$F(x) = 1 / O(x) \times P_{height} \times P_{tensile} \qquad \dots (8.51)$$

8.10.4 Processors Developed

- **The preprocessor:** The preprocessor is aimed at supplying the basic input data through graphical user interface in the form of menus and forms. The various forms developed under preprocessor are displayed in the following screenshots. Fig. 8.21 shows the form through which input data is supplied.



Form2

Material

Fill Material

Material: Wheat

Density: 8 KN / M^3

Weight: 5000 KN

Friction Coefficient

Fill on Fill: 0.466

Fill on Concrete: 0.444

Grade of Concrete: M15

Grade of Steel: Fe415

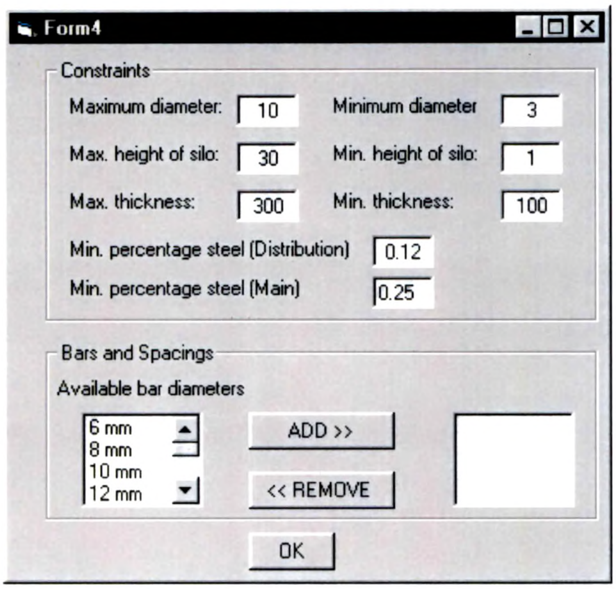
Cost

Steel: 30 Rs. / Kg

Concrete: 2000 Rs. / M^3

Fig. 8.21 Form for Supplying Input Data for Silo

The designed silo must also adhere to certain geometric constraints. The constraints for maximum and minimum height, diameter and thickness of the silo and minimum reinforcement percentages are specified as shown in Fig. 8.22. Input must also be given for the genetic parameters. It is difficult to decide some parameters at the first go. Hence, the parametric study was carried out to decide the probability of cross over which is depicted in Fig. 8.23. From the plot (Fig. 8.23) it can be verified that a value of 0.6 is suitable for the present problem. Thus, genetic parameters can be supplied as shown in Fig. 8.24.



Constraints

Maximum diameter: 10 Minimum diameter 3

Max. height of silo: 30 Min. height of silo: 1

Max. thickness: 300 Min. thickness: 100

Min. percentage steel (Distribution) 0.12

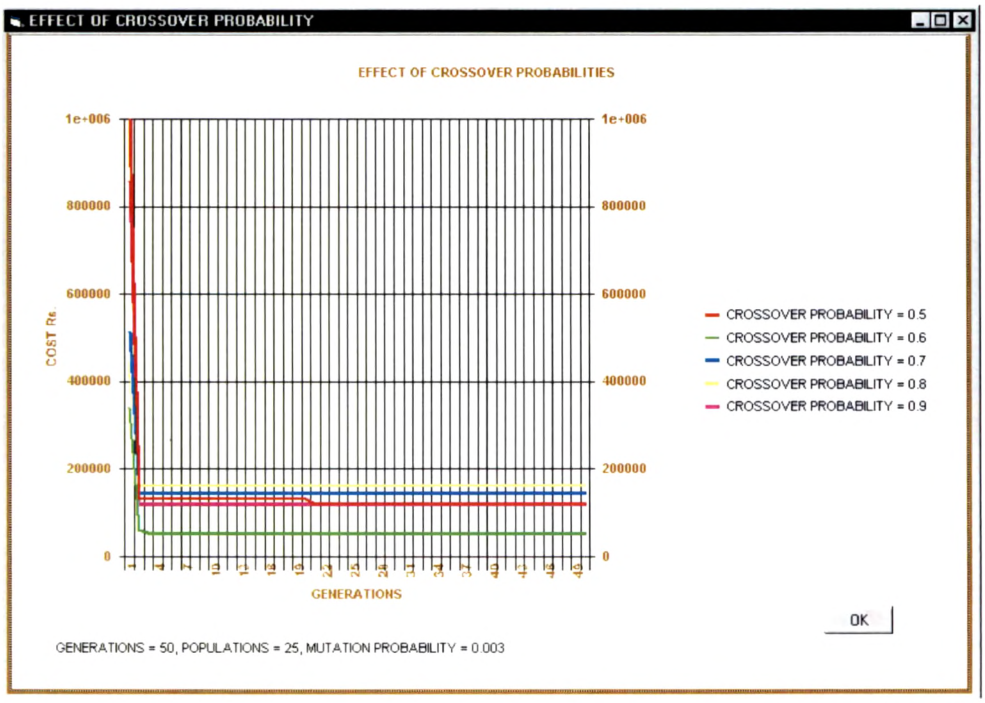
Min. percentage steel (Main) 0.25

Bars and Spacings

Available bar diameters

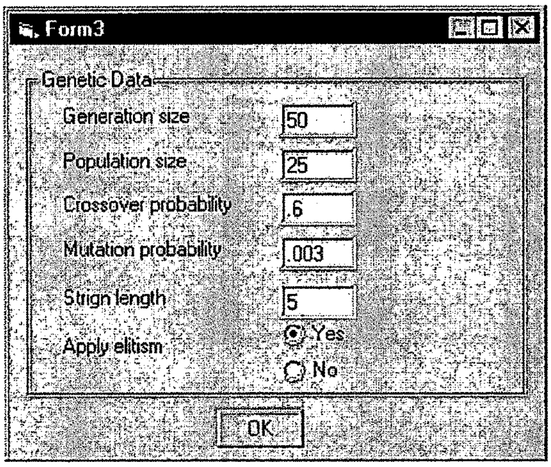
6 mm
8 mm
10 mm
12 mm

Fig. 8.22 Form for Defining Constraints for Silo



GENERATIONS = 50, POPULATIONS = 25, MUTATION PROBABILITY = 0.003

Fig. 8.23 Form Showing Effect of Crossover Probability



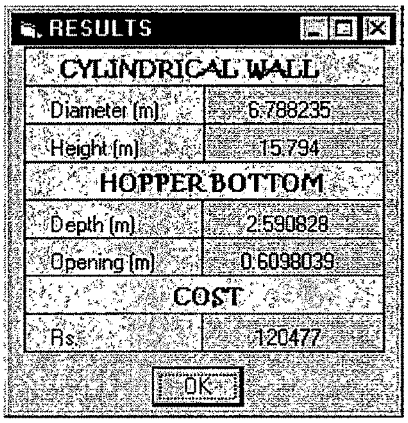
Genetic Data

Generation size	50
Population size	25
Crossover probability	.6
Mutation probability	.003
Strign length	5
Apply elitism	● Yes ○ No

Fig. 8.24 Input of Genetic Parameters

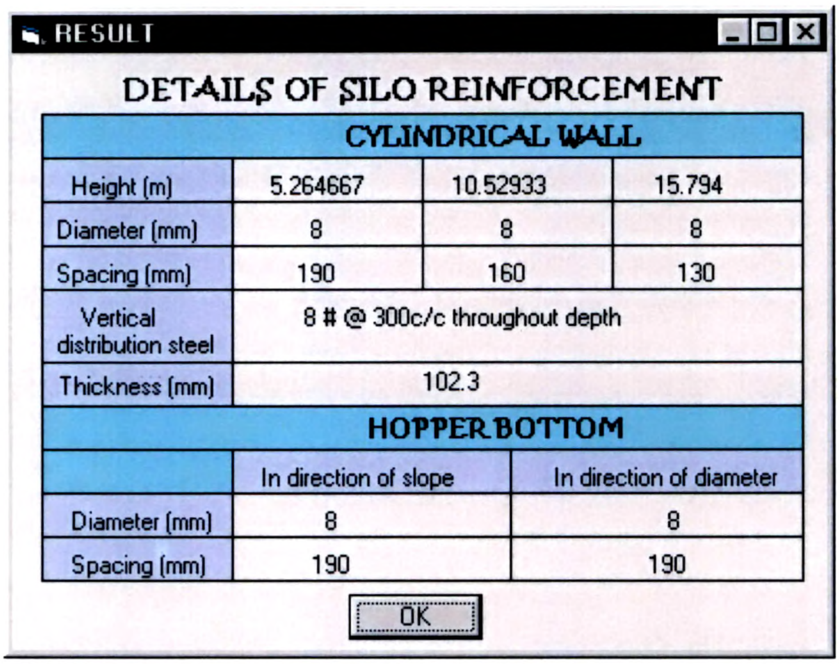
Clicking on the run menu of the main form runs the algorithm for the problem and displays the output. The design of silo and evaluation of constraints and objective function is carried out in the main processor in ***DESIGN*** subroutine.

- **The postprocessor:** The postprocessor gives geometric as well as reinforcement details as shown in Figures 8.25- 8.27. Provision is made to display the output in the form of graphs, drawings, charts etc. This has been accomplished by plotting the graph of Fitness Vs. Generations and Cost Vs Generations as shown in Fig. 8.28. Finally, to get an idea of the algorithm progress, a series of graphs of different generations are plotted which show the variation of one of the variables. Figure 8.29 indicates the gradual convergence that occurs with respect to one of the variables i.e. diameter of the silo.



CYLINDRICAL WALL	
Diameter (m)	6.788235
Height (m)	15.794
HOPPER BOTTOM	
Depth (m)	2.590828
Opening (m)	0.6098039
COST	
Rs.	120477

Fig. 8.25 Form Showing Obtained Results for Silo

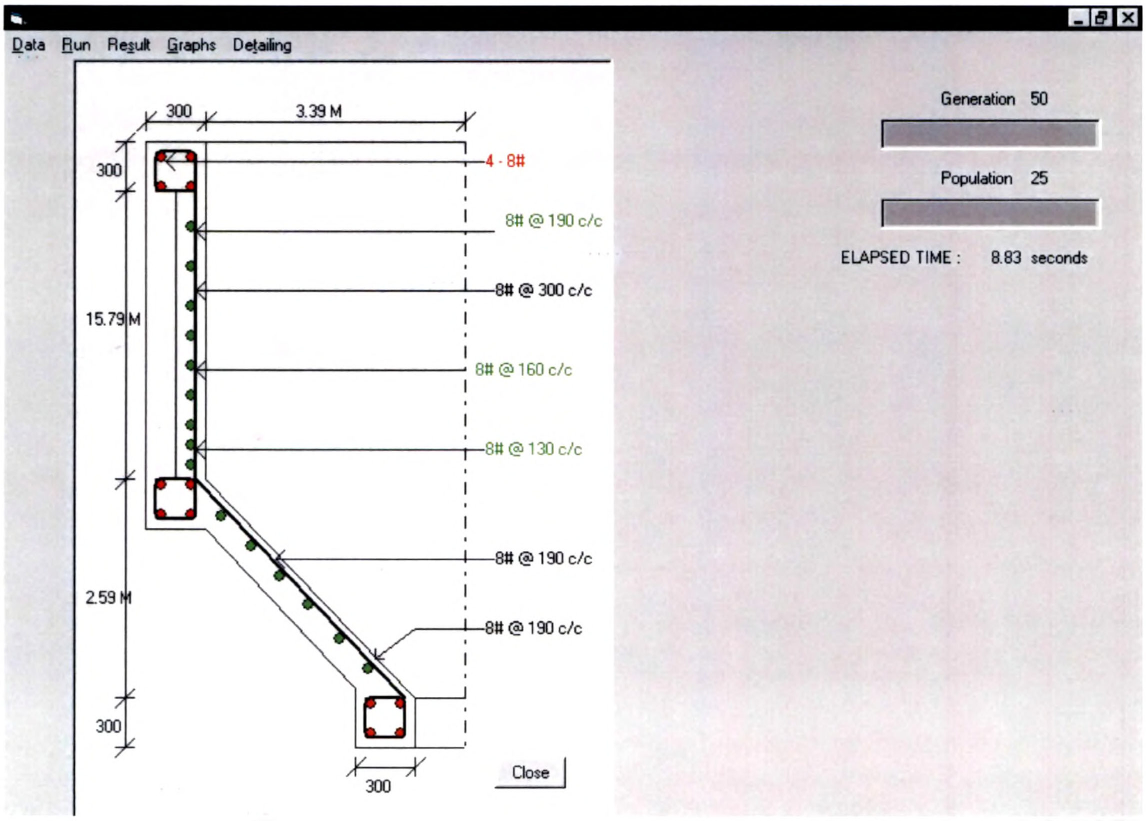


RESULT

DETAILS OF SILO REINFORCEMENT

CYLINDRICAL WALL			
Height (m)	5.264667	10.52933	15.794
Diameter (mm)	8	8	8
Spacing (mm)	190	160	130
Vertical distribution steel	8 # @ 300c/c throughout depth		
Thickness (mm)	102.3		
HOPPER BOTTOM			
	In direction of slope	In direction of diameter	
Diameter (mm)	8	8	
Spacing (mm)	190	190	

Fig. 8.26 Form Giving Details of Reinforcement for Silo



Generation 50

Population 25

ELAPSED TIME : 8.83 seconds

Fig. 8.27 Form Showing Detailing of the Silo Section

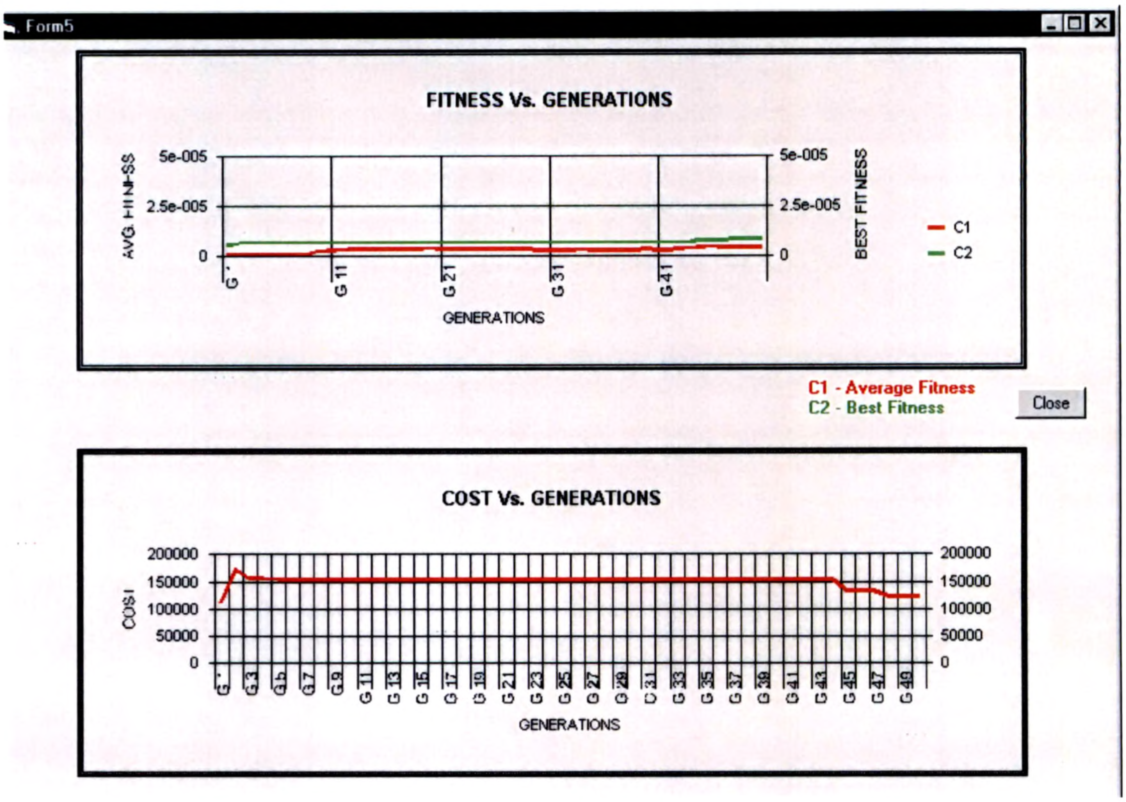


Fig. 8.28 Plot of Fitness vs. Generations and Cost vs. Generations

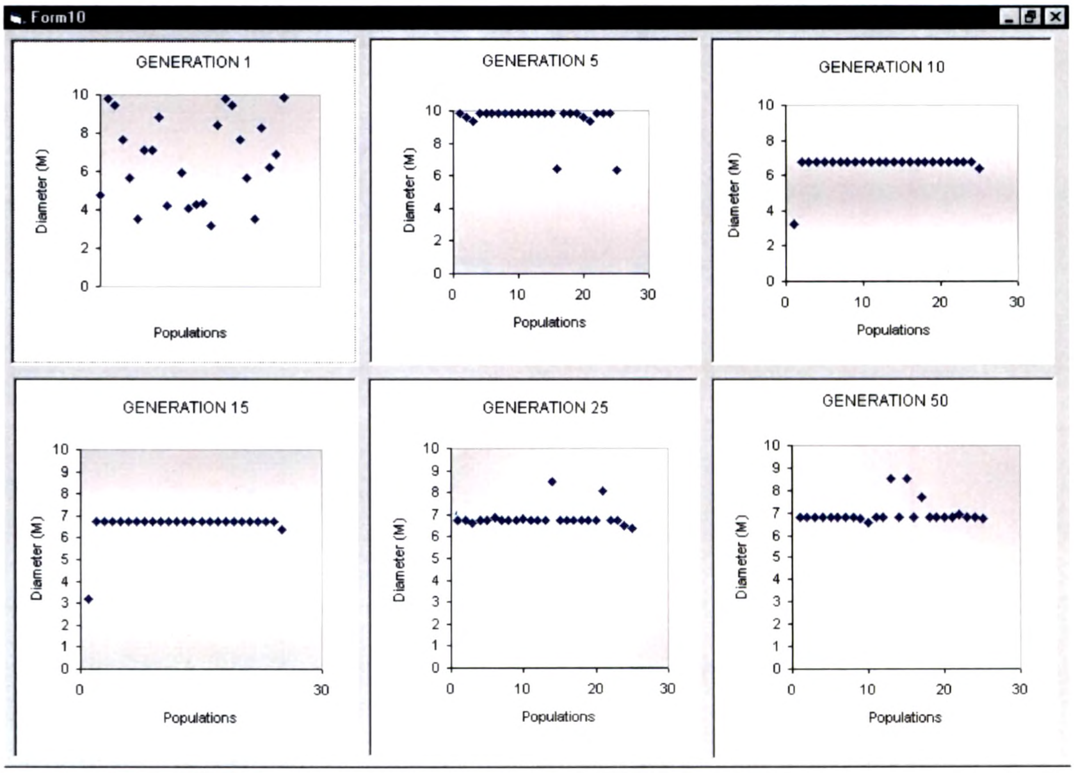


Fig. 8.29 Form Showing Variation of Diameter of the Silo over Generations

8.10.5 Example of Design of Silo

A silo is to be designed with minimum storing capacity of 625 M^3 with the coefficient of friction between wall and material as 0.444 and the ratio of horizontal to vertical pressure intensity as 0.4. Angle of repose of the material is 25° and material used is M15 grade concrete and Fe415 steel. The material stored is wheat with a density of 8 kN / M^3.

A comparison of results (Table 8.5) obtained based on GA with that based on conventional method of design indicates a percentage saving of about 27 %.

Table 8.5 Comparison of Results

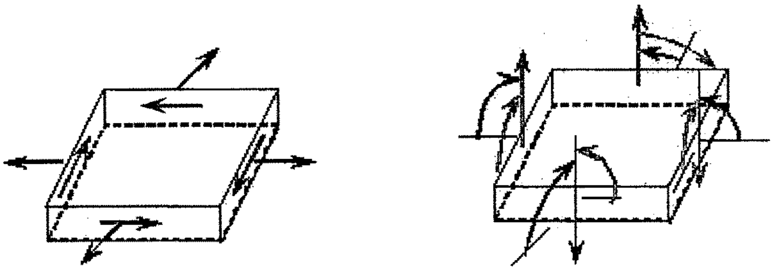
PARAMETER	REFERENCE [86]	GA RESULTS
Diameter of silo	6 M	6.78 M
Depth of cylindrical portion	20 M	15.79 M
Thickness of side wall	175 MM	102.3 MM
Depth of hopper bottom	2.5 M	2.59 M
Diameter of opening	1 M	0.61 M
Cost of silo Rs.	165151 /-	120477 /-

8.11 OPTIMUM DESIGN OF FOLDED PLATES

A folded plate is composed of interconnected plates and is considered to span as a beam between diaphragm supports. The longitudinal stresses in a folded plate or cylindrical shell can be obtained by using simple beam formulae, e.g., $\sigma_x = (M.z / I)$, where M is the bending moment at any section x, and I is the moment of inertia of the cross section about the neutral axis.

Figure 8.30 shows an element of a folded plate. The element is subjected to a general system of external loading, defined by the three components X, Y and Z corresponding to x, y and z directions. The internal actions developed to resist the external loading can be divided into two actions: namely slab action [Fig. 8.30(b)] and plate action [Fig .8.30(a)].

For small displacements, the slab action supports the Z-component of the external load while the plate action takes care of the in-plane load components X and Y of the external load. The slab action deals with the bending of the individual plates out of their plane, that is the plate acts as a two-way slab continuous over junctions.



(a) Plate Action **(b) Slab Action**

Fig. 8.30 Elements of Folded Plates under Different Actions

The plate action involves primarily the displacements and stresses in the plane of the plate and is identical to a deep beam action. It should be noted that the slab and plate actions are independent but interact at the folds or junctions of the plates.

In general, the existing methods of analysis of folded plate structures may be broadly classified under the following three categories: Elasticity method, Ordinary folded plate theory and Beam method.

For the present work, beam method is adopted as the method of analysis because of the following advantages:

- It is simple and gives the designer a physical insight in the behaviour of the shell.
- It can be applied with equal ease to v-shaped, trough-shaped as well as shells of noncircular cross sections.

Using the beam theory, the analysis steps are as follows:

1. Calculate the load per unit length, along the span, for the given distributed loading.
2. Find the neutral axis location and the moment of inertia, I, about the Neutral axis.
3. Find the variation over depth of the maximum longitudinal stresses, σ_x, at x = L / 2.
4. Find the moment of area, Q about the neutral axis and solve for transverse shear $N_{x\Phi}$. values at the support x = 0.
5. Find the specific shear q, at various points along the cross section.
6. Under the loads, p and q, analyze the arch by the plane frame analysis method, in order to obtain M_Φ, Q_Φ and N_Φ.

The present section deals with the optimization of a V-shaped folded plate. The details of the same are given below.

8.11.1 Design Variables

The variables considered are as follows:

1. Inclined length of the plate.
2. Angle of inclination of the plate with the horizontal.
3. Thickness of the plate.

Keeping in mind the above considerations for selecting the dimensions of the folded plate, the variables are allowed to vary within the following ranges:

1. Inclined length of the plate: 0.6 to span / 5.
2. Inclination of the plate : 35° to 45°.
3. Thickness of the plate : 80 MM to 110 MM.

Once the above geometric details are decided, the remaining parameters like the number of V-units as well as the structural depth of the V-unit can be determined.

8.11.2 Constraints

1. Minimum thickness provided is 80 MM as per IS: 2210 [88].
2. Maximum angle of inclination provided is 45° to prevent the use of back form.
3. Minimum steel provided, in either direction is 0.3 %.
4. Minimum structural depth is taken as span / 12.

For handling constraints 1 to 3, no extra step is required. However, to satisfy constraint 4 a penalty is applied as follows whenever structural depth is less.

$$P_{sd} = (1 + V_{kjsd} / V_{ksd})^n \qquad \ldots (8.52)$$

where V_{kjsd} is span /12 – structural depth, V_{ksd} is span /12 and n is the penalty factor for structural depth. The value has been selected as 5 for the present case.

8.11.3 Objective Function

Objective function is the governing criterion for fitness function. The fitness being the driving force of the GA, the objective function must be decided carefully. For the present work, the objective function is selected as

$$O(x) = V_c \text{ x } C_c + W_s \text{ x } C_s + A_f \text{ x } C_f \qquad \dots (8.53)$$

Where V_c = Volume of concrete in M^3, C_c = Cost of concrete in Rs. / M, W_s = Weight of steel in Kg, C_s = Cost of steel in Rs. / Kg, A_f = Area of formwork required and C_f = Cost of formwork per M^2.

The fitness function is then calculated as

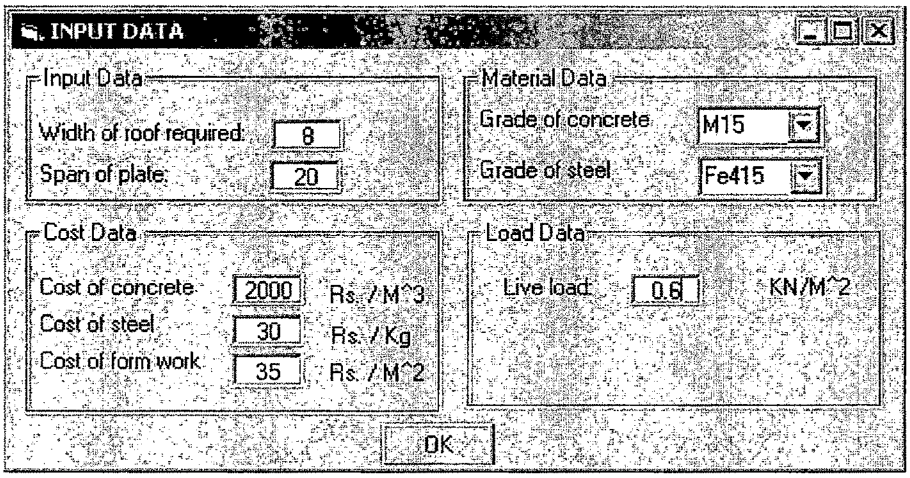
$$F(x) = 1 / (\text{cost} \times P_{sd}) \qquad \dots (8.54)$$

where cost is the cost of the structure for the corresponding solution.

8.11.4 Example of V - Shaped Folded Plate

A V-shaped folded plate problem is considered here for 20 M span. The roof is 8 M wide and subjected to a live load of 0.6 kN / M^2.

The pre-processor developed consists of the form for supply of input data (Fig.8.31) and the form for supply of genetic data like crossover probability, mutation probability, string length, penalty factors etc. (Fig. 8.32).



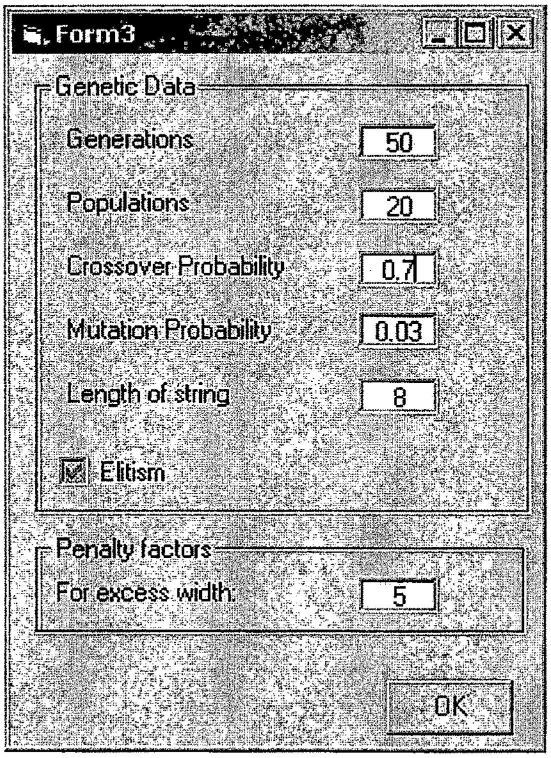
Input Data
Width of roof required: 8
Span of plate: 20

Material Data
Grade of concrete M15
Grade of steel Fe415

Cost Data
Cost of concrete 2000 Rs. / M^3
Cost of steel 30 Rs. / Kg
Cost of form work 35 Rs. / M^2

Load Data
Live load 0.6 KN/M^2

Fig. 8.31 Form for Input of Data for Folded Plate



Genetic Data

Generations 50

Populations 20

Crossover Probability 0.7

Mutation Probability 0.03

Length of string 8

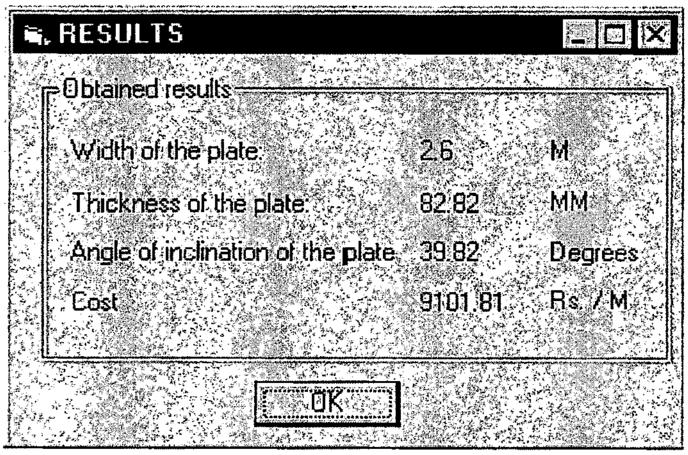
☑ Elitism

Penalty factors

For excess width: 5

Fig. 8.32 Form for Input of Genetic Parameters for Folded Plate

The program is then run, by clicking on the run menu of the main form, which also initiates the progress and the timer features of the software. The obtained results are as seen in Fig. 8.33. The detailing of the section is also carried out by the software as shown in Fig. 8.34.



Obtained results

Width of the plate: 2.6 M

Thickness of the plate: 82.82 MM

Angle of inclination of the plate 39.82 Degrees

Cost 9101.81 Rs. / M.

Fig. 8.33 Form Showing Obtained Results for Folded Plate

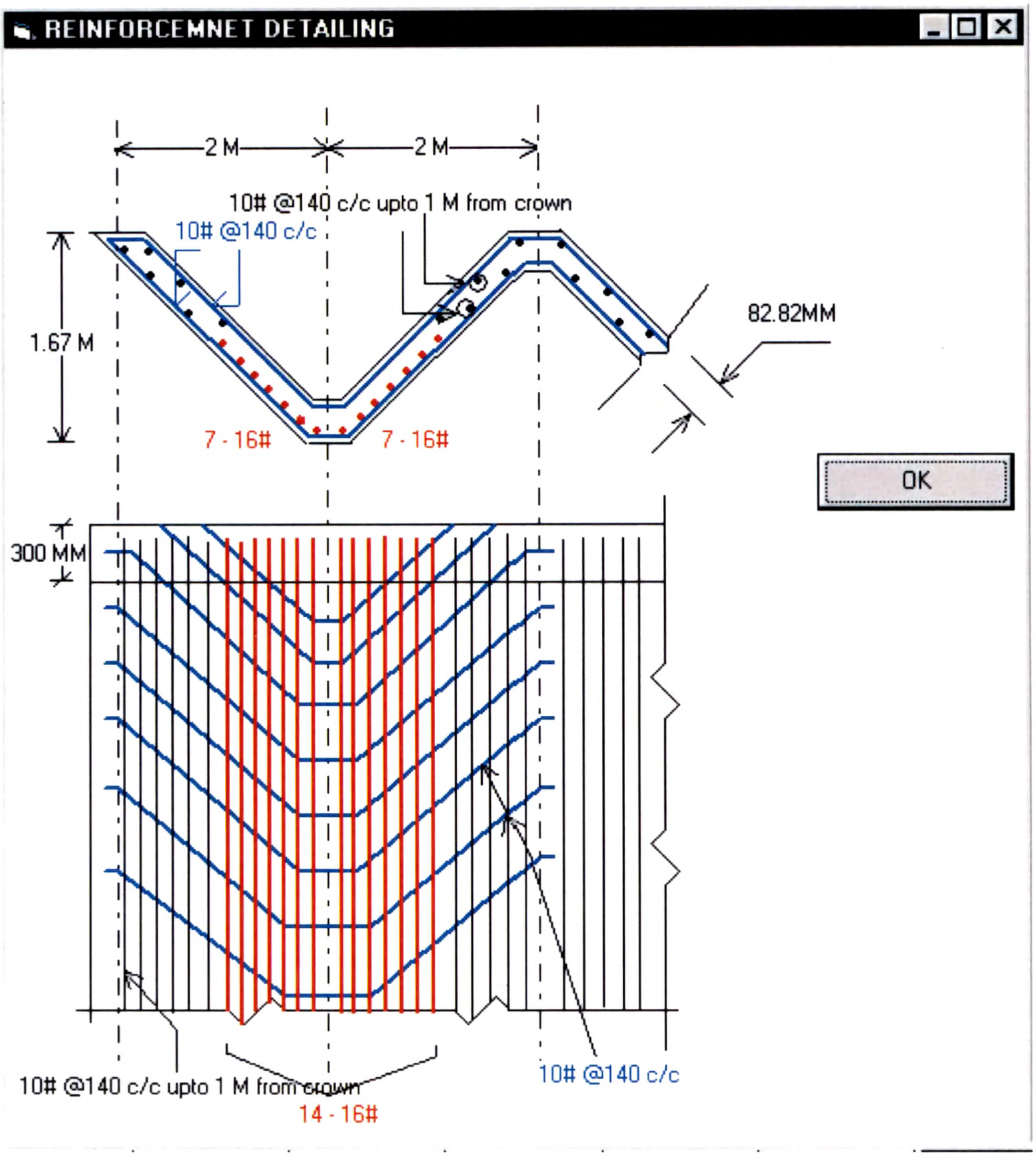


Fig. 8.34 Form Showing Detailing of the folded Plate Section

A graph of Fitness vs. Generations is plotted and is as seen in Fig. 8.35. The variation of transverse moments along the plate is observed as shown in Fig. 8.36. It can be seen from the graph that the average fitness and the maximum fitness improve as the generation progress. However, there is no significant improvement in the minimum fitness. A report generated by the software is as shown in Fig. 8.37, which gives details of the input data, genetic data and the results obtained.

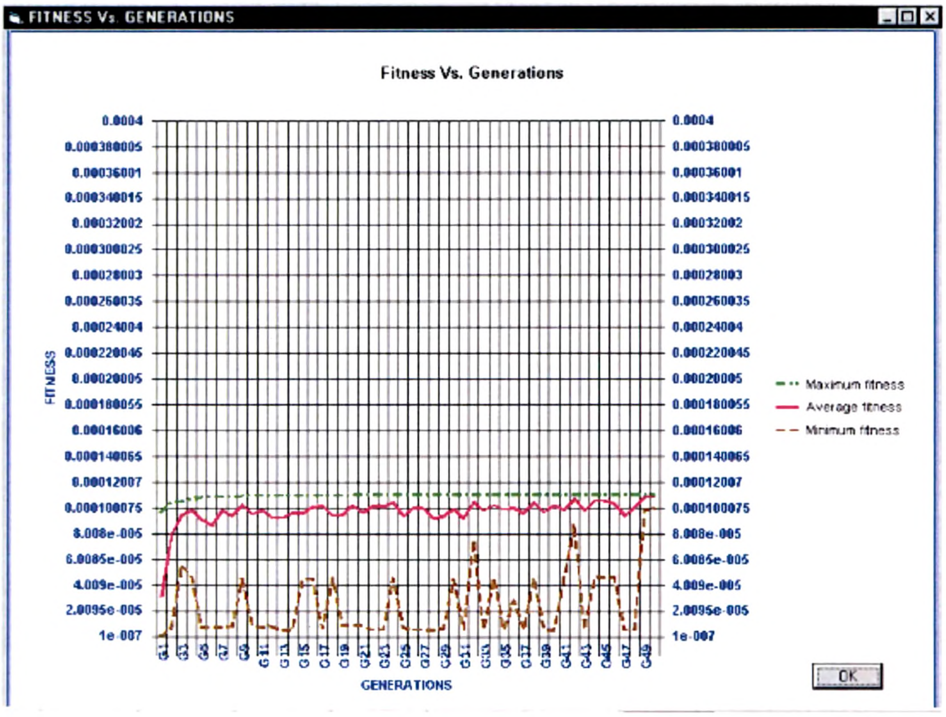


Fig. 8.35 Form showing Plot of Fitness vs. Generations

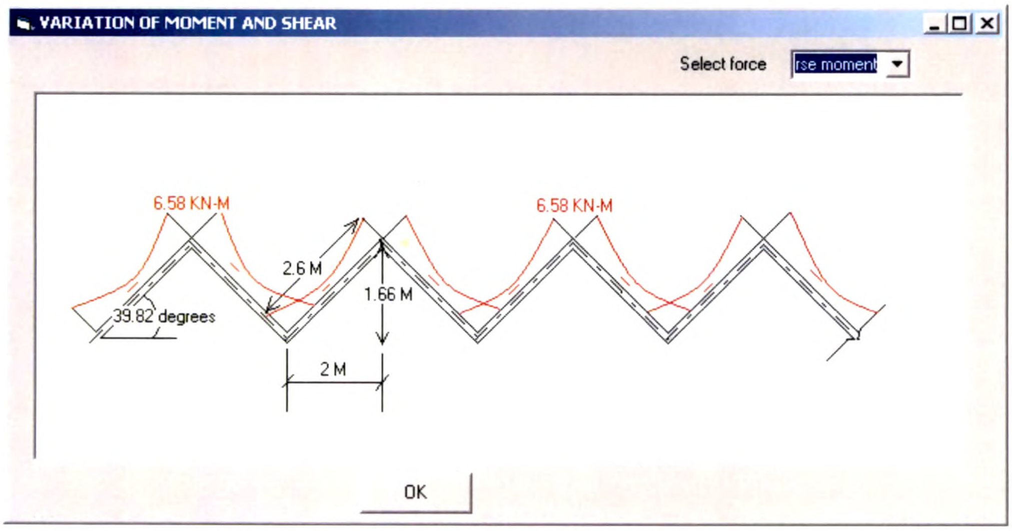
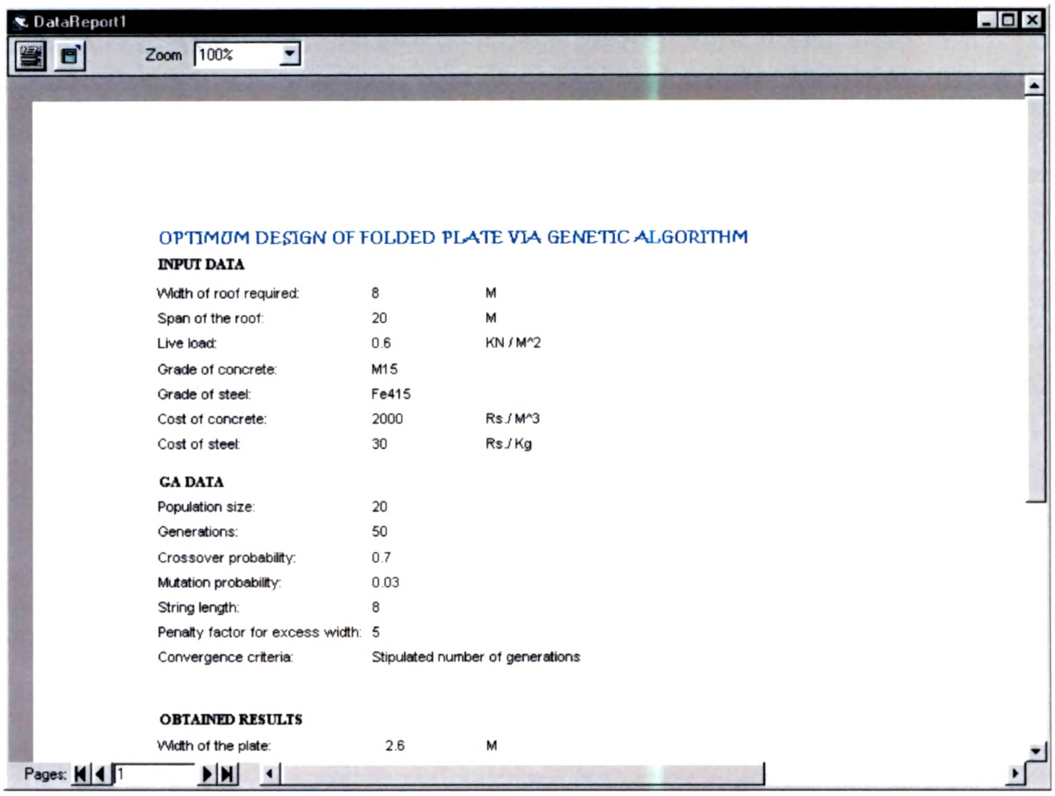


Fig. 8.36 Variation of Transverse Moments Along the Plate

Finally, to get an idea of the algorithm progress a series of graphs showing the status of variables in the corresponding generation are plotted as shown in Fig. 8.38. During the first generation since the values of the variables are a result of randomly generated strings, they are scattered. As the run matures, each variable converges to a particular value indicating the optimum.



OPTIMUM DESIGN OF FOLDED PLATE VIA GENETIC ALGORITHM

INPUT DATA

Width of roof required:	8	M
Span of the roof:	20	M
Live load:	0.6	KN / M^2
Grade of concrete:	M15	
Grade of steel:	Fe415	
Cost of concrete:	2000	Rs./ M^3
Cost of steel:	30	Rs./ Kg

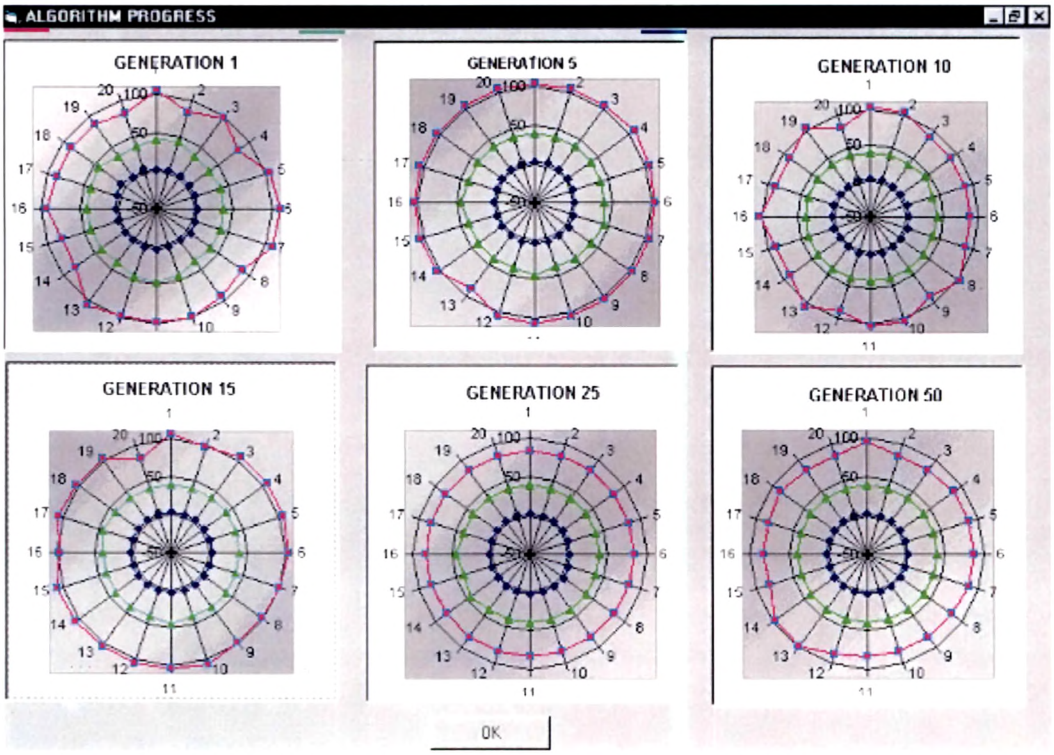
GA DATA

Population size:	20
Generations:	50
Crossover probability:	0.7
Mutation probability:	0.03
String length:	8
Penalty factor for excess width:	5
Convergence criteria:	Stipulated number of generations

OBTAINED RESULTS

Width of the plate:	2.6	M

Fig. 8.37 Report Generated



- Thickness - Inclination - Inclined length of plate

Fig. 8.38 Form Showing Algorithm Progress

8.11.5 Comparison of results

The obtained results are compared with those obtained by conventional design method.

PARAMETERS	REFERENCE [87]	GA RESULTS
Inclined length of plate	2.82 M	2.6 M
Structural depth	2 M	1.66 M
Angle of inclination	45°	39.82°
Thickness of plate	100 MM	82.82 MM
Cost / M length	9876 /-	9102 /-

8.12 Cost Optimization of Machine Foundation

The optimum design of RCC structures subjected to static loading is extended in the present section to design of RCC structures subjected to dynamic loading, by considering a case of optimization of block type machine foundation for reciprocating type machines. Selection is one of the operators used in genetic evolution based optimization. In this work different selection schemes have been tried and compared. The operator elitism is also tested in the present work. In addition to constant rate mutation operator, the variable rate mutation operator is also tried and seemed to make no significant improvement in the search process in the presented work. During the search process the constraint violation is penalized by suitable penalty function. The result obtained by the software for block type machine foundation is presented and compared with available literature.

8.12.1 Design Variables

As the cost of the foundation depends entirely on volume of concrete and this finally is affected by the dimensions of concrete block the length, width and depth of the foundation block are considered as design variables. The solution string thus consist of three substring of predefined length representing three design variables.

8.12.2 Objective Function

The optimum design of the block type machine foundation using GA is formulated as:

Minimize, $O = V_c C_c + W_s C_s + A_f C_f$,

subject to, $A_z \leq A_{per}$ and $A_{top} \leq A_{per}$ ... (8.55) where O is an objective function (i.e. cost of the foundation), V_c is volume of concrete in m^3, C_c is cost of concrete per m^3, W_s is weight of reinforcing steel in kg, C_s is cost of steel per kg, A_f is area of form work for foundation, C_f is cost of form work per m^2, A_z is vertical amplitude of machine foundation system in m, A_{per} is permissible amplitude in m and A_{top} is total amplitude at the top of foundation in m.

8.12.3 Constraints

❖ **Amplitude Constraints**

$A_z \leq A_{per}$ and $A_{top} \leq A_{per}$ ... (8.56)

If these constraints are violated penalty factors PF1 and PF2 are calculated as,

$$\text{PF1} = \begin{cases} 1 & \text{if, } \frac{|A_z|}{A_{per}} \leq 1 \\ k1\frac{|A_z|}{A_{per}} & \text{if, } 1 < \frac{|A_z|}{A_{per}} \leq c_1 \\ k2\frac{|A_z|}{A_{per}} & \text{if, } \frac{|A_z|}{A_{per}} > c_1 \end{cases} \quad \text{and,} \quad \text{PF2} = \begin{cases} 1 & \text{if, } \frac{|A_{top}|}{A_{per}} \leq 1 \\ k1\frac{|A_{top}|}{A_{per}} & \text{if, } \frac{|A_{top}|}{A_{per}} > 1 \end{cases}$$

... (8.57)

where k1 and k2 are penalty rates; and c_l is limiting percentage of minor constraint violations. PF1 is calculated as per linear triple segment penalty function. The main advantage of this function is that solutions that slightly violate the constraints may be penalized at a lower rate than solutions with severe violations. PF1 takes value 1 when the solution does not violate the constraint. PF2 is calculated as per double segment penalty function shown above.

❖ **Frequency Constraints**

$1.4\omega_m < \omega_{nz}, \omega_{n1}, \omega_{n2} < 0.5\omega_m$ [89] (constraint preventing resonance) ... (8.58)

where ω_m is operating frequency of machine, ω_{nz} is vertical natural frequency of machine foundation system and ω_{n1} and ω_{n2} are coupled natural frequencies of the system ($\omega_{n1} > \omega$-$_{n2}$).

If these constraints are violated penalty factors PF3 and PF4 are calculated as,

$$\mathrm{PF3} = \begin{cases} 1 & \text{if, } 1.4 < \dfrac{|\omega_{nz}|}{\omega_m} < 0.5 \\ \mathrm{k1}\dfrac{|\omega_{nz}|}{\omega_m} & \text{if, } 1.4 \geq \dfrac{|\omega_{nz}|}{\omega_m} \geq 0.5 \end{cases} \quad \text{and, } \mathrm{PF4} = \begin{cases} 1 & \text{if, } 1.4 < \dfrac{|\omega_{nl}|}{\omega_m} < 0.5 \\ \mathrm{k1}\dfrac{|\omega_{nl}|}{\omega_m} & \text{if, } 1.4 \geq \dfrac{|\omega_{nl}|}{\omega_m} \geq 0.5 \end{cases} \qquad \text{... (8.59)}$$

When the solution violates the constraints, penalized objective function (O_p) is calculated as,

$$O_p = O \times PF1 \times PF2 \times PF3 \times PF4 \qquad \text{... (8.60)}$$

The fitness (F) of a solution string is calculated as,

$$F = \frac{C_{max} - O_p}{C_{max}} \qquad \text{... (8.61)}$$

where C_{max} is the maximum possible cost of the foundation.

The effect of penalty factors PF_1, PF_2, PF_3, and PF_4 is to increase the objective function depending on the degree of constraint violation and to decrease the fitness function written above, thus a solution that violates the constraints will have low fitness value and will have less chances to be selected in the next generation.

8.12.4 The Developed Processors

The preprocessor consists of forms for supplying initial input data such as machine data, spring absorber data, data related to dimension variability and genetic data. These forms are shown in Figures 8.39 to 8.41. The main-processor controls the operation such as generation of random solutions, decoding of variables, dynamic analysis of machine foundation system, evaluation of solution string through the calculation of objective function, penalty functions and fitness functions. It also carries out various GA operators such as selection, crossover, mutation and elitism. Postprocessor provides the analysis results for the optimum solution found in each generation, final dimensions of foundation block and reinforcement detail.

8.12.5 Example of Block Foundation

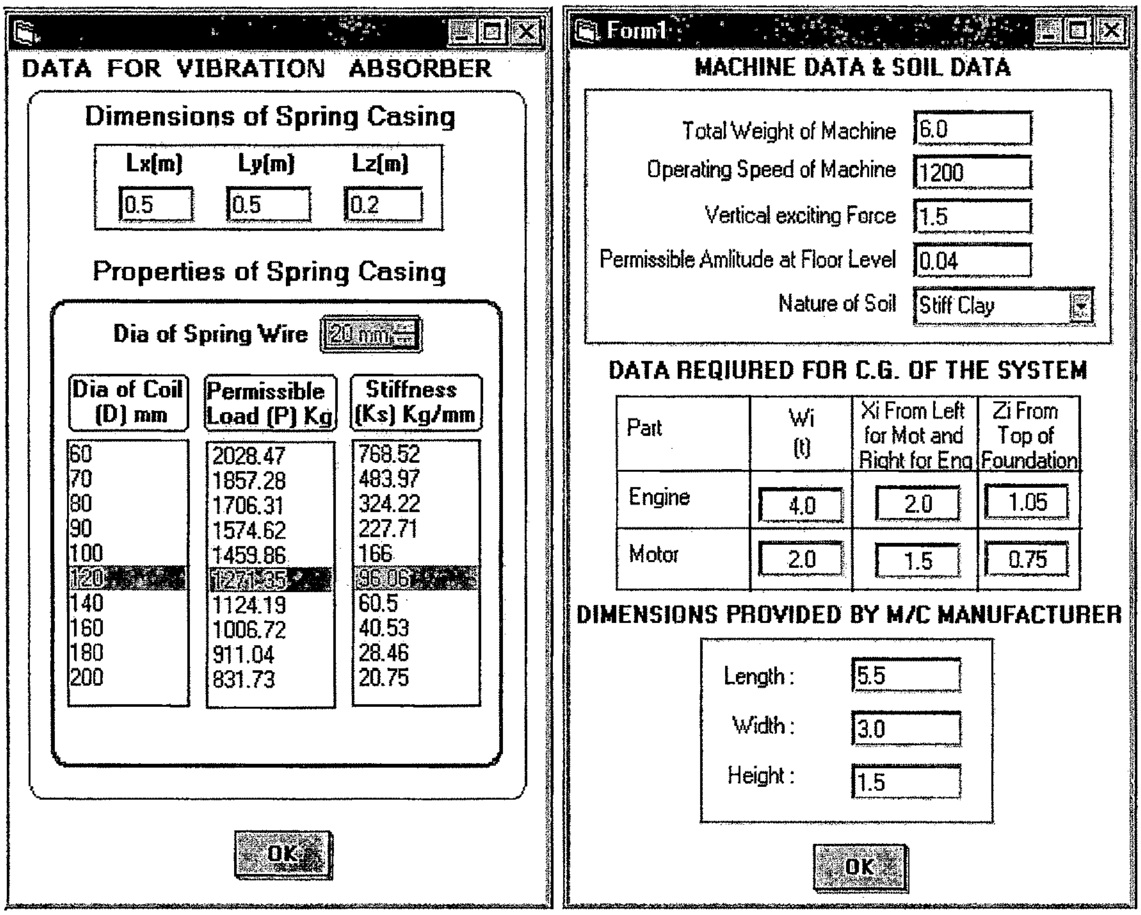
Here, an example of a block foundation for a Diesel Engine is considered and results are compared with the solution available in the literature [89]. The data considered is as follows:

(i)	Total weight of machine	: 6 t
(ii)	Operating speed of machine (f_m)	: 120 rpm
(iii)	Vertical exciting force (P_z)	: ± 1.5 t
(iv)	Permissible amplitude at floor level	: 0.04 mm
(v)	Nature of soil	: stiff clay
(vi)	Dimensions of foundation (L X B X D)	: 5.5 X 3.0 X 1.5 m
(vii)	Centre of gravity of machine parts:	

Part	W_i (t)	$\bar{X}_i$ (m)	$\bar{Z}_i$ (m)
Engine	4.0	3.5	2.55
Motor	2.0	1.5	2.25

Figure 8.39 shows the forms developed for data entry related to machine, soil, vibration absorber and dimensions suggested by the machine manufacturer. User is allowed to fix any dimension for operational requirement of machine using form shown in Fig 8.40. Appropriate values of genetic parameters can be supplied using the form depicted in Fig. 8.41. This form also displays the values of GA parameter selected for this problem. To study the effect of different selection schemes and variable rate mutation the graphs showing improvement in the maximum fitness value during the search process are developed and presented in Fig. 8.42. Figures 8.43 and 8.44 show the forms giving optimum dimensions and reinforcement details of machine foundation respectively. It should be noted that dimensions of foundation are so large that the maximum bending moment is very less and therefore nominal reinforcement is provided as per the specifications given in the literature [89].

The dimensions suggested by machine manufacturer are 5.5m X 3.0m X 1.5m., whereas Genetic Evolution based software gives more than one alternative dimension sets giving the costs nearer to the optimum cost, to provide the options to select such dimensions that satisfy functional requirement without compromising with the safety and serviceability of the machine foundation.



DATA FOR VIBRATION ABSORBER

Dimensions of Spring Casing

Lx(m)	Ly(m)	Lz(m)
0.5	0.5	0.2

Properties of Spring Casing

Dia of Spring Wire 20 mm

Dia of Coil (D) mm	Permissible Load (P) Kg	Stiffness (Ks) Kg/mm
60	2028.47	768.52
70	1857.28	483.97
80	1706.31	324.22
90	1574.62	227.71
100	1459.86	166
120	1271.35	96.06
140	1124.19	60.5
160	1006.72	40.53
180	911.04	28.46
200	831.73	20.75

MACHINE DATA & SOIL DATA

Total Weight of Machine 6.0

Operating Speed of Machine 1200

Vertical exciting Force 1.5

Permissible Amlitude at Floor Level 0.04

Nature of Soil Stiff Clay

DATA REQUIRED FOR C.G. OF THE SYSTEM

Part	Wi (t)	Xi From Left for Mot and Right for Eng	Zi From Top of Foundation
Engine	4.0	2.0	1.05
Motor	2.0	1.5	0.75

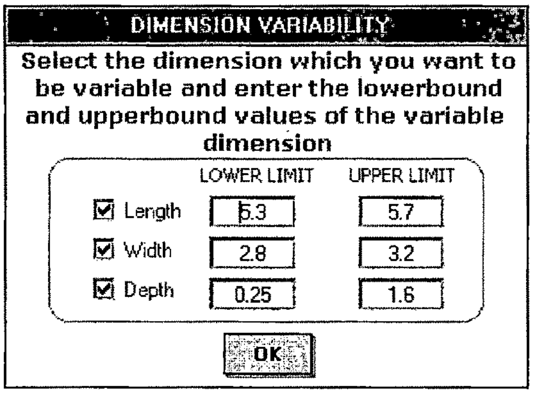
DIMENSIONS PROVIDED BY M/C MANUFACTURER

Length : 5.5

Width : 3.0

Height : 1.5

Fig 8.39 Input Data for Machine Foundation System

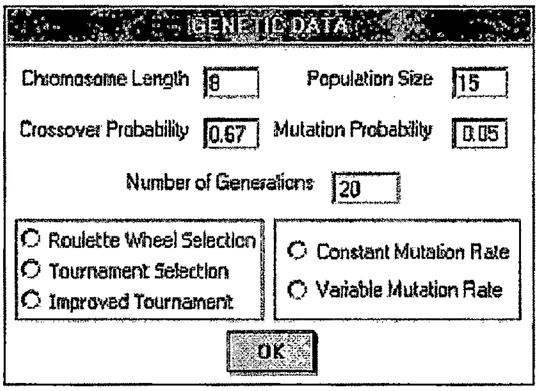


DIMENSION VARIABILITY

Select the dimension which you want to be variable and enter the lowerbound and upperbound values of the variable dimension

	LOWER LIMIT	UPPER LIMIT
☑ Length	5.3	5.7
☑ Width	2.8	3.2
☑ Depth	0.25	1.6

Fig. 8.40 Form for Defining Search Space



GENETIC DATA

Chromosome Length 8

Population Size 15

Crossover Probability 0.67

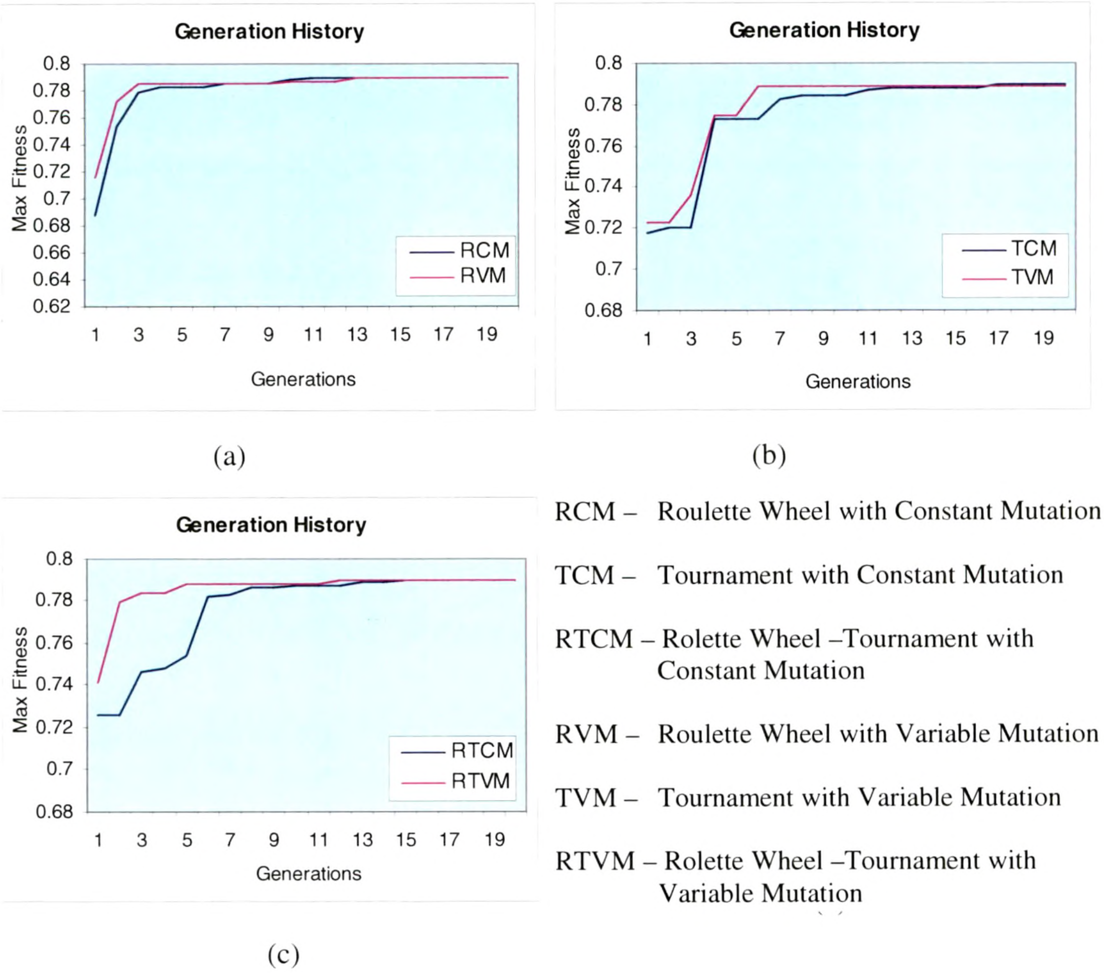
Mutation Probability 0.05

Number of Generations 20

○ Roulette Wheel Selection
○ Tournament Selection
○ Improved Tournament

○ Constant Mutation Rate
○ Variable Mutation Rate

Fig. 8.41 Form to Define GA Parameters



(a)

(b)

RCM – Roulette Wheel with Constant Mutation

TCM – Tournament with Constant Mutation

RTCM – Rolette Wheel –Tournament with Constant Mutation

RVM – Roulette Wheel with Variable Mutation

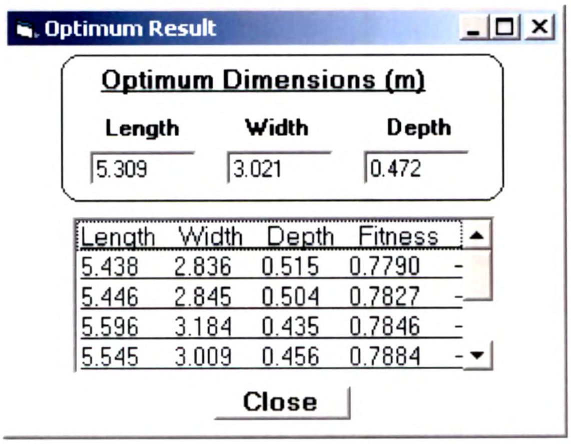
TVM – Tournament with Variable Mutation

RTVM – Rolette Wheel –Tournament with Variable Mutation

(c)

Fig. 8.42 Convergence Obtained by Different Selection Schemes

(a) Roulette Wheel (b) Tournament (c) Roulette Wheel – Tournament



Optimum Dimensions (m)

Length	Width	Depth
5.309	3.021	0.472

Length	Width	Depth	Fitness
5.438	2.836	0.515	0.7790
5.446	2.845	0.504	0.7827
5.596	3.184	0.435	0.7846
5.545	3.009	0.456	0.7884

Fig. 8.43 Form Showing Optimum Dimensions

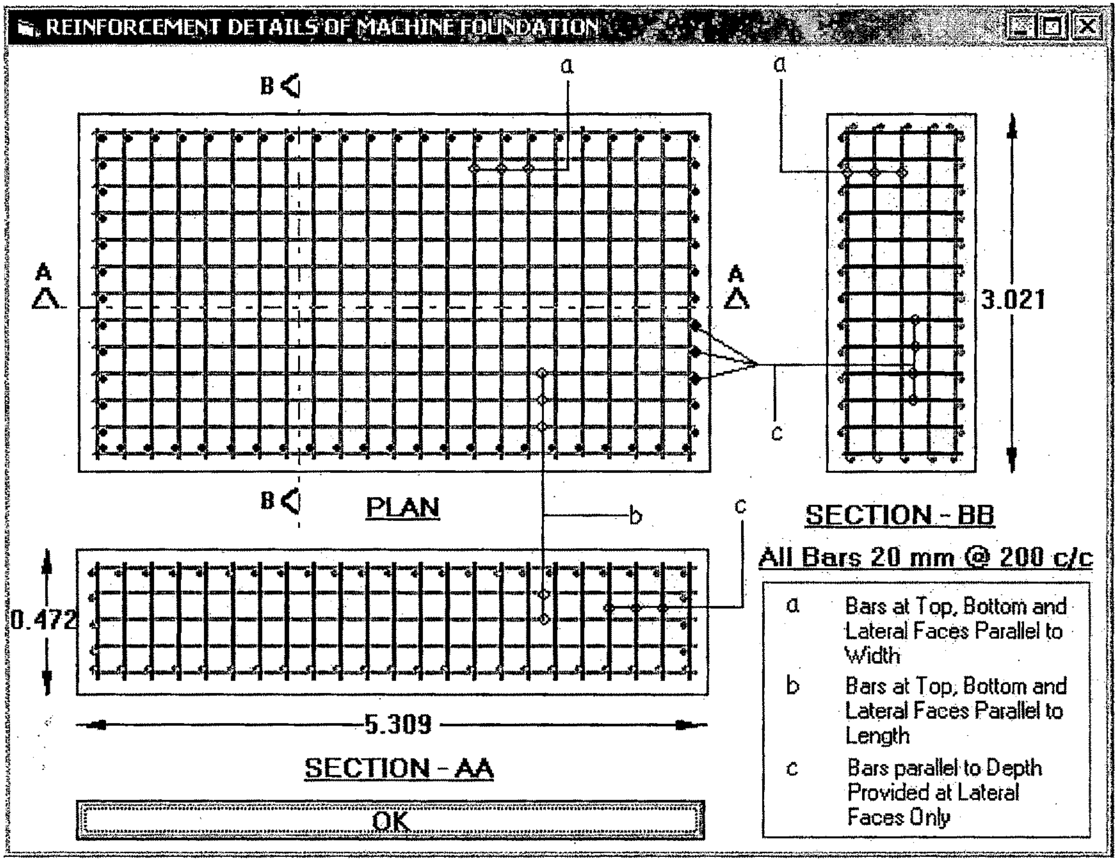

REINFORCEMENT DETAILS OF MACHINE FOUNDATION
a
a
B
A
A
3.021
c
B
PLAN
b
c
SECTION - BB
All Bars 20 mm @ 200 c/c
0.472
5.309
SECTION - AA
OK
a Bars at Top, Bottom and Lateral Faces Parallel to Width
b Bars at Top, Bottom and Lateral Faces Parallel to Length
c Bars parallel to Depth Provided at Lateral Faces Only

Fig. 8.44 Reinforcement Detailing for Machine Foundation

8.13 Weight Optimization of Gantry Girder

8.13.1 Objective Function

As gantry girder is a steel structure, the objective is to minimize the weight of the gantry steel which in turn would minimize the cost. The major contributors to the weight of gantry girder are the weight of beam section and weight of channel section. The objective function for such structures may be written as

$$f(\mathrm{W}) = \mathrm{W_B} + \mathrm{W_c} \qquad \ldots (8.62)$$

where $\mathrm{W_B}$ = weight of beam section in kg/m and $\mathrm{W_C}$ = weight of channel section in kg/m.

8.13.2 Constraints and Penalties

- **Bending Tensile Stress**

  The bending stress in tension should be less than the critical value of 0.66fy i.e 165 N/mm^2 for fy = 250 grade steel.

  Constraint : $\sigma_{bt} \leq 0.66\mathrm{fy}$ … (8.63)

  Penalty : $\sigma_{bt} / 0.66\mathrm{fy}$ … (8.64) where σ_{bt} = bending stress in tension in gantry girder.

- **Bending Stress in Compression**

The total bending stress in compression should be less than allowable bending stress in compression.

$$\sigma_{btotal} = \sigma_{bcv} + \sigma_{bch} \quad \ldots (8.65)$$

Constraint : $\sigma_{btotal} \leq \sigma_{bc}$ ... (8.66)

Penalty : $\sigma_{btotal} / \sigma_{bc}$ ... (8.67)

where σ_{bc} = allowable bending stress in compression in gantry girder, σ_{bcv} = bending stress in compression due to vertical loads, σ_{bch} = bending stress in compression due to horizontal loads, σ_{btotal} = total bending stress in compression due to combined, effect of σ_{bcv} and $\sigma_{bch.}$

- **Stresses Due To Longitudinal Force**

The longitudinal force causes an axial force with moment, so maximum stress f_{max} is calculated from,

$$f_{max} = W / A + M.y / I \quad \ldots (8.68)$$

where f_{max} = maximum stress induced due to longitudinal force, W = axial force developed due to the longitudinal force, A = total c/s area of gantry girder resisting longitudinal force, M = moment developed due to the longitudinal force, Y = distance from the C.G. of the gantry girder section, I = moment of inertia of the gantry girder section.

The maximum stress f_{max} should be less than the permissible bending stress in compression $\sigma_{bc.}$

Constraint : $f_{max} \leq \sigma_{bc}$ ... (8.69)

Penalty : f_{max} / σ_{bc} ... (8.70)

- **Shear stress**

The gantry girder should be safe from shear point of view and so the maximum shear stress ζ_{max} should be less than 0.4fy i.e 100 N/mm^2 for fy = 250 grade steel.

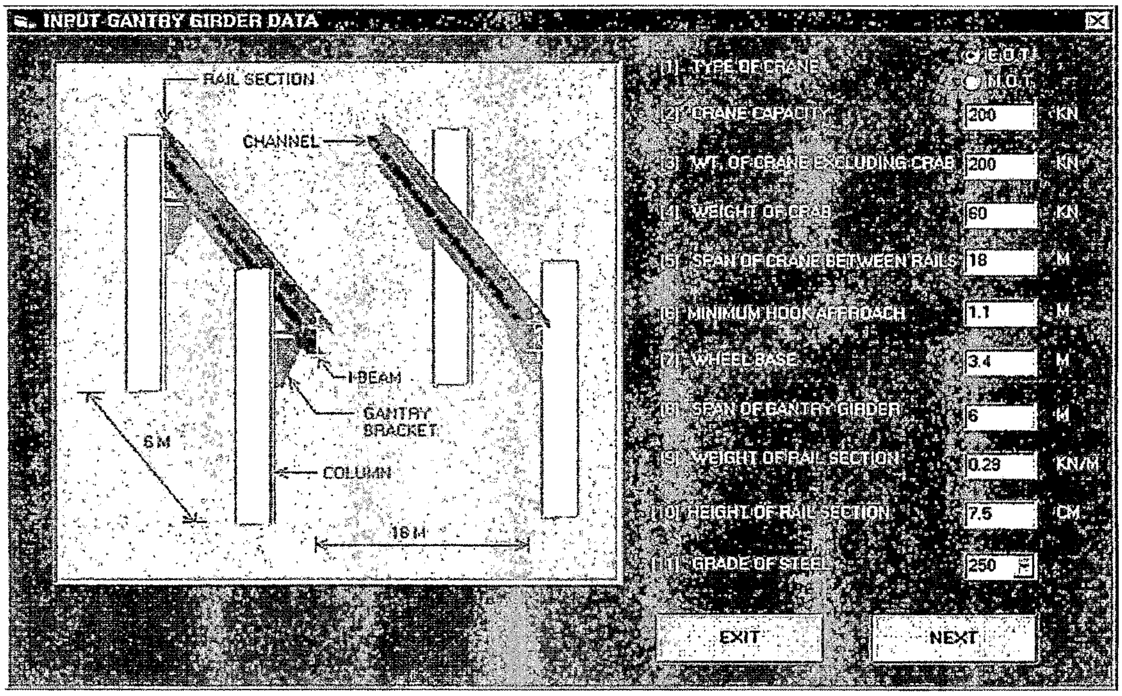
Constraint : $\zeta_{max} \leq 0.4fy$ ... (8.71)

Penalty : $\zeta_{max} / 0.4fy$ ... (8.72)

8.13.3 Screen Shots

The working of the software and example solved by the software is illustrated below.

1) Input gantry girder data



[1] TYPE OF CRANE ● E.O.T. ○ M.O.T.

[2] CRANE CAPACITY 200 KN

[3] WT. OF CRANE EXCLUDING CRAB 200 KN

[4] WEIGHT OF CRAB 60 KN

[5] SPAN OF CRANE BETWEEN RAILS 18 M

[6] MINIMUM HOOK APPROACH 1.1 M

[7] WHEEL BASE 3.4 M

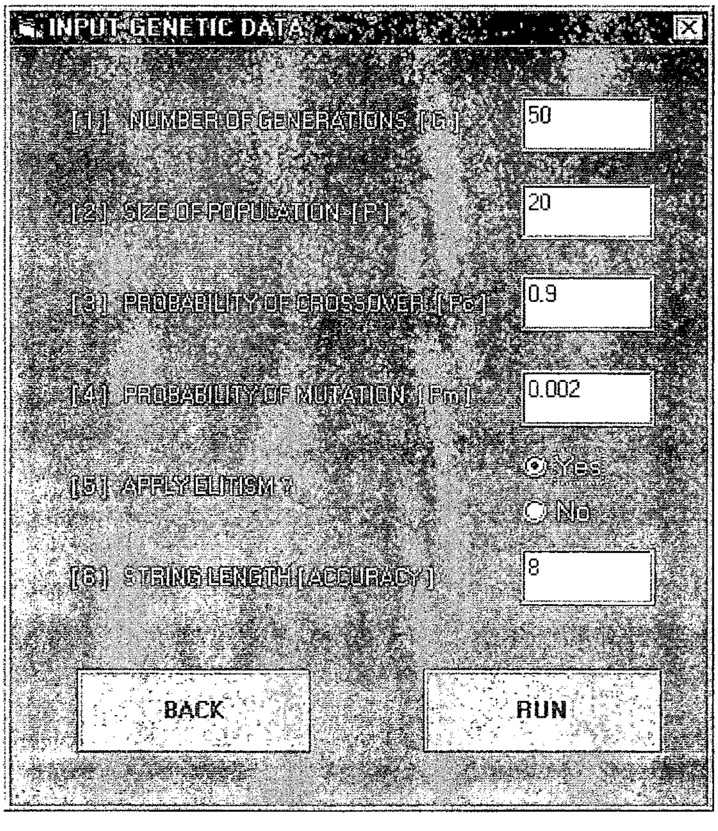
[8] SPAN OF GANTRY GIRDER 6 M

[9] WEIGHT OF RAIL SECTION 0.29 KN/M

[10] HEIGHT OF RAIL SECTION 7.5 CM

[11] GRADE OF STEEL 250

2) Input the genetic parameters



[1] NUMBER OF GENERATIONS [G] 50

[2] SIZE OF POPULATION [P] 20

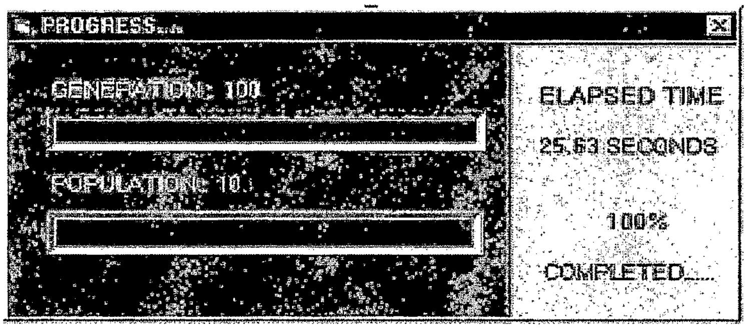
[3] PROBABILITY OF CROSSOVER [Pc] 0.9

[4] PROBABILITY OF MUTATION [Pm] 0.002

[5] APPLY ELITISM ? ● Yes ○ No

[6] STRING LENGTH [ACCURACY] 8

3) Run the program



GENERATION: 100

POPULATION: 10

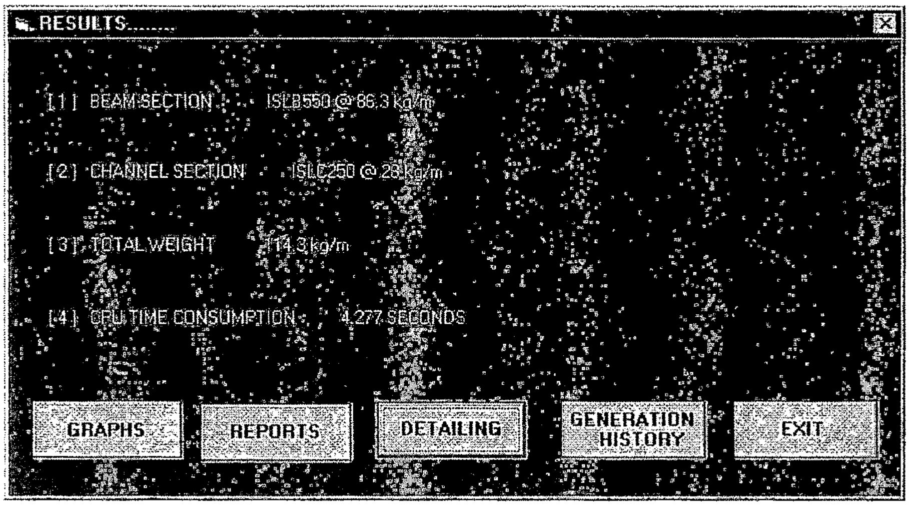
ELAPSED TIME

25.63 SECONDS

100%

COMPLETED......

4) View results



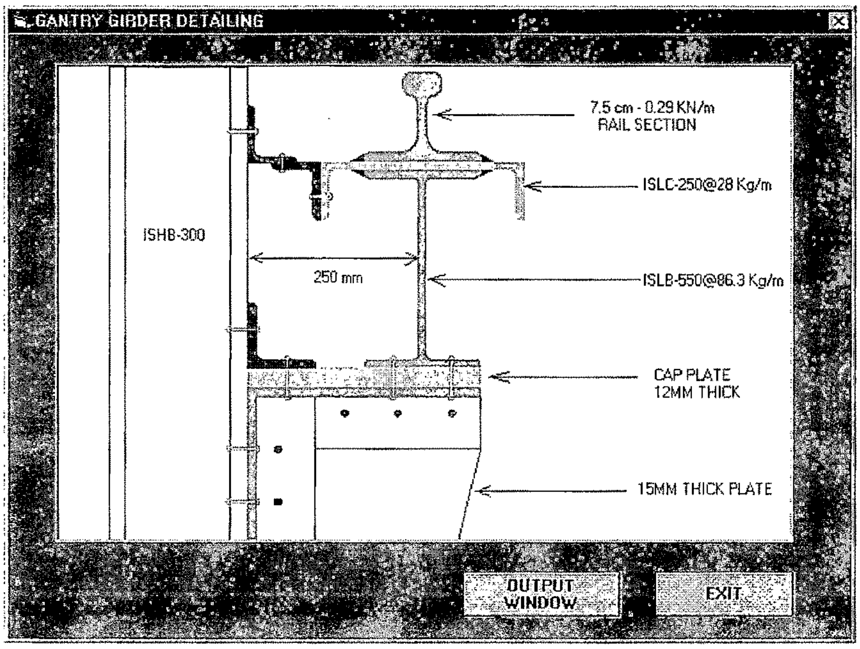
[1] BEAM SECTION ISLB550 @ 86.3 kg/m

[2] CHANNEL SECTION ISLC250 @ 28 kg/m

[3] TOTAL WEIGHT 114.3 kg/m

[4] CPU TIME CONSUMPTION 4.277 SECONDS

5) View the detailing



8.13.4 Example of Weight Optimization

To demonstrate the capabilities of the software a design problem with following data is solved and results are compared (Table 8.6) with those available.

1) crane capacity = 200 kN
2) weight of crane excluding crab = 200 kN
3) weight of crab = 60 kN
4) span of the crane between rails = 18 m
5) minimum hook approach = 1.1 m
6) wheel base = 3.4 m
7) span of gantry girder = 6 m
8) mass/weight of rail section = 0.79 kN/m
9) height of section = 7.5 cm
10) steel grade, fy = 250
11) young's modulus, E = 2 X 10^5 N/mm^2.

Table 8.6 Weight Comparison of Gantry Girder

Parameter	Results By Present Software	Result By Traditional Method [90]
Weight of I-beam section	ISLB550@86.3 kg/m	ISWB550@112.5 kg/m
Weight of channel section	ISLC250@28 kg/m	ISMC300@35.8 kg/m
Total weight of gantry	114.3 kg/m	148.3 kg/m

8.14 Weight Optimization of Plane Frames

In the optimum design of steel plane frames the member sizes are considered as design variables which are represented in binary form. When decoded the strings return the unique integer code numbers assigned to available steel sections (such as ISLB, ISMB, ISHB etc.). Thus the design variables are discrete in nature. The steel sections data is stored in the Microsoft Access database along with code numbers based on which section properties are imported and used for the analysis and evaluation of constraints. The advantage of doing so is reduced length of string and ease of finding the section from the database. For example, there are 13 number of ISMB sections available, hence a 4 bit string is enough to generate 16 distinct integers which otherwise would not be sufficient if sections were selected based on cross sectional area or section modulus. Further one has to find a local range from decoded cross-sectional area, to find the section having cross sectional area falling in the range using the formula.

$$V_{ar} = \min + \left[\frac{D}{2^{l}-1} \times [\max - \min] \right] \qquad \ldots (8.73)$$

where V_{ar} = mapped value of decoded string, Min = Minimum actual value of design variable in search space, Max = Maximum actual value of design variable in search space, D = decoded value of string and l = length of string.

8.14.1 Constraints

The objective function is the weight of the frame which can be obtained by adding the individual member weights. The design of beam and column follows the IS: 800 [91] provisions. The main constraint in this case is stress in the element, slenderness ratio and provision of minimum section and thickness etc. The constraints considered are as follows:

I For Beams

(i) **Bending stress constraint**: The maximum bending stress in tension (σ_{bt},cal) or in compression (σ_{bc},cal) in extreme fibre, calculated on the effective section of beam, shall not exceed the maximum permissible bending stress in tension (σ_{bt}) or in compression (σ_{bc}).

Constraint : $\sigma_{bc,Cal}$ or $\sigma_{bt,Cal} \leq 0.66 f_y$ ... (8.74)

Constraint Function : max (($\sigma_{bc,Cal}$ or $\sigma_{bt,Cal}$ $0.66 f_y$) -1, 0) ... (8.75)

For symmetrical sections,

$$\sigma_{bc} = \sigma_{bt} = M/Z \qquad \ldots (8.76)$$

Where, M = maximum bending moment due to actual loading in the member and Z = section modulus in the direction of the applied moment.

(ii) **Shear stress constraint**: The average shear stress (τ_{va}) in a member calculated on the cross section of the web shall not exceed 0.4 f_y

Constraint : $\tau_{va} \leq 0.4\ f_y$ ... (8.77)

For I section τ_{va} can be calculated as

$$\tau_{va} = \frac{V}{t_w \text{ x } h} \qquad \ldots (8.78)$$

where τ_{va} = average shear stress, V = maximum shear force on member, t_w = thickness of web and h = depth of web (overall depth of beam).

Constraint Function : max (τ_{va} / 0.4 f_y – 1, 0) ... (8.79)

II **For Columns**

(i) **Slenderness ratio constraint:** The maximum slenderness ratio of compression member shall not exceed 180. The slenderness ratio, denoted by λ can be calculated by the equation,

$$\lambda = l/r \qquad \ldots (8.80)$$

where l = effective length of member which depends on the end conditions of member, and r = appropriate radius of gyration.

Constraint : $\lambda < 180$ ... (8.81)

Constraint Function : max (λ/180-1, 0) ... (8.82)

(ii) **Combined stress constraint:** Member subjected to axial compression and bending (uniaxial bonding in this work) shall satisfy the following condition,

$$\chi = \frac{\sigma_{ac}\,,\text{Cal}}{\sigma_{ac}} + \frac{C_m}{\left[1 - \frac{\sigma_{ac}\,,\text{Cal}}{0.6\ f_{ccx}}\right]\sigma_{bc}} \le 1.0 \qquad \ldots (8.83)$$

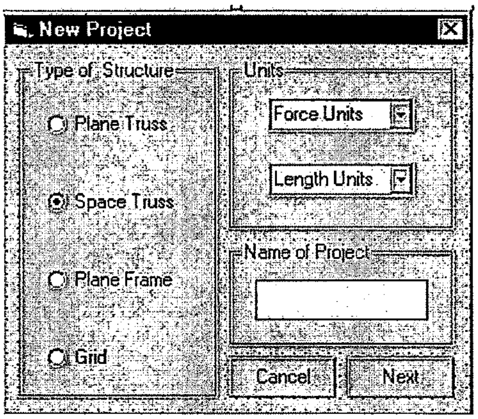
where, σ_{ac},cal = Calculated average axial compressive stress, σ_{bc},cal = Calculated bending compressive stress at extreme fibre, σ_{ac} = permissible axial compressive stress in the member, f_{ccx} = elastic critical stress in compression = $\pi^2 E/\lambda^2$ and C_m = a coefficient taken as 0.85 (IS:800)

Constraint Function: $\chi - 1.0 \le 0$ ... (8.84)

8.14.2 Solution Steps

Availability of various forms, menus, tool boxes to solve any problem makes the use of software quite simple. Following steps are to be executed for solving any problem.

(i) Select plane frame option from the new project dialog box as shown in Fig. 8.45 and press OK button which displays preprocessor forms for plane frame.

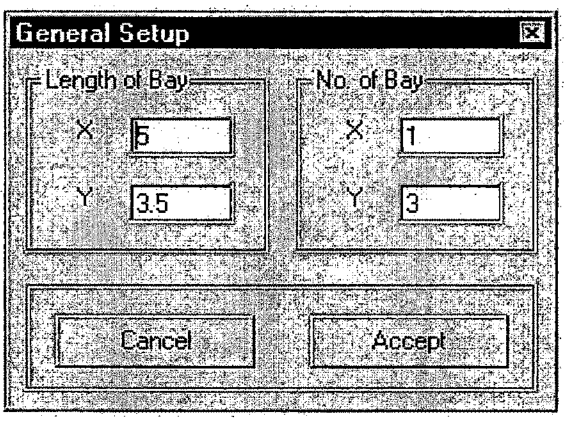

New Project

Type of Structure: ○ Plane Truss ● Space Truss ○ Plane Frame ○ Grid

Units: Force Units; Length Units

Name of Project:

Fig. 8.45 New Project Form


General Setup

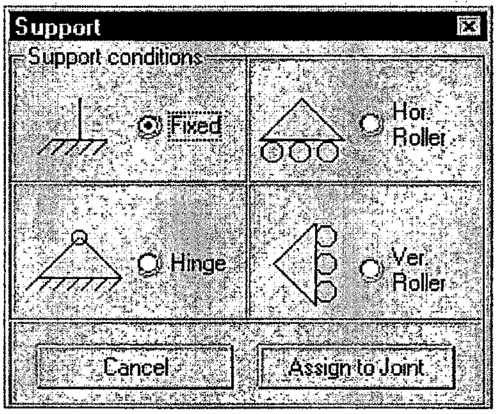
Length of Bay: X 5; Y 3.5

No. of Bay: X 1; Y 3

Fig. 8.46 Form for Creating Geometry

(ii) Structural data of plane frame can be generated by menu asking number of bays and storeys and their dimensions (Fig. 8.46). On clicking Access button the frame with joint and member number displayed on Z, will be drawn.

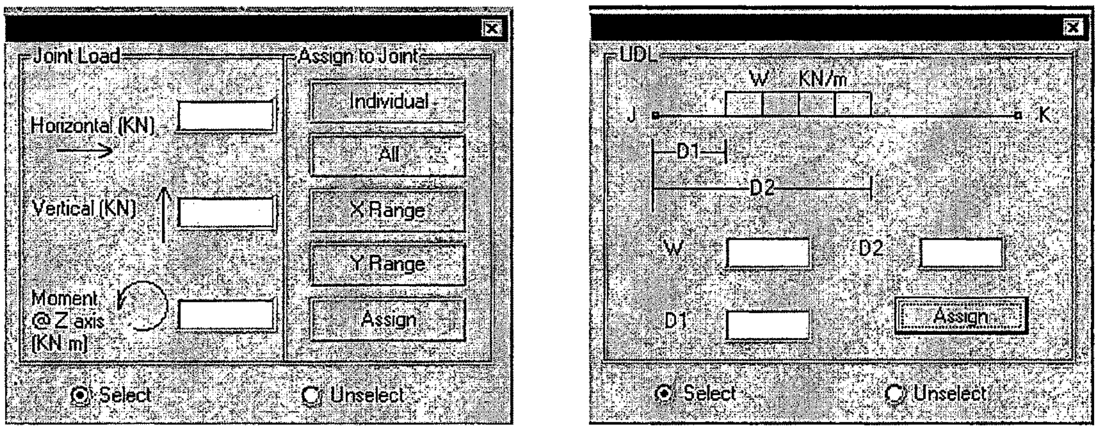
(iii) Support condition can be assigned to a joint by selecting appropriate support from a dialog box shown in Fig. 8.47 which is displayed on clicking the support menu from member.


Support

Support conditions: ● Fixed ○ Hor. Roller ○ Hinge ○ Ver. Roller

Fig. 8.47 Form for Assigning Support

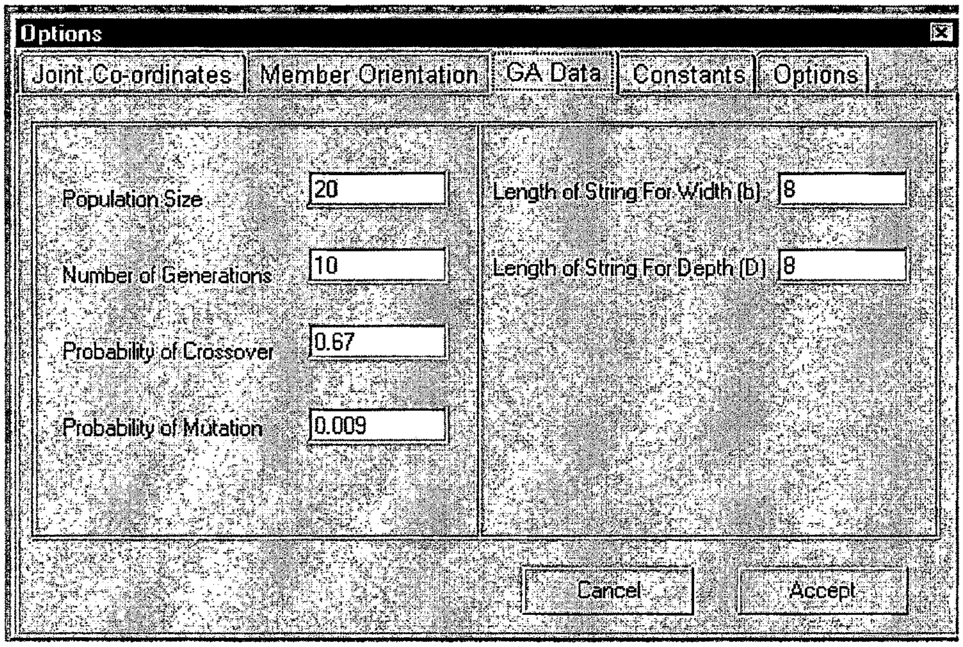
(iv) The joint load and member load (UDL or concentrated load) can be specified by separate forms as shown in Figs. 8.48 (a) and (b).



(a) Joint Load **(b) Member Load**

Fig. 8.48 Form for Assigning Load

(v) The data related to GA and design data are supplied in the form containing SSTab control as shown in Fig. 8.49. Upon clicking the steel design menu optimization process will start.



Population Size	20
Number of Generations	10
Probability of Crossover	0.67
Probability of Mutation	0.009
Length of String For Width (b)	8
Length of String For Depth (D)	8

Fig. 8.49 Form for Supplying GA Data

Results will be tabulated for each member showing number of generations and populations. Optimum section will be written on the geometry to get the idea of optimum results at a glance. Following example demonstrate the capabilities of the software.

8.14.3 A Plane Frame Example

A 3 storey single bay plane frame problem subjected to the loading and support condition as shown in Fig.8.50 is taken as first example. Number of design variables is same as number of members i.e. 9 (6 columns and 3 beams). Young's modulus for steel is considered as 2 x 10^5 N/mm^2. The height of floor and bay width are taken as 3.5 m and 5 m respectively. The program selects Standard ISHB sections for columns and Standard ISLB, ISMB and ISWB sections for beams. The data related to GA are: (i) length of string = 6, (ii) population size = 20, (iii) maximum generations = 30, (iv) crossover probability = 0.78 and (v) mutation probability = 0.03. Tournament selection scheme is used. As design variables are discrete and taking maximum 48 discrete values, 6 bit long string is sufficient. Elitism has been used here to force the search process to achieve higher and higher goal in the succeeding generations. The final results shown in Fig. 8.51 are the outcome after 4 GA runs.

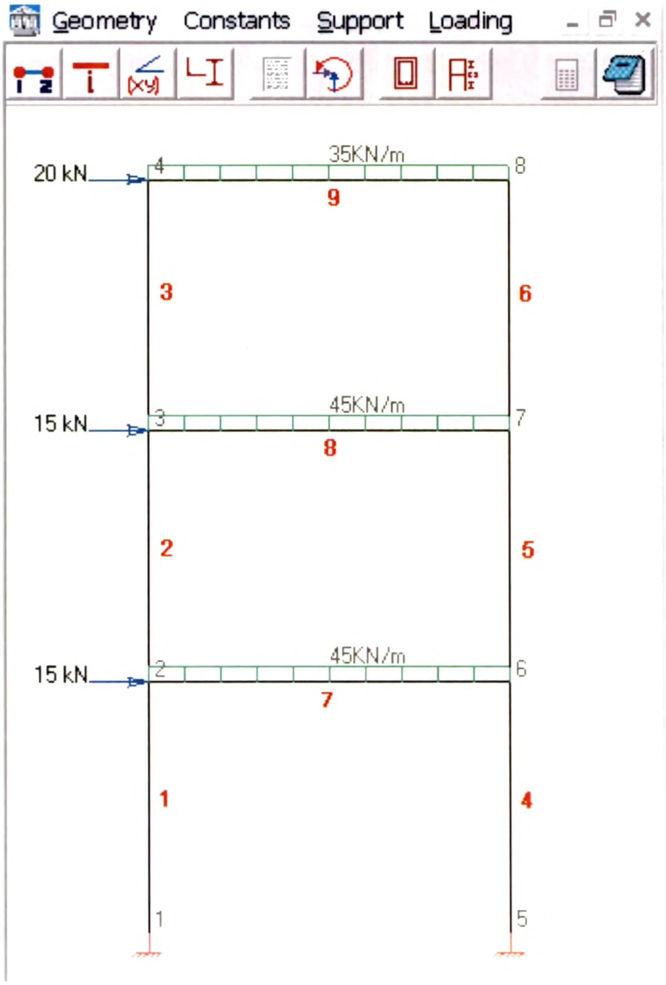
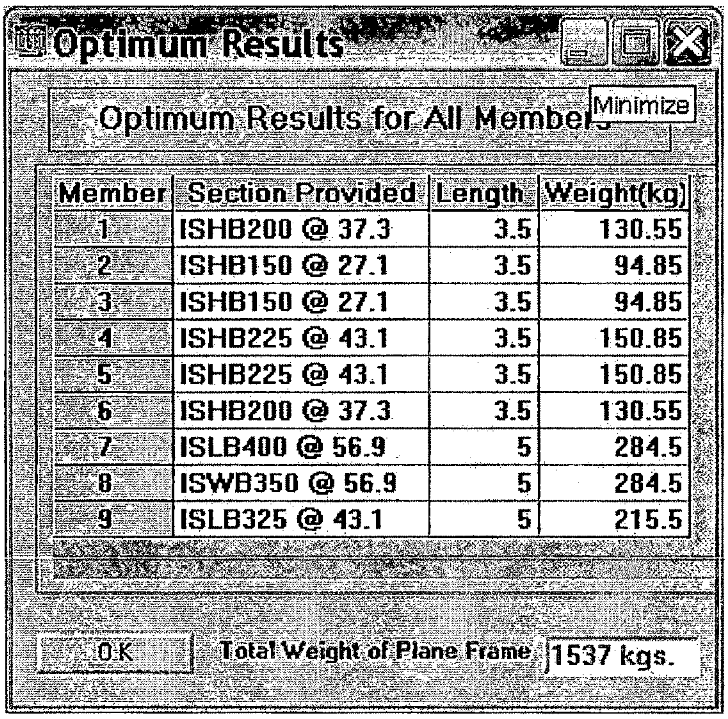


Fig. 8.50 Screenshot Showing Geometry of the Plane Frame in the Software



Member	Section Provided	Length	Weight(kg)
1	ISHB200 @ 37.3	3.5	130.55
2	ISHB150 @ 27.1	3.5	94.85
3	ISHB150 @ 27.1	3.5	94.85
4	ISHB225 @ 43.1	3.5	150.85
5	ISHB225 @ 43.1	3.5	150.85
6	ISHB200 @ 37.3	3.5	130.55
7	ISLB400 @ 56.9	5	284.5
8	ISWB350 @ 56.9	5	284.5
9	ISLB325 @ 43.1	5	215.5

Total Weight of Plane Frame 1537 kgs.

Fig. 8.51 Final Results Showing Total Weight of the Structure

8.15 OPTIMIZATION OF PLANE TRUSSES IN GENERAL

8.15.1 Size Configuration and Topology Optimization

In size optimization of trusses, cross-sectional areas of members are considered as design variables and the coordinates of the nodes and the connectivity among the various members are considered as fixed. The resulting problem is a nonlinear programming problem. The sizing optimization problem is made practically useful by restricting the member cross-sectional areas to take only certain pre-specified discrete values (commercially available sections).

In configuration optimization of trusses, the nodal coordinates are kept as design variables. In this category number of members, number of joints, member connectivities and support conditions are held unchanged. Simultaneous optimization of sizing and configuration is another possibility. The resulting problem is also a non linear programming problem with member area and nodal coordinates as variables.

In topology optimization of trusses, the nodal coordinates, the member cross-section areas, the member connectivity, the presence of absence of a node, the presence or absence of a member, restraints positions and conditions etc. are treated as design variables. Classical optimization methods cannot be used adequately in topology optimization because they lack the efficient ways to represent member connectivity in a truss and they are incapable to handle the combination of continuous and discrete variables of the resulting non-convex and discontinuous design space.

Although the above three optimization problems are separate in nature, the most efficient way to design a truss is to consider all the three aspects simultaneously. Most of the optimization attempts focus on use of multi-level optimization. In such a method, when topology optimization is considered, member areas and the truss configuration is assumed to be fixed. Once an optimized topology is found, the member area and configuration of obtained topology is optimized. But such a multilevel optimization technique may not always provide a globally optimum design, since all the three problems are not linearly separable.

Most of the methods using GA for topology optimization use the "ground structure" concept [14, 17, 20]. The use of conceptual design, based on ground structures makes the design problem amenable to solutions, but, in this approach the generated design would be strongly influenced by the conceptual designs. This class of the problems which are best described as "structured" optimization problems are characterized by overall designs that are preconceived at the outset. If on the other hand, the design problems are kept free of conceptual designs, there is a potential for generating more efficient and innovative designs, especially when more complex design problems are attempted. However, the design freedom thus achieved makes the problem more difficult to solve. This class of problems, which may be best described as "unstructured" optimum design problems are characterized by designs that are allowed to emerge free of preconceived designs. A methodology for solving this class of problem is also described in the present chapter.

8.15.2 Problem Formulation

In the context of GA, the general truss design problem is posed as:

$$\begin{aligned} \text{Find,}\quad & x\,(x_s, x_c, x_t), \\ \text{to minimize,}\quad & f(x) + f_{\text{penalty}}, \\ \text{subject to,}\quad & g_i(x) \leq 0 \qquad i = 1, 2, 3, \ldots, n, \\ & x_j^L \leq x_j \leq x_j^U \; j = 1, 2, 3, \ldots, k \end{aligned} \qquad \ldots (8.85)$$

where, x = vector of design variables which consists of, x_s = size variables, x_c = configuration variables and x_t = topology variables; $f(x)$ = objective function; $f_{penalty}$ = weight penalty for unacceptable (unfeasible) design; $g_i(x)$ = performance constraints; x_j^L and x_j^U = lower and upper bound of the design variables; n = number of constraints; and k = number of design variables.

The objective function for truss problem is the structural weight, expressed as:

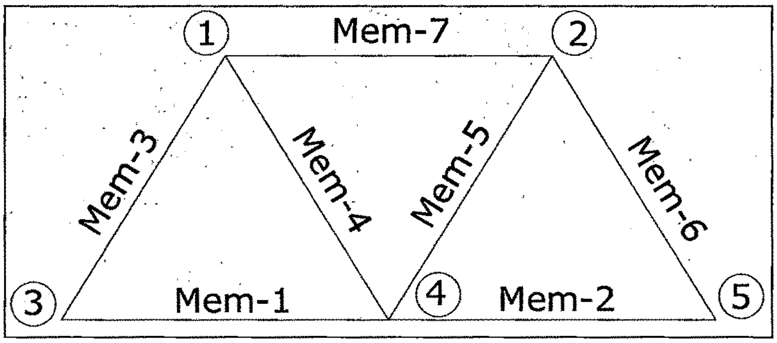
$$f(x) = \sum_{i=1}^{ns} \rho A_i L_i \qquad \ldots (8.86)$$

where, ρ = unit weight of material, A_i = cross sectional area of i^{th} member, L_i = length of i^{th} member and ns = number of sizing variables.

8.15.3 Design Variables and their Representation

8.15.3.1 Size optimization

In size optimization problems cross-sectional areas are considered as design variables whose optimized values are obtained to get minimum weight of truss. Number of variables depends on number of truss members. Each variable is represented by a sub- string of predetermined number of binary characters (i.e. 0 or 1). The solution string consists of sub-strings in numbers equal to number of design variables. That is, a solution string for a truss having n members contains n substrings concatenated physically. Thus for a truss having seven members the binary solution string consists of seven sub-strings as shown in Fig. 8.52.



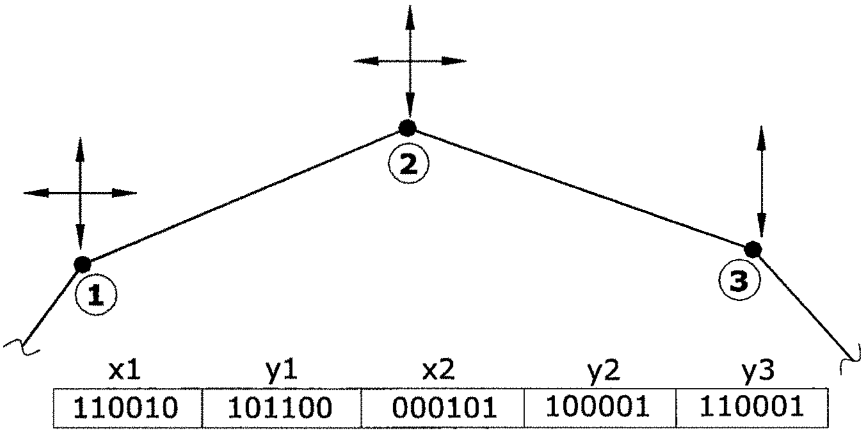
member 1	member 2	member 3	member 4	member 5	member 6	member 7
1010101	0001110	0101010	0011001	0111001	1110001	0011111

Fig. 8.52 Representation of Size Variables

In the symmetrical truss the number of variables can be reduced by grouping the symmetric members.

8.15.3.2 Configuration optimization

For configuration optimization modifying nodal coordinates results in a global variation of the structural design shape. Binary coded string is used for the configuration optimization. Each string contains a finite sub string representing x and y coordinates of a particular joint. The joints having a load point or a support conditions are considered as immovable points (the coordinates of these joints do not change) while other joints can be defined as movable or partially movable. However, some of them can also be made immovable if required. Partially movable joints are allowed to move either horizontally or vertically. Following is the example (Fig. 8.53) of a binary string for joint 1 movable in x and y direction and joint 2 movable in y direction only.



x1	y1	x2	y2	y3
110010	101100	000101	100001	110001

Fig. 8.53 Representation of Configuration Variables

Length of each sub-string can be selected depending on the desired solution accuracy. In the simultaneous size and configuration optimization solution strings of Figs. 8.52 and 8.53 are concatenated to represent the shape of the truss and areas of the members in combination.

8.15.3.3 Topology optimization

In the topology optimization problems both nodes and members can be used to represent the skeletal structures in strings. Here, only nodes are explicitly represented in fixed-length strings, using binary coding. Each string encodes information on a fixed number of nodes in the physical design space. The nodal information represented is: (i) nodal active/inactive status and (ii) nodal coordinate in X and Y directions.

The space around each node is discretized into a number of plane angular sectors as shown in Fig. 8.54, each possessing a specific set of member properties and a measure of its priority with respect to other sectors. Each member passes through two sectors, one at each end node, and inherits its properties from the sector with higher priority, or the dominant sector. Each sector around a node is uniquely related to a specific sector around another node such that a member connecting the two nodes passes through the related sectors, provided the number and size of sectors around each node is equal. The size of the sectors should be kept reasonably small in order to minimize the possibility of two or more members passing through the same sector and inheriting the same member properties set. The member information set in each sector consists of: (i) member active/inactive status; (ii) sector priority; and (iii) member type as shown in Fig 8.55.

The string representing the structure is made up of fixed number of identical sub strings, each corresponding to a specific node. The use of fixed-length strings does not limit the number of nodes in a structure, since the strings can be initially specified to any desired length. The present string representation scheme does not impose an upper limit on the string length. Within the bound of the string length used, the synthesized structures can acquire any number of the nodes and members to switch themselves ON or OFF. Only active nodes, as determined from nodal active and inactive genes are used to synthesize the represented structure. The active nodes are connected to each other forming a fully connected structure. The members inherit their properties from the dominant sector. Some of the members may be inactive depending upon the member active/inactive genes. The final synthesized structure comprises of all active nodes and active members.

If the generated structures are unstable or they contain freely "hanging" nodes or completely separated parts, these deficiencies are eliminated during evolution through selection pressure alone. No special measures are taken for this purpose. Decoding of the solution string representing topology is explained below.

The genes representing nodal coordinates and member types are first transformed to their raw values through binary to integer transformation. They are then converted to their actual values by means of pre-assigned mappings between the possible raw values and the actual values. The sector priorities are obtained through binary to integer transformation.

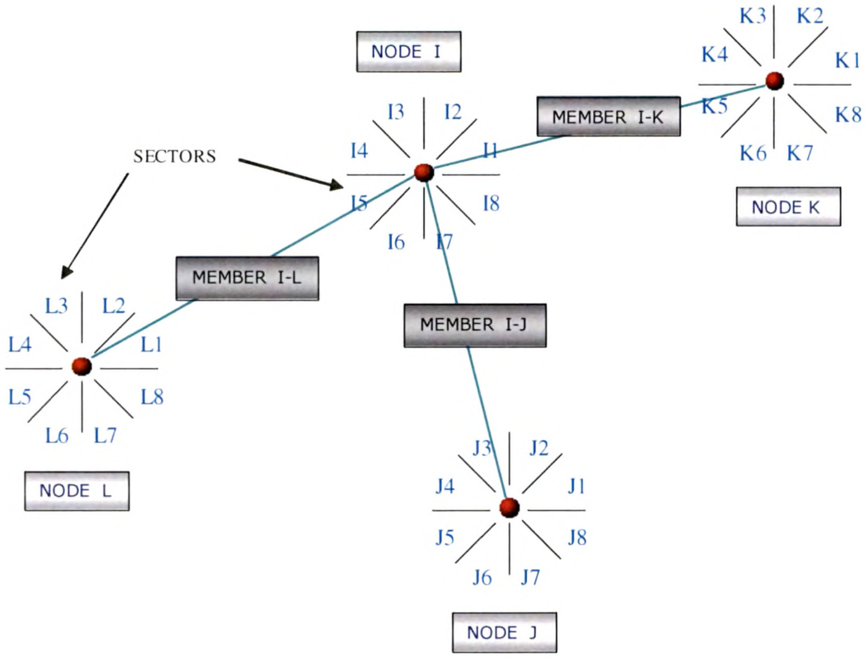
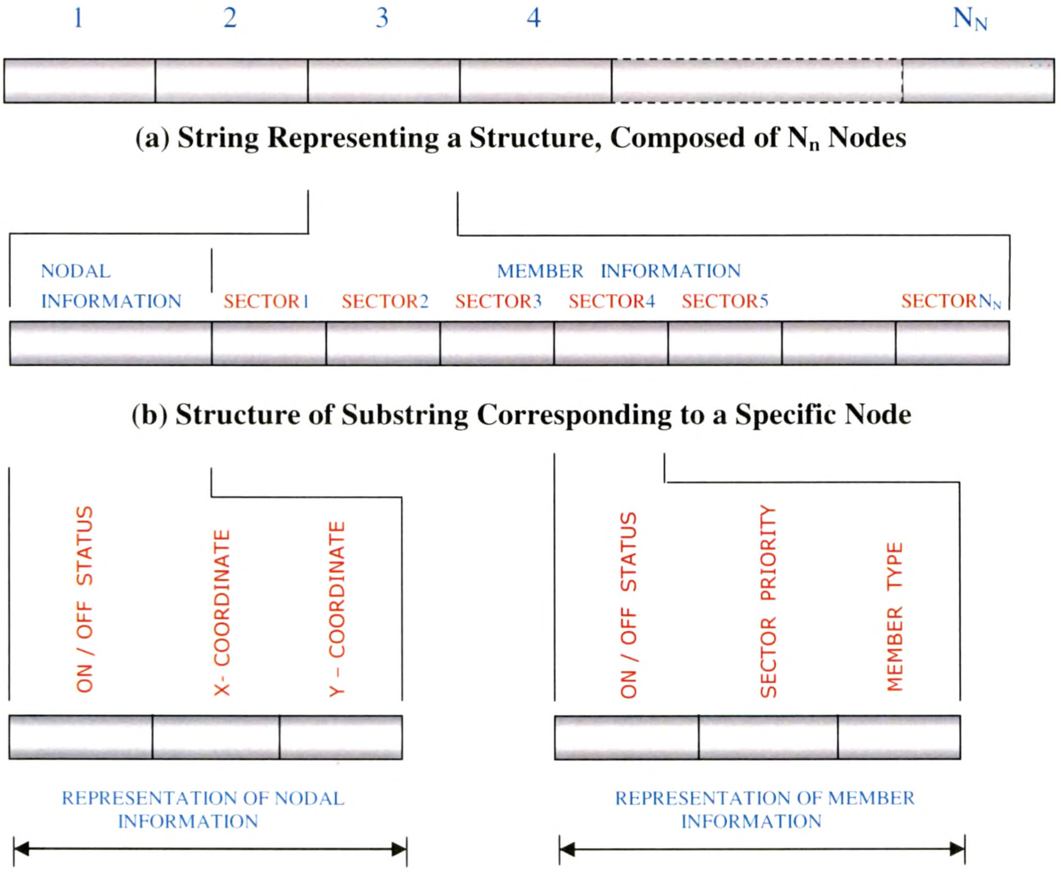


Fig 8.54 Sectorial Representation Scheme for Member Properties



(a) String Representing a Structure, Composed of N_n Nodes

(b) Structure of Substring Corresponding to a Specific Node

(c) Representation of Nodal and Member Information

Fig. 8.55 The String Representation Scheme for Topology Optimization

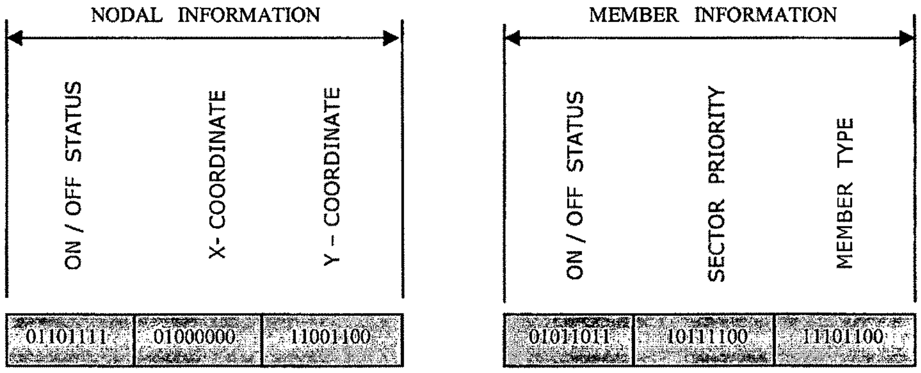
A node of a member is considered active if the proportion of the ON/OFF bits containing "1" in the corresponding genes at least equals to the threshold factors, f^{N}_{on} for nodes and f^{M}_{on} for members. The threshold factors can be assigned a value between 0 and 1 in order to control the proportion of nodes and members active in the initial population. The use of more than one bit in the ON/OFF genes allows the possibility of the crossover sites falling within these genes. The genes representing fixed and partially fixed nodes are located at known fixed positions on all the strings. These nodes are assumed to be active and their fixed coordinates are assigned the specified values instead of the values encoded in the strings or in other words these values overrules the values obtained by decoding the strings. The fixed members are also treated similarly.

To demonstrate the proposed method of string representation and decoding of the represented information an example of 20 m wide and 10 m height design space is considered here. The string comprises of 10 identical sub strings, numbered 1 through 10, each representing a potential node in the rectangular design space. Three sub strings, 5, 8 and 10, are specified to represent fixed nodes located on the base of the physical design space. Sub strings 8 and 10 represent the support nodes located at the corners and sub string 5 represents the loaded node located at the center. The remaining 7 sub strings represent variable nodes. Each sub string contains one nodal information set and 8 members information set each corresponding to a sector around that particular node. Figure 8.56 shows details of nodal and member information sets, their structure, binary representation, raw transformed values, and final normalized values. Each component of nodal and member information sets is represented by 8 binary bits.

The decoded nodal information in the sub strings is shown in Table 8.7. Of the seven sub strings representing variable nodes, only 2 and 7 are active (i.e. "ON"). Of the three sub strings representing fixed nodes, string 8 is inactive, but as all the fixed nodes must be present in the synthesized structure, this sub string is treated as active. The specified coordinates of the fixed nodes are used instead of the values encoded from the string. The string is synthesized from nodes represented by the 5 active sub strings. All such nodes are fully connected by members forming an intermediate structure (Fig. 8.57 (a)). Each member passes through two sectors, one at each end. Table 8.8 lists all the members, sectors through which they pass, and the member information represented in those sectors. The members inherit member properties from the sector with higher priority. For example, member 1-2 passes through sector 1 of sub string 8 and sector 5 of sub string 5; it inherits its member properties from sector 1 of sub string 8 because it has a higher priority (212) compared to sector 5 of sub string 5, which has a lower priority (131). Based on member information represented in dominant sectors, some of the members are active and others inactive. For example, member 1-2 is active because its dominant sector shows its status to be "ON"; member 1-3 is inactive because its dominant sector shows its status to be "OFF". Thus, only members 1-2, 1-4, 2-3, 2-4, 2-5, 3-5, and 4-5 are active and, therefore, present in the synthesized structure. The final synthesized structure is shown in Fig 8.57 (b).

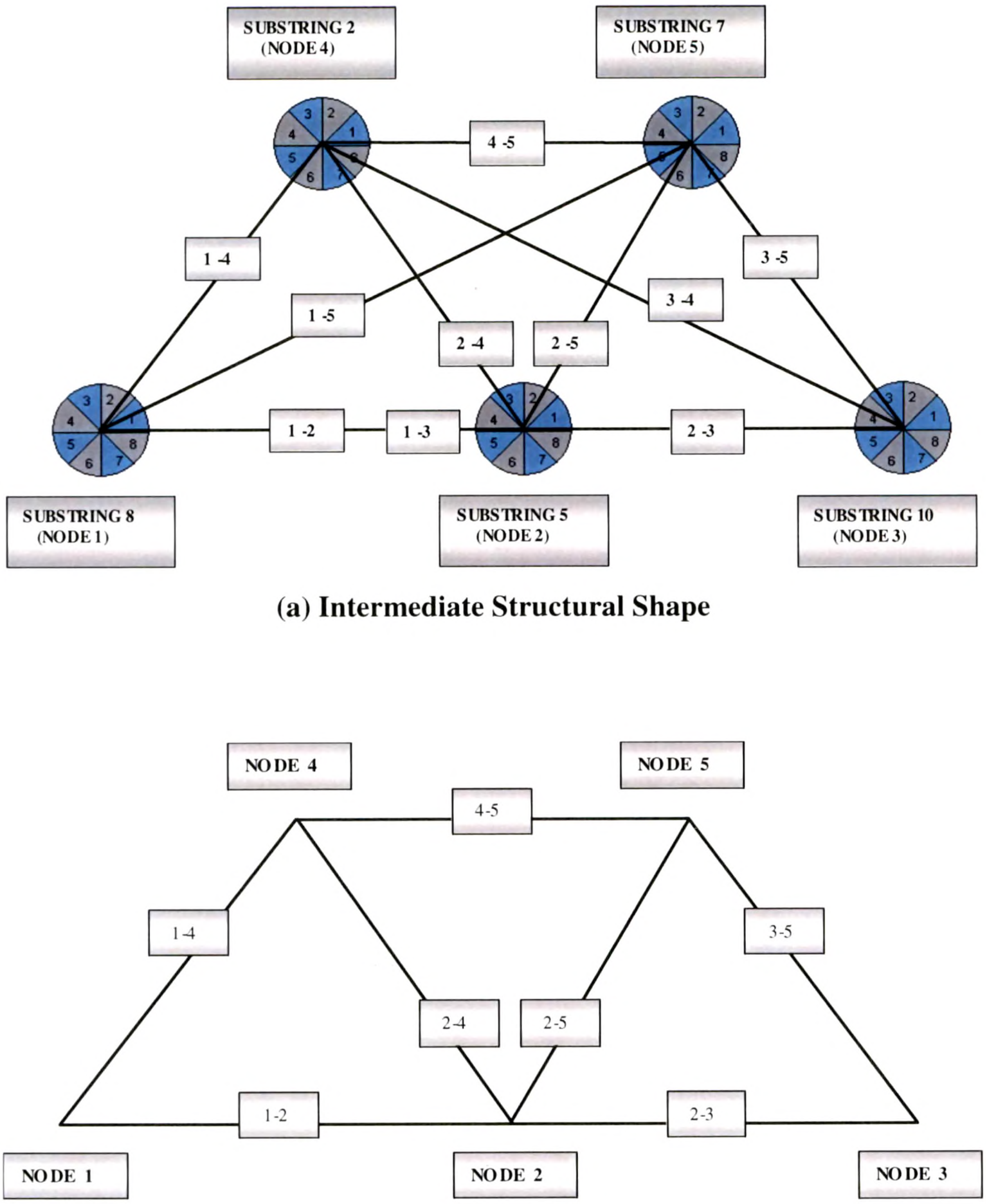
Table 8.7 Example of string representation and structural synthesis schemes

Substring no. (1)	Nodal on/off Status (2)	X coordinate (3)	Y coordinate (4)	Structural node no. (5)
1	**OFF**	**18.43**	**3.69**	**-***
2	**ON**	**5.02**	**8.00**	**4**
3	**OFF**	**11.29**	**9.10**	**-***
4	**OFF**	**3.92**	**9.29**	**-***
5	**ON**	**9.04**	**0.48**	**2**
6	**OFF**	**8.47**	**2.31**	**-***
7	**ON**	**15.06**	**8.00**	**5**
8	**OFF**	**0.64**	**0.32**	**1**
9	**OFF**	**4.08**	**7.61**	**-***
10	**ON**	**19.44**	**0.40**	**3**
*** Inactive OFF nodes do not exist in the synthesized structure.**				



RAW VALUES:	6	64	204	5	188	236
NORMALIZED VALUES:	ON	5.02	8.00	ON	188	25

Fig. 8.56 Node and Member Information Representation and Their Interpretation



(a) Intermediate Structural Shape

(b) Final Structural Shape

Fig. 8.57 Intermediate and Final Structural Shape Synthesized from the Solution String

Final structural shape thus obtained is analyzed to find the member forces which are then used for calculation of member stresses and evaluation of constraints. Weight of the truss is calculated using Eq. (8.86) to get the objective function.

Table 8.8 Example of String Representation and Structural Synthesis Scheme

(List of Members and Member Properties Represented at Their End Nodal Sectors)

First End Node					
Member **(1)**	**Substring No.** **(2)**	**Sector No.** **(3)**	**Sector Priority** **(4)**	**Mem ON/OFF Status** **(5)**	**Section Type** **(6)**
1-2	8	1	212	ON	2
1-3	8	1	121	ON	22
1-4	8	2	98	OFF	23
1-5	8	1	179	OFF	21
2-3	5	1	136	OFF	25
2-4	5	3	184	ON	26
2-5	5	2	235	ON	26
3-4	10	4	132	OFF	22
3-5	10	3	185	ON	20
4-5	2	1	247	ON	19

Second End Node				
Substring No. **(7)**	**Sector No.** **(8)**	**Sector Priority** **(9)**	**Mem ON/ OFF Status** **(10)**	**Section Type** **(11)**
5	5	131	ON	24
10	5	197	OFF	24
2	6	203	ON	20
7	5	101	OFF	18
10	5	245	ON	27
2	7	59	ON	27
7	6	156	ON	24
2	8	188	OFF	25
7	7	87	OFF	22
7	5	129	ON	17

8.15.4 The Constraints

Size and Configuration Optimization: In the size optimization problems following four constraints are imposed to get the feasible optimum solution.

(a) **Geometric constraint for cross section area**: The search space is confined by this constraint which is given by,

$$A_i^L \leq A_i \leq A_i^U, \qquad \text{... (8.87)}$$

where A_i^L and A_i^U are lower and upper bound values of the cross sectional area for ith member. This constraint is handled by selecting the upper and lower bound values of the size variables.

(b) **Stress constraint**: This constraint reduces the probability of selection of the string in which the actual member stress σ_j is more than allowable value σ_{aj}. The constraint can be written as:

$$\sigma_i \leq \sigma_{ai}, \qquad \text{... (8.88)}$$

and the corresponding constraint function is,

$$g_j(x) = \max(\sigma_i / \sigma_{ai} - 1, 0). \qquad \text{... (8.89)}$$

(c) **Displacement constraint**: If d_i and d_{ai} as actual and permissible displacements the constraint is,

$$d_i \leq d_{ai}, \qquad \text{... (8.90)}$$

and the corresponding constraint function is,

$$g_j(x) = \max(d_i / d_{ai} - 1, 0). \qquad \text{... (8.91)}$$

(d) **Nodal co-ordinates constraints**: This constraint is applicable to the configuration optimization only, where the structural shape is allowed to change by varying nodal co-ordinates. But the co-ordinates are allowed to take the value from the range prescribed by the designer as per,

$$x_k^L \leq x_k \leq x_k^U \text{ and,} \qquad \text{... (8.92)}$$

where x_k and y_k are x and y coordinates of kth joint (node) and $x_k^L, x_k^U,$, y_k^L and y_k^U are their lower and upper bound values. This constraint is handled by selecting appropriate values of lower and upper bound values.

Topology Optimization: In the topology optimization the constraints are handled in slightly different manner and are therefore described separately. The constraints are classified into nodal constraints and member constraints. These constraints are tabulated in Table 8.9

Table 8.9 Details of Member and Nodal Constraints

Constraint	Necessary condition	Remarks
(a) Member Constraints		
Stress	$\sigma_j / \sigma_{aj} \leq 1$	σ_j = stress in member j σ_{aj} = allowable stress for member j for tension, $\sigma_{aj} = \sigma_a^T$; for compression $\sigma_{aj} = \sigma_a^C$
Slenderness ratio	$s_j / s_{aj} \leq 1$	s_j = slenderness ratio of member j s_{aj} = allowable slenderness ratio of member j for tension, saj = s_a^T;for compression $s_{aj} = s_a^C$
Minimum Length	$L_j / l_{a1} \geq 1$	L_j = length of member j l_{a1} = allowable minimum length for members
Maximum Length	$L_j / l_{a2} \leq 1$	L_j = length of member j l_{a2} = allowable maximum length for members
(b) Nodal Constraints		
Displacement	$d_{jl} / d_{al} \leq 1$	d_{jl} = displacement of node j along coordinate direction l d_{al} = allowable nodal displacement along coordinate direction l
Nodal Symmetry	$x'_{jl} / x''_{jl} = 1$	x'_{jl} , x''_{jl} = coordinates l of symmetric nodal pair j

8.15.5 Penalty and Fitness Functions

Size and Configuration Optimization: If a solution string violates constraint, the constraint function takes non-zero value otherwise it takes zero value. If C is the summation of all such constraint functions for a candidate solution violating constraints, it is penalized using the penalty function given as:

$$P(x) = (1 + K.C) \qquad \ldots (8.93)$$

where K is penalty parameter which is selected judiciously. K can be taken as 10 or it can be varied to take more value in the last generations. In the present study K is kept low in the initial generations and is gradually increased to large values in the subsequent generations using the equation: $K = K_{initial} \{ 1 + 0.2 (n_g - 1) \}$, where n_g is the generation number. A value of 10 has been found suitable for $K_{initial}$. The penalty function value obtained in Eq. (8.93) is multiplied with objective function of Eq. (8.86) to get penalized objective function Op(x).

To avoid negative fitness values following fitness function is used here.

$$f(x) = \frac{1}{1 + O_p(x)} \qquad \text{... (8.94)}$$

***Topology Optimization*:** All penalty functions are expressed in the same general form, as shown below:

$$P_k = \prod_{j=1}^{N_{ck}} (1 + v_{kj} / v_k) \qquad \text{... (8.95)}$$

where, p_k = penalty incurred by a structure for violating constraint k; v_{kj} = a measure of violation of constraint k by component j of the structure (j refers to nodes for nodal constraints and members for member constraints); v_k = a constant corresponding to constraint k, which has the same units as v_{kj}; and N_{ck} = number of components of the structure over which constraint k is applicable. The expression within parenthesis is the part of the penalty function that corresponds to component j of the structure; the penalty function consists of a number of such parts, which are multiplied together to magnify their individual contributions. For those components that do not violate the constraints, $v_{kj} = 0$, which reduce the expression within parenthesis to 1. The constant v_k, is a reference value which is assigned a value greater than the estimated maximum value of v_{kj}. This limits the value of the expression within parenthesis to a range between 1 and 2 in most cases. For all the constraints except the stress constraint the corresponding v_k is kept constant. The penalty function corresponding to various constraints differs from each other only in the definition of the three penalty factors v_{kj}, v_k, and N_{ck}. They are shown in Table 8.10 for penalty functions corresponding to nodal constraints and in Table 8.11 for penalty functions corresponding to member constraints.

All null members subjected to a force exceeding a critical value F_{cr} are assumed to be critical inactive members, whose absence would make the structure unstable. The structures are penalized for such members. The applicability of member constraints to the active, critical inactive and non critical inactive members is shown in Table 8.12. Since structures are synthesized from the active nodes only, they are required to satisfy nodal constraints. The presence of critical inactive members will cause excessive displacements in at least some of the nodes in structure. If particular nodal displacement exceeds a critical value (D_{cr}) it is assumed to be due to the presence of critical inactive members in the structure and is penalized more severely. Thus, structures containing critical inactive members are penalized twice, by means of stress and displacement penalties. A delicate balance needs to be maintained between penalizing the design either very heavily or inadequately. A large penalty is equivalent to restricting the design space and nullifies the positive traits of the GA whereas a small value is not sufficient to push the design into the feasible domain. With the present strategy, it is possible to adjust various penalty factors so as to push the design into the feasible domain by accounting for all the violations and not just the largest violation.

Table 8.10 Penalty Factors for Nodal Constraints

Constraint	Penalty Factors			
	Vkj	Vk	Nck	Condition When Penalty is Applicable
Displacement	Critical Limit on Displacement D_{cr}	Specified Constant D_c	Number of Nodes N_N	$Djl/D_{cr} > 1$
Nodal Symmetry	Positional Error $\delta_{ej} = [\sum(x'jl - x''jl)^2]^{1/2}$	Specified Constant Ec	Number of Nodal Pairs Np_n	$\delta_{ej} > 0$

In Table 8.10, djl = displacement of node j in coordinate direction 1 and x'jl, x"jl = coordinates 1 of symmetric nodal pair j.

Table 8.11 Penalty Factors for Member Constraints

Constraint	Penalty Factors			
	Vkj	Vk	Nck	Condition when Penalty is Applicable
Stress	Deficit Weight $\delta W_j = W_j (\sigma_j/\sigma_{aj}-1)$	Total Weight of Structure $W= \quad W_j$	Number of Members N_M	$\sigma_j / \sigma_{aj} > 1$
Slenderness Ratio	Excess Slenderness Ratio $\delta_{sj} = (S_j - S_{aj})$	Specified Constant S^Tc for Tension, and S^Cc for Compression	Number of Members N_M	$s_j / s_{aj} > 1$
Minimum Length	Deficit Length $\delta l^1_j = (l_{a1} - l_j)$	Specified Constant L_{c1}	Number of Members N_M	$l_j / L_{a1} < 1$
Maximum Length	Excess Length $\delta L^2_j = (l_j - l_{a2})$	Specified Constant L_{c2}	Number of Members N_M	$l_j / l_{a2} > 1$

In Table 8.11, W_j , S_j, l_j = weight, slenderness ratio, and length of member j, σ_j, σ_{aj} = actual and allowable stress in member j, S_{aj} = allowable slenderness ratio for member j and l_{a1}, l_{a2} = allowable minimum and maximum lengths of the members respectively.

Table 8.12 Applicability of Member constraints

Member Constraint	Active Member in Tension or Compression	Inactive Member* in T or C	
		Critical Members	Non-Critical Members
Stress	Yes	Yes	No
Slenderness ratio	Yes	No	No
Minimum length	Yes	Yes	Yes
Maximum length	Yes	No	No

For each structure, the total cost function C^T, is calculated from the objective function and the weighted penalty functions as follows:

$$C^T = C \prod_{k=1}^{N_k} (p_k)^{\alpha k} \qquad \ldots (8.96)$$

where, C = objective function expressed as the cost function that needs to be minimized; p_k = penalty incurred by the structure for violating constraint k; N_k = number of constraints; and αk = exponential penalty weight factor for constraint k. The cost function and the weighted penalty function are multiplied together to magnify their individual contributions. Unlike the objective function that can be exactly evaluated, the penalty functions are highly empirical and, as such, do not have "correct" answers. Contributions from the penalty functions with respect to each other as well as with respect to the objective function need to be manipulated in order to strike a reasonable balance that would lead to evolution of fitter individuals; this is facilitated by the form of the total cost function. The structures that violate constraints are allowed a reasonable chance for propagating their genes into the next generation, depending upon their degree of constraint violations. This helps to preserve genetic diversity in the population. To be acceptable, however the evolved structures must satisfy all design constraints.

8.16 Size Optimization of Plane Truss Example

A 18-bar truss problem considered for the size optimization is shown in Fig. 8.58 along with the support and loading conditions. Panel width = 2.5 m, panel height = 3.0 m and load at all top joints is considered as 45 kN. The program selects discrete sections for various members randomly form the database linked with it and arrives at near optimal set of steel sections giving minimal weight while satisfying all the constraints.

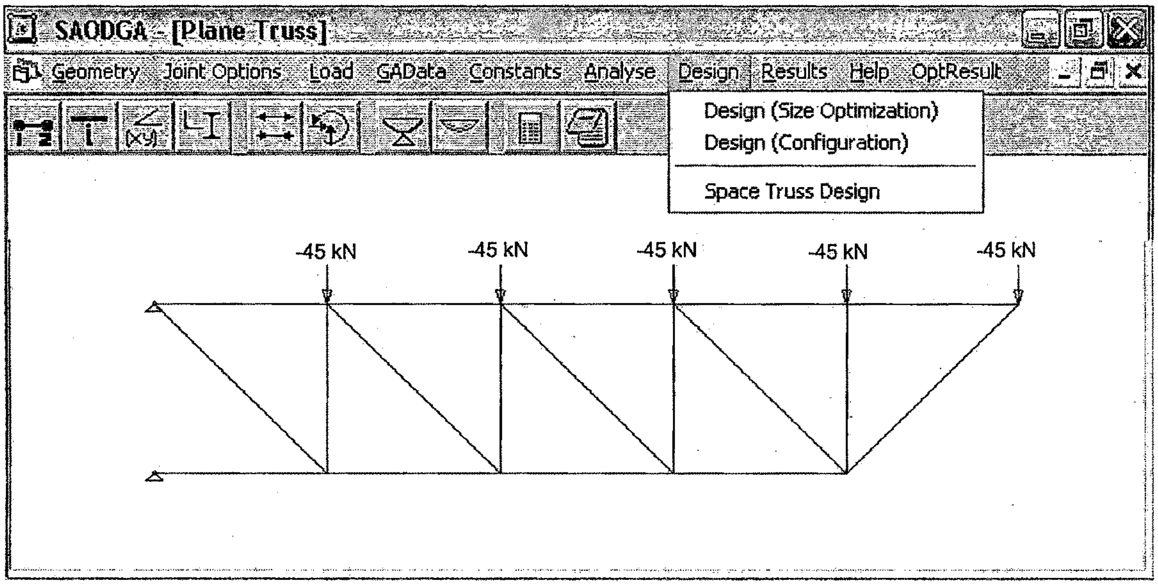


Fig. 8.58 Geometry and Loading of Truss

The size optimization of truss is carried out with population size = 20, number of generations = 30 and crossover and mutation probabilities = 0.67 and 0.03 respectively. Tournament selection scheme is used for this example. Figure 8.58 shows the screenshot of the software indicating the geometry of truss under consideration with support and loading conditions. Figure 8.59 shows member and joint numbers. In the size optimization, member cross sectional area is considered as the design variable and hence number of design variables for this problem is 18. The string length of 8 is considered to represent each variable. Thus total length of solution string turns out to be 144 binary digits. These 18 binary sub strings are decoded to get decimal values which are then mapped to section code specified to each section of the database. Thus design variables are forced to take only discrete values.

The software facilitates the user to input the geometry in three ways. One of these is geometry definition through text file with .sga extension. The above geometry is created using text file written in notepad editor and saved as .sga file. The *Design* menu of the main screen of the software contains sub menus suggesting option of size optimization, configuration optimization and combined optimization. On clicking the *Size Optimization* submenu the program starts finding the set of discrete design variables satisfying all the codal provisions and yet giving minimum weight. The final result in tabular form is depicted in Fig. 8.60. The optimum weight found by the program is 881 kgs.

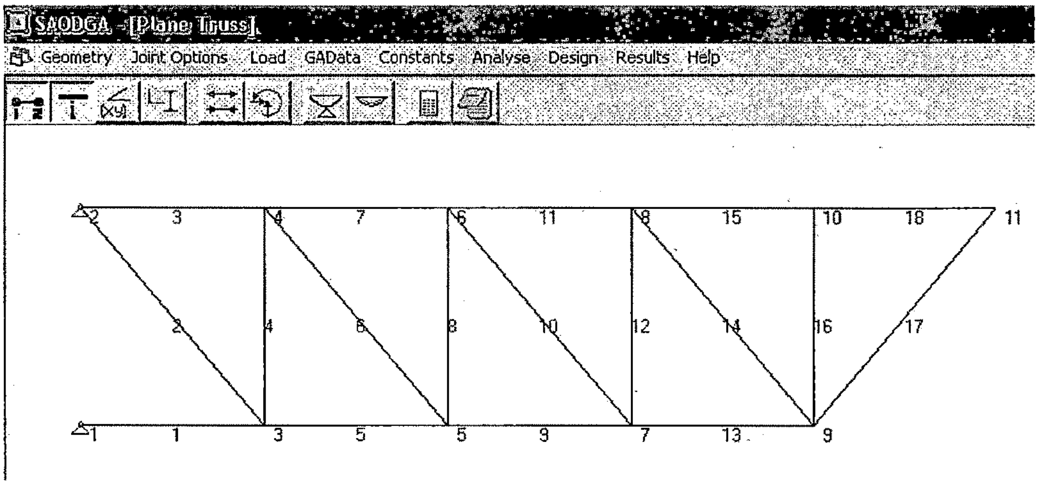
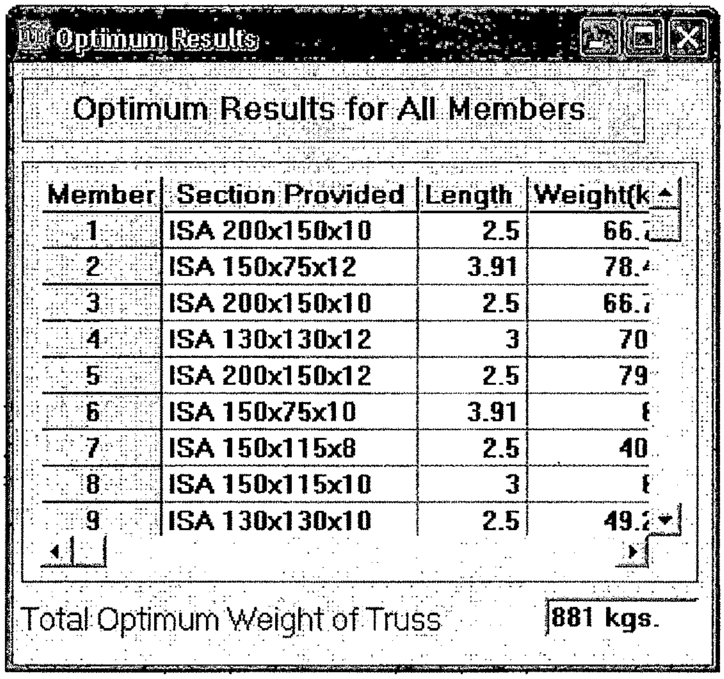


Fig. 8.59 Geometry with Joint and Member Numbering



Optimum Results for All Members

Member	Section Provided	Length	Weight(k
1	ISA 200x150x10	2.5	66.
2	ISA 150x75x12	3.91	78.
3	ISA 200x150x10	2.5	66.
4	ISA 130x130x12	3	70
5	ISA 200x150x12	2.5	79
6	ISA 150x75x10	3.91	[illegible]
7	ISA 150x115x8	2.5	40
8	ISA 150x115x10	3	[illegible]
9	ISA 130x130x10	2.5	49.

Total Optimum Weight of Truss 881 kgs.

Fig. 8.60 Final Results

8.17 CONFIGURATION OPTIMIZATION OF PLANE TRUSS EXAMPLE

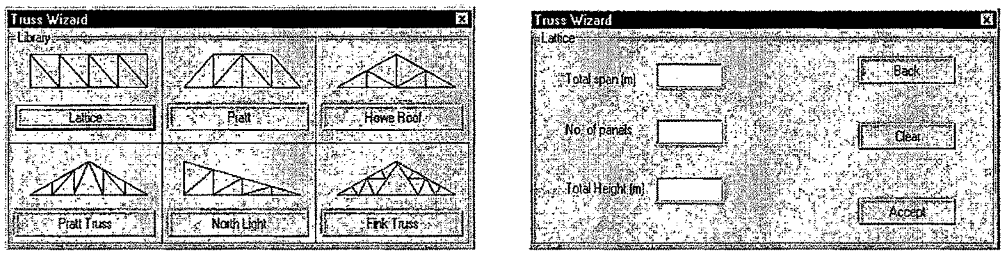
The working of the software is self explanatory with graphical user interface. Following the steps listed below, one can solve any plane truss problem for size and configuration optimization.

❖ The very first step is to run the software from either VB-6 or SAODGA.exe directly. From start menu – click Run, with this, the main *MDI form* will be visible containing *File* menu.

❖ Selecting *New* from file menu, a *New Project* dialog box will be visible showing various options like plane frame, plane truss, space truss, grid etc.

❖ Click on the plane truss option button and enter title of project (optional). On Clicking the OK button, the form for plane truss configuration and size optimization will appear.

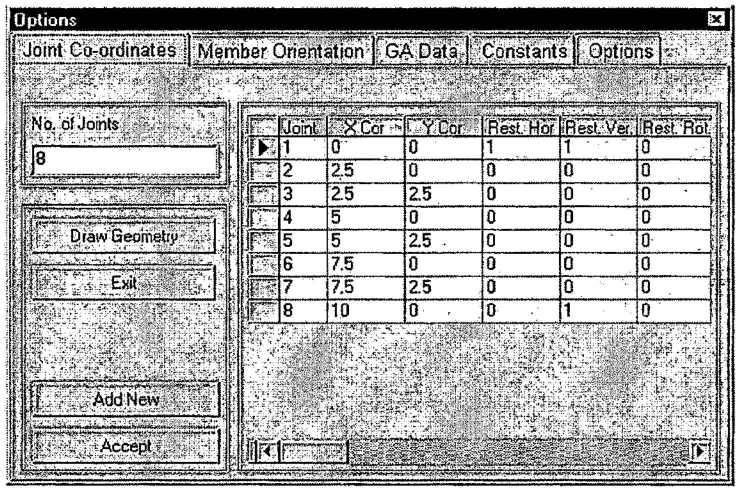
❖ Input of the structural data like joint coordinates and member orientation for initial shape for configuration can be made in two ways i.e. by clicking on standard shapes or by coordinates. For common/standard geometry, click *LIBRARY* submenu of *GEOMETRY* menu which displays a Library menu (Tool bar) as shown in Fig. 8.61(a). Clicking on the appropriate button displays input box for number of panels, total span and total height of the truss as shown in Fig. 8.61(b). Clicking on Accept button draws the geometry.



(a) (b)

Fig. 8.61 Forms for Generating Standard Configuration

For creating geometry other than that provided in standard shapes in Library submenu, the input can be made by coordinates and member orientation, by clicking *COORDINATES* Sub MENU of *Geometry* menu. This displays a form containing *SSTAB* control having tabs like joint information, member information, material properties and options etc. as shown in Fig. 8.62. After tabulating the joint and member, one can click on *Draw Geometry* button to draw the geometry.

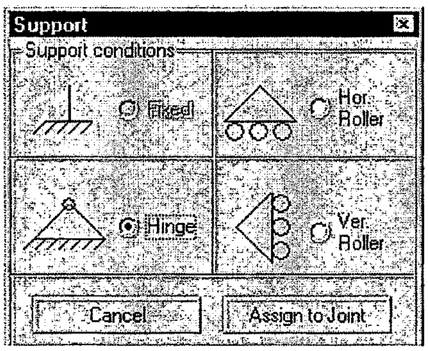


No. of Joints: 8

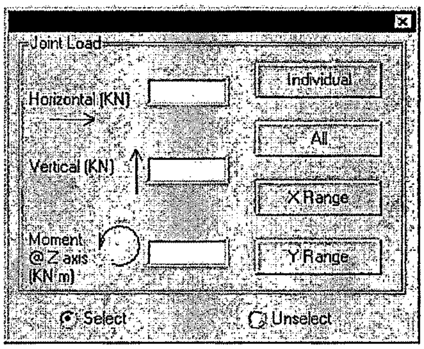
Joint	X Cor	Y Cor	Rest. Hor	Rest. Ver.	Rest. Rot
1	0	0	1	1	0
2	2.5	0	0	0	0
3	2.5	2.5	0	0	0
4	5	0	0	0	0
5	5	2.5	0	0	0
6	7.5	0	0	0	0
7	7.5	2.5	0	0	0
8	10	0	0	1	0

Fig. 8.62 Form for Coordinate Input

- ❖ The next step is to assign support condition. On clicking *SUPPORT* sub menu of *Joint Option,* menu toolbox for assigning support condition is displayed as shown in Fig. 8.63 (a). After Clicking suitable support option and *ASSIGN* button, click on the joint to put that support condition on that joint. Right click of mouse will again display the support menu for changing the support condition.



(a) Support Menu

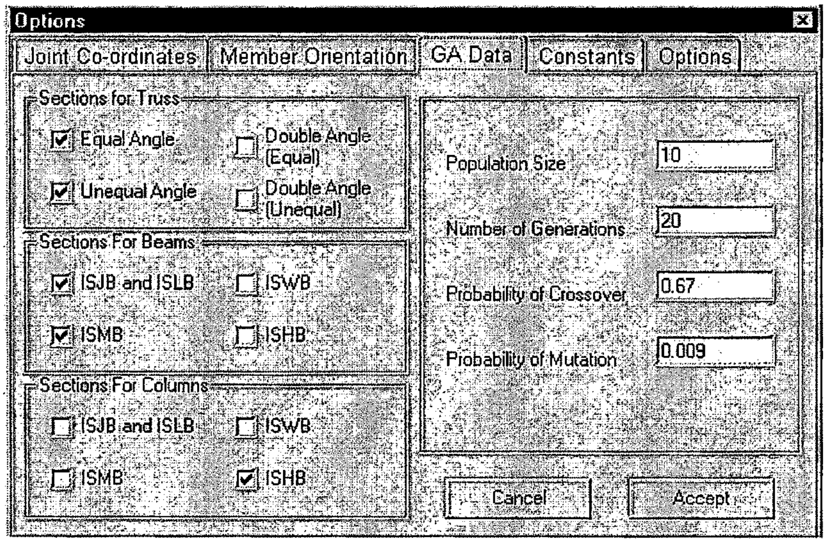


(b) Load Menu

Fig. 8.63 Forms for Assigning Support and Load

Joint loads can be applied by *load* menu as shown in Fig. 8.63 (b) which is available on clicking the *load* menu of main form. The load can be assigned to any joint in a similar way as that of assigning a support condition.

- ❖ The various GA data like population size, number of generations, crossover and mutation probabilities, modulus of elasticity and permissible stresses etc. can be supplied in *OPTION* dialog box as shown in Fig. 8.64 by clicking *GAdata* menu.



Sections for Truss

☑ Equal Angle □ Double Angle (Equal)

☑ Unequal Angle □ Double Angle (Unequal)

Sections For Beams

☑ ISJB and ISLB □ ISWB

☑ ISMB □ ISHB

Sections For Columns

□ ISJB and ISLB □ ISWB

□ ISMB ☑ ISHB

Population Size 10

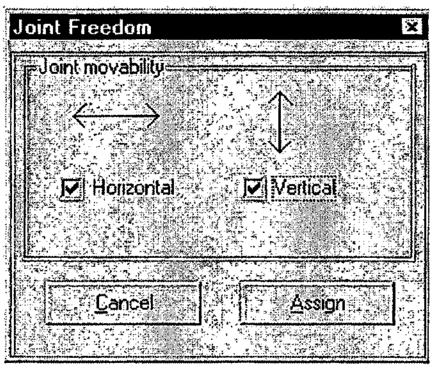
Number of Generations 20

Probability of Crossover 0.67

Probability of Mutation 0.009

Fig. 8.64 Form for GA Data

- ❖ As a GA data, one has to specify the joint as either free joint or fixed joint for configuration. One can restrict a joint to move only in one direction or in both directions by *check box* available in dialog box shown in Fig. 8.65. Both *check box* with their value checked will allow the joint to move in both directions.

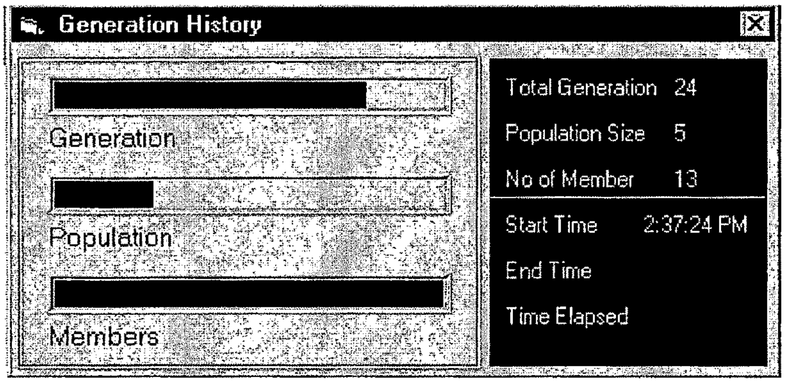


Joint movability

☑ Horizontal ☑ Vertical

Fig. 8.65 Form for Joint Freedom

- ❖ After all the structural and GA data input, click *DESIGN* menu and then *Configuration* sub menu. This action starts the GA operations for configuration optimization. The randomly generated shapes will be drawn on the *picture box* and will be analyzed and designed till the convergence is met or the maximum generation is reached. A form *GENERATION HISTORY* displays the number of Generations, population being processed, the convergence iteration progress bar and start and end time of the run as shown in Fig. 8.66.



Generation History

Generation

Population

Members

Total Generation 24

Population Size 5

No of Member 13

Start Time 2:37:24 PM

End Time

Time Elapsed

Fig. 8.66 Generation History Form

A design and configuration optimization of an 18-bar cantilever truss is carried out using the developed software. Initial geometry and other structural data like support and loading are as shown in Fig. 8.67. The truss has 11 joints. There are two support joints and five loaded joints. Position of 4 remaining joints can be varied during the optimization process. This results in eight geometry variables, viz. x- and y-coordinates of the four joints 3, 5, 7, and 9. The GA parameters considered are: population size = 20, number of generations = 30, and probabilities of crossover and mutation as 0.67 and 0.03 respectively.

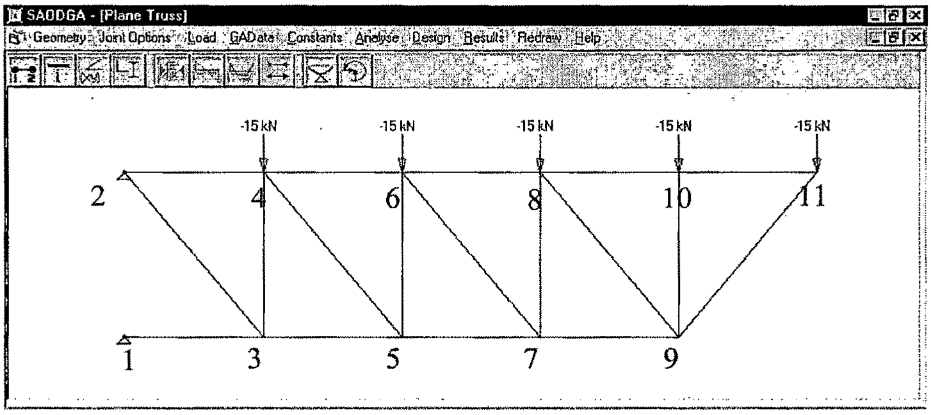
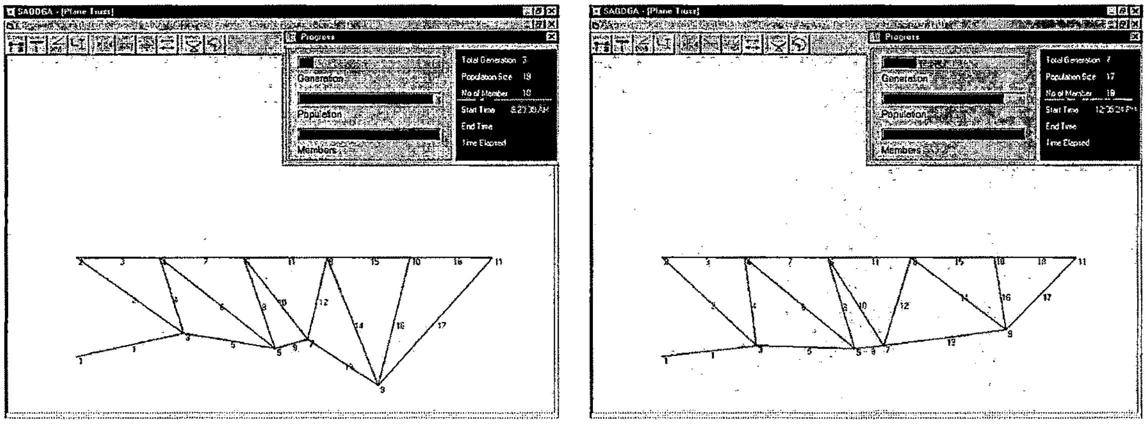


Fig. 8.67 Initial Geometry with Support and Loading Condition

Figures 8.68(a) and 8.68(b) show evolved shapes during search. Figures 8.69 and 8.70 finally display the final configuration and the optimum sections for all the members in graphical and tabular form respectively. It can be seen that the total optimum weight is 353 kg.



(a) At Generation 1

(b) At Generation 15

Fig. 8.68 Intermediate Configurations during Search Process

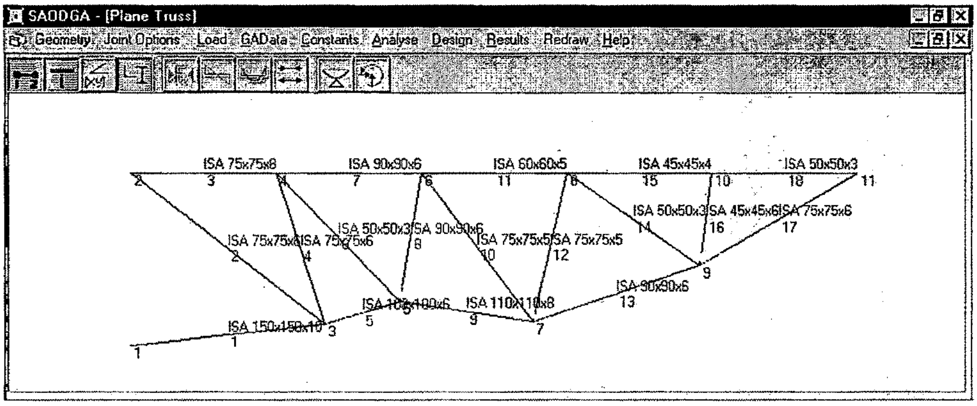
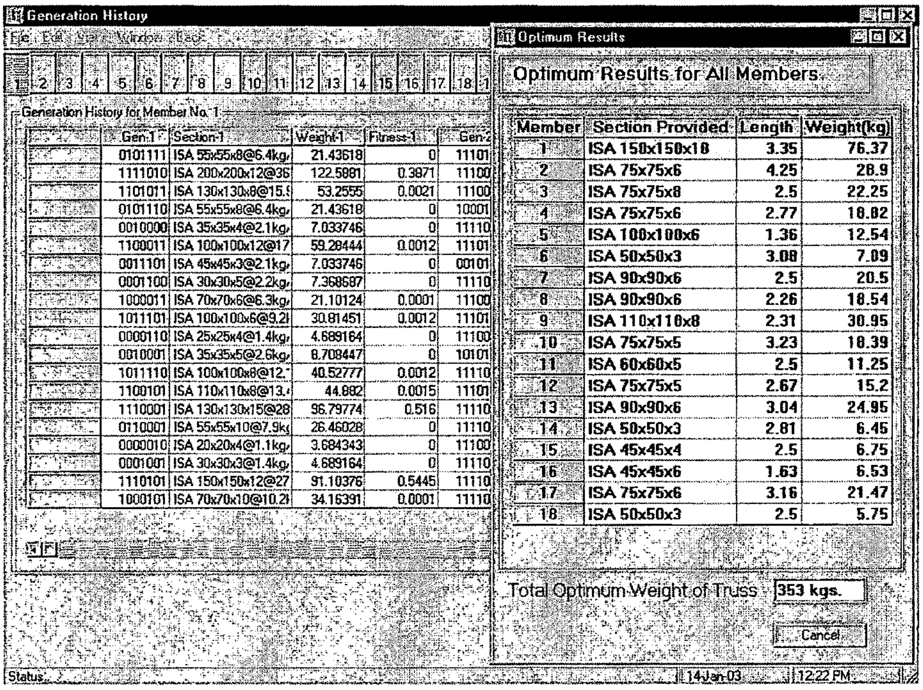


FIG. 8.69 Final Configuration with Selected Sections



Generation History for Member No. 1

Gen-1	Section-1	Weight-1	Fitness-1	Gen-2
0101111	ISA 55x55x8@6.4kg/	21.43618	0	11101
1111010	ISA 200x200x12@36	122.5881	0.3871	11100
1101011	ISA 130x130x8@15.9	53.2555	0.0021	11100
0101110	ISA 55x55x8@6.4kg/	21.43618	0	10001
0010000	ISA 35x35x4@2.1kg/	7.033746	0	11110
1100011	ISA 100x100x12@17	59.28444	0.0012	11101
0011101	ISA 45x45x3@2.1kg/	7.033746	0	00101
0001100	ISA 30x30x5@2.2kg/	7.368687	0	11110
1000011	ISA 70x70x6@6.3kg/	21.10124	0.0001	11100
1011101	ISA 100x100x6@9.2l	30.81451	0.0012	11101
0000110	ISA 25x25x4@1.4kg/	4.689164	0	11100
0010001	ISA 35x35x5@2.6kg/	8.708447	0	10101
1011110	ISA 100x100x8@12.7	40.52777	0.0012	11110
1100101	ISA 110x110x8@13.4	44.882	0.0015	11101
1110001	ISA 130x130x15@28	96.79774	0.516	11110
0110001	ISA 55x55x10@7.9kg	26.46028	0	11110
0000010	ISA 20x20x4@1.1kg/	3.684343	0	11100
0001001	ISA 30x30x3@1.4kg/	4.689164	0	11110
1110101	ISA 150x150x12@27	91.10376	0.5445	11110
1000101	ISA 70x70x10@10.2l	34.16391	0.0001	11110

Optimum Results for All Members

Member	Section Provided	Length	Weight(kg)
1	ISA 150x150x18	3.35	76.37
2	ISA 75x75x6	4.25	28.9
3	ISA 75x75x8	2.5	22.25
4	ISA 75x75x6	2.77	18.82
5	ISA 100x100x6	1.36	12.54
6	ISA 50x50x3	3.08	7.09
7	ISA 90x90x6	2.5	20.5
8	ISA 90x90x6	2.26	18.54
9	ISA 110x110x8	2.31	30.95
10	ISA 75x75x5	3.23	18.39
11	ISA 60x60x5	2.5	11.25
12	ISA 75x75x5	2.67	15.2
13	ISA 90x90x6	3.04	24.95
14	ISA 50x50x3	2.81	6.45
15	ISA 45x45x4	2.5	6.75
16	ISA 45x45x6	1.63	6.53
17	ISA 75x75x6	3.16	21.47
18	ISA 50x50x3	2.5	5.75

Total Optimum Weight of Truss : 353 kgs.

Fig. 8.70 Final Section Sizes for All the Members

Figure 8.71 is the plot of number of generation versus average and best weight of the generation.

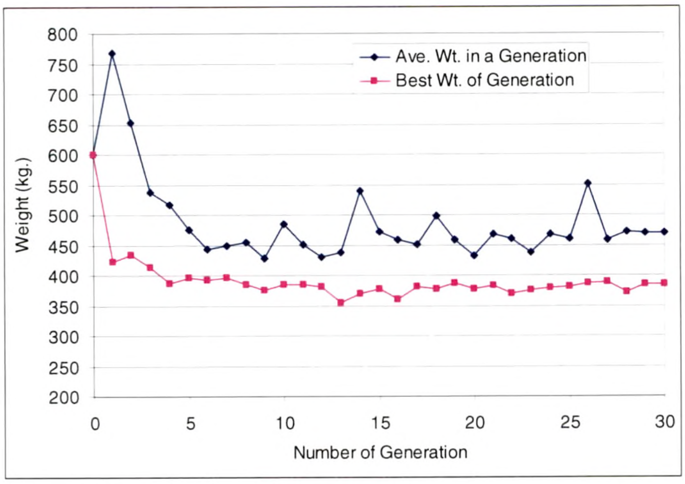


FIG. 8.71 Generation History Showing Average Weight and Best Weight

8.18 TOPOLOGY OPTIMIZATION OF PLANE TRUSS EXAMPLES

8.18.1 Through Configuration Optimization

The configuration optimization algorithm has been used to find the best topology among four predetermined topologies and optimized for optimum configuration. Variation in topology implies variation in number of joints and members defining the truss. Four different topologies are considered in this example to find the optimum topology among the three with optimum configuration. Figures 8.72 (a), (b), (c), (d) show initial geometry and loading on four trusses. Figures 8.73 (a), (b), (c), (d) show the obtained best configuration of the respective initial geometry. The total optimum weight for topology (a), (b), (c) and (d) is 226 kg, 196 kg, 188 kg and 208 kg respectively.

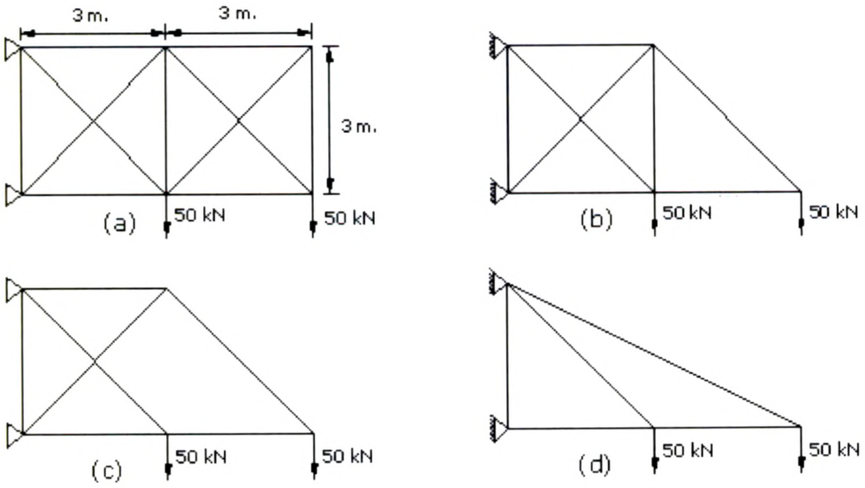


Fig. 8.72 Initial Topologies of Four Trusses

Hence the optimum topology which corresponds to minimum weight is topology(c) with 188 kg as shown in Fig. 8.74.

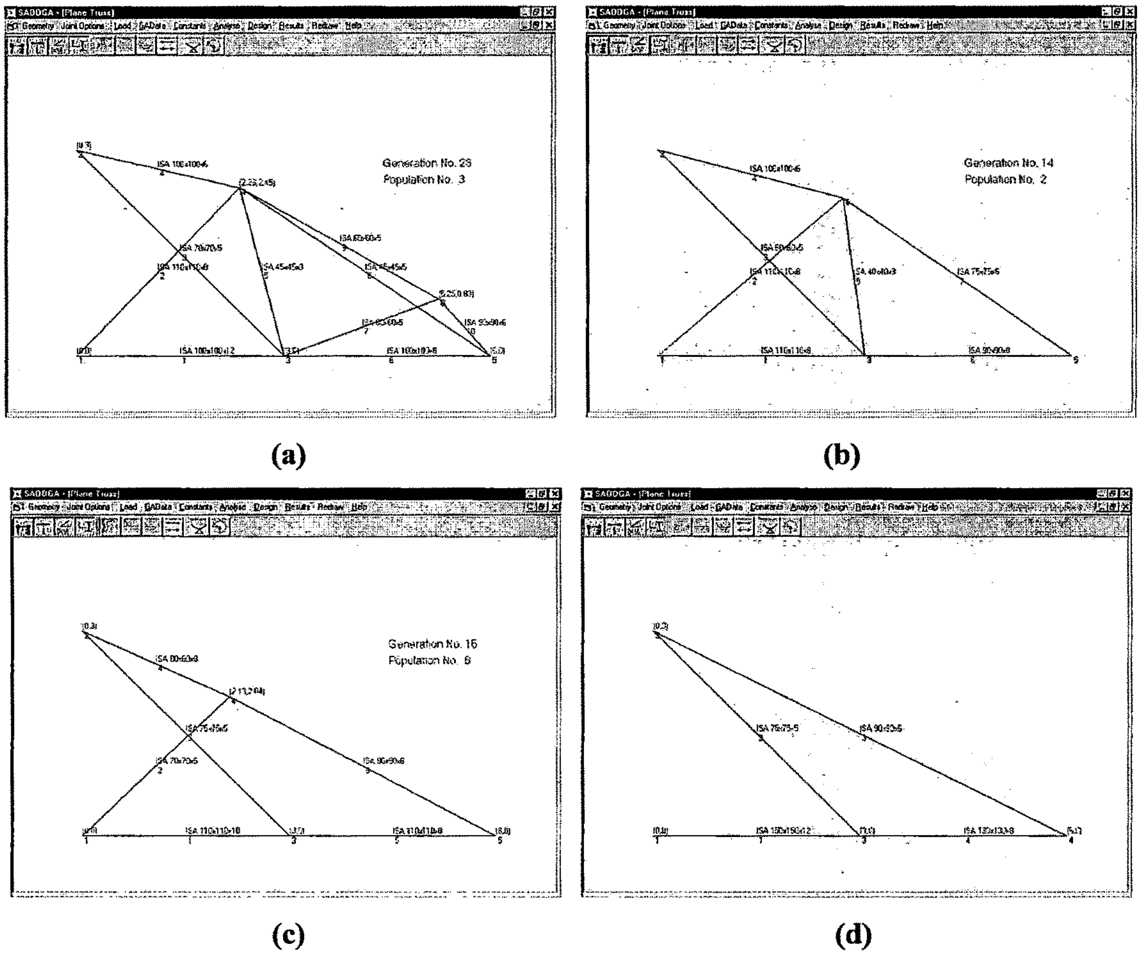
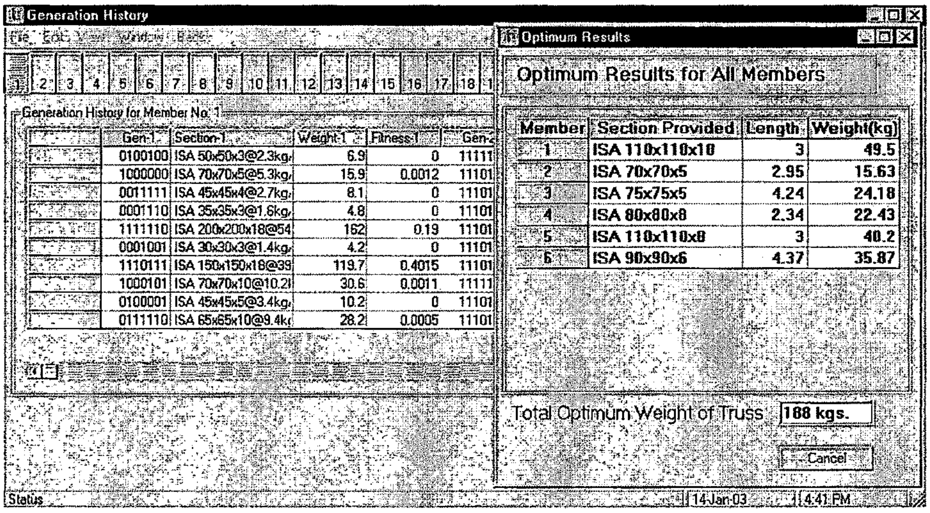


Fig. 8.73 Obtained Optimum Configuration of Trusses



Generation History for Member No. 1

Gen-1	Section-1	Weight-1	Fitness-1	Gen-2
0100100	ISA 50x50x3@2.3kg/	6.9	0	11111
1000000	ISA 70x70x5@5.3kg/	15.9	0.0012	11101
0011111	ISA 45x45x4@2.7kg/	8.1	0	11101
0001110	ISA 35x35x3@1.6kg/	4.8	0	11101
1111110	ISA 200x200x18@54	162	0.19	11101
0001001	ISA 30x30x3@1.4kg/	4.2	0	11101
1110111	ISA 150x150x18@39	119.7	0.4015	11101
1000101	ISA 70x70x10@10.2l	30.6	0.0011	11111
0100001	ISA 45x45x5@3.4kg/	10.2	0	11101
0111110	ISA 65x65x10@9.4k(	28.2	0.0005	11101

Optimum Results for All Members

Member	Section Provided	Length	Weight(kg)
1	ISA 110x110x10	3	49.5
2	ISA 70x70x5	2.95	15.63
3	ISA 75x75x5	4.24	24.18
4	ISA 80x80x8	2.34	22.43
5	ISA 110x110x8	3	40.2
6	ISA 90x90x6	4.37	35.87

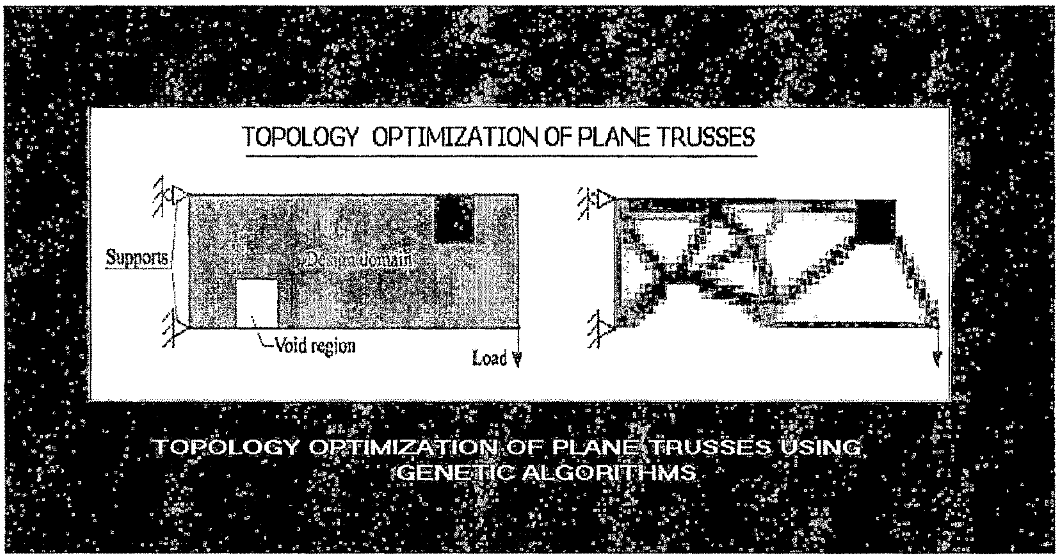
Total Optimum Weight of Truss: 188 kgs.

Fig. 8.74 Optimum Weight for Topology (c)

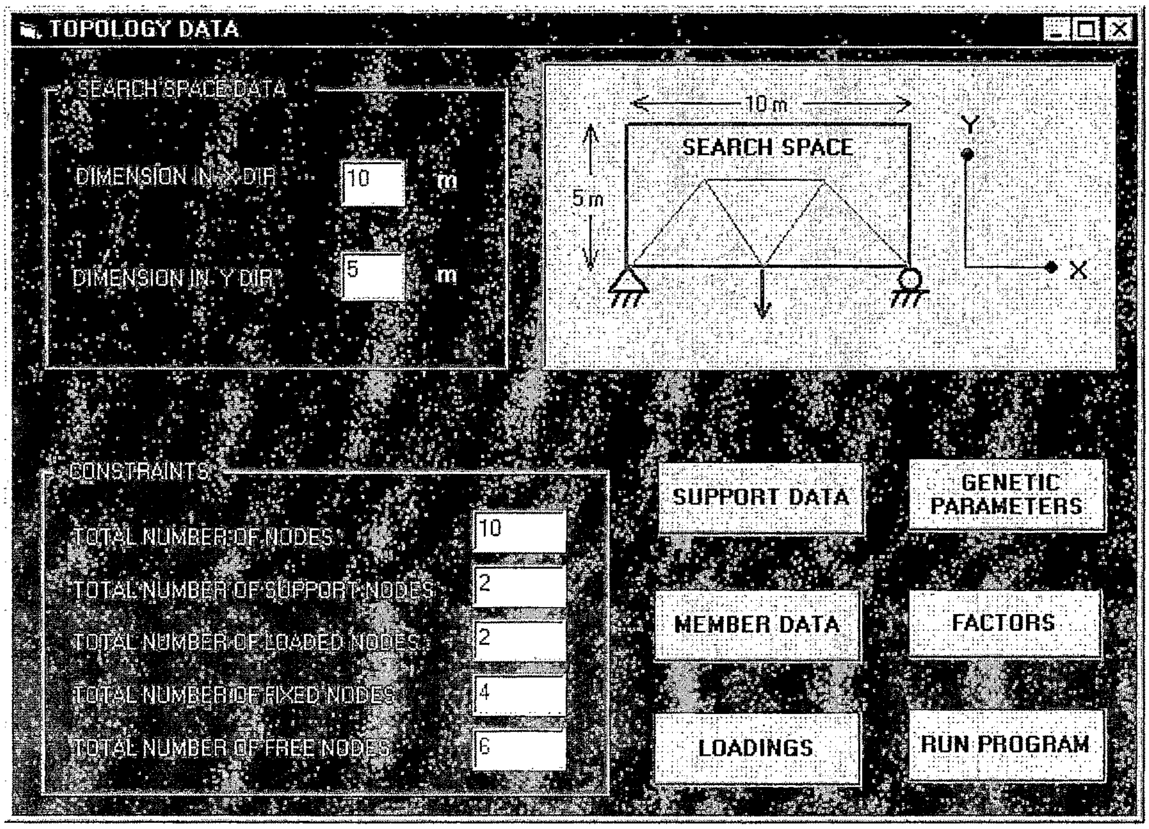
8.18.2 Direct Topology Optimization of Plane Truss Example

8.18.2.1 Steps to use the software and screenshots

1). **Startup screen of the software**


TOPOLOGY OPTIMIZATION OF PLANE TRUSSES
Supports
Design domain
Void region
Load
TOPOLOGY OPTIMIZATION OF PLANE TRUSSES USING GENETIC ALGORITHMS

2). **Input the data**


TOPOLOGY DATA

SEARCH SPACE DATA

DIMENSION IN X DIR 10 m

DIMENSION IN Y DIR 5 m

CONSTRAINTS

TOTAL NUMBER OF NODES 10

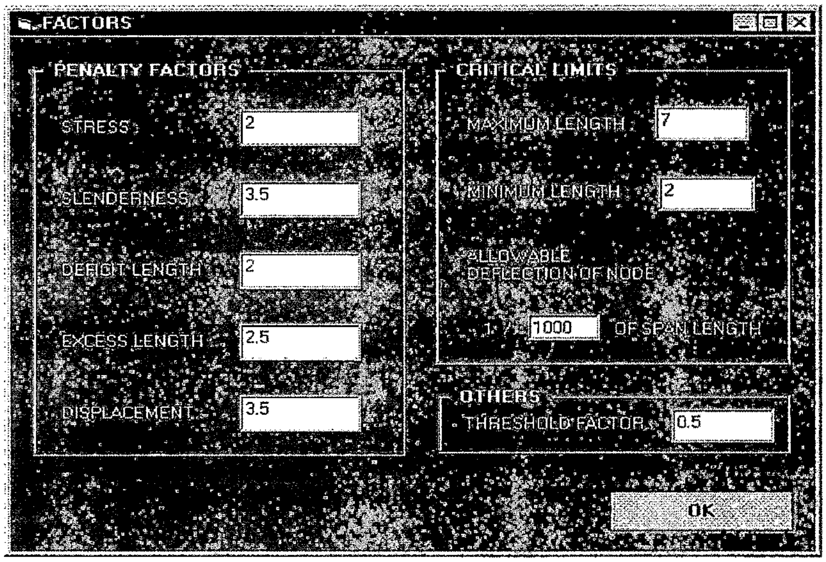
TOTAL NUMBER OF SUPPORT NODES 2

TOTAL NUMBER OF LOADED NODES 2

TOTAL NUMBER OF FIXED NODES 4

TOTAL NUMBER OF FREE NODES 6

3). Input the parameter settings



FACTORS

PENALTY FACTORS

STRESS : 2

SLENDERNESS : 3.5

DEFICIT LENGTH : 2

EXCESS LENGTH : 2.5

DISPLACEMENT : 3.5

CRITICAL LIMITS

MAXIMUM LENGTH : 7

MINIMUM LENGTH : 2

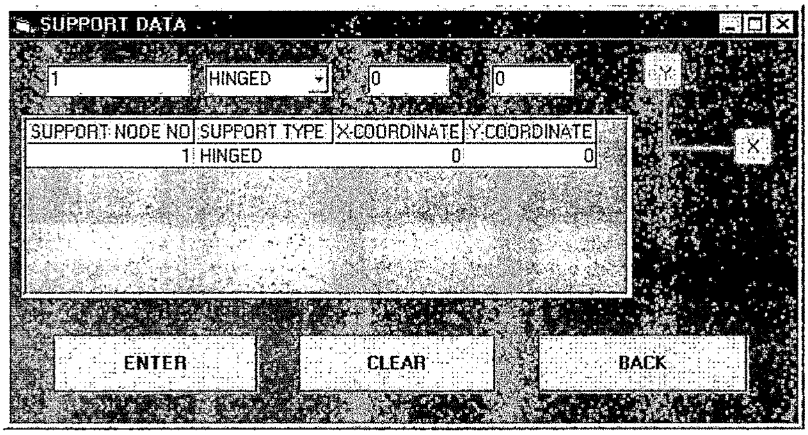
ALLOWABLE DEFLECTION OF NODE

1 / 1000 OF SPAN LENGTH

OTHERS

THRESHOLD FACTOR : 0.5

4). Input the support data

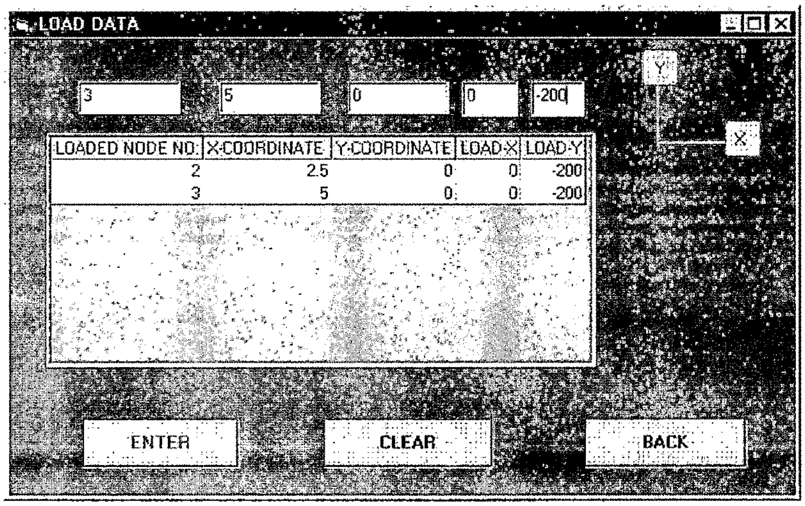


SUPPORT DATA

1 HINGED 0 0

SUPPORT NODE NO	SUPPORT TYPE	X-COORDINATE	Y-COORDINATE
1	HINGED	0	0

5). Input the loadings

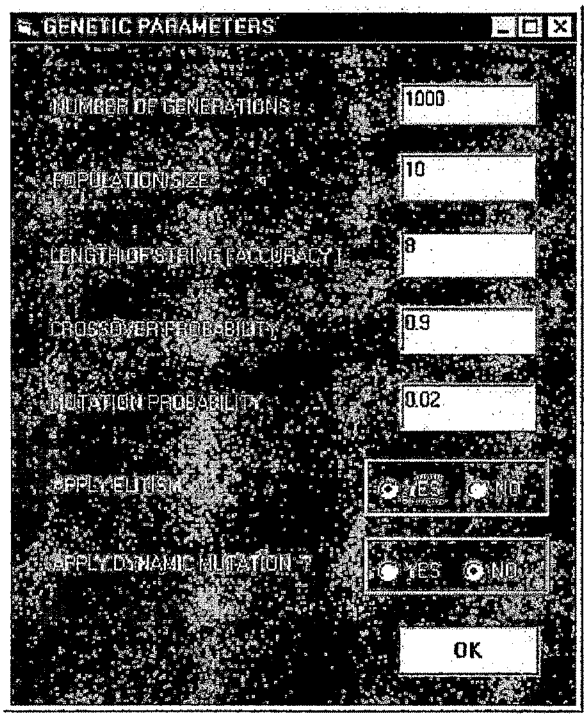


LOAD DATA

3 5 0 0 -200

LOADED NODE NO	X-COORDINATE	Y-COORDINATE	LOAD-X	LOAD-Y
2	2.5	0	0	-200
3	5	0	0	-200

6). Input the genetic parameters



GENETIC PARAMETERS

NUMBER OF GENERATIONS : 1000

POPULATION SIZE : 10

LENGTH OF STRING [ACCURACY] : 8

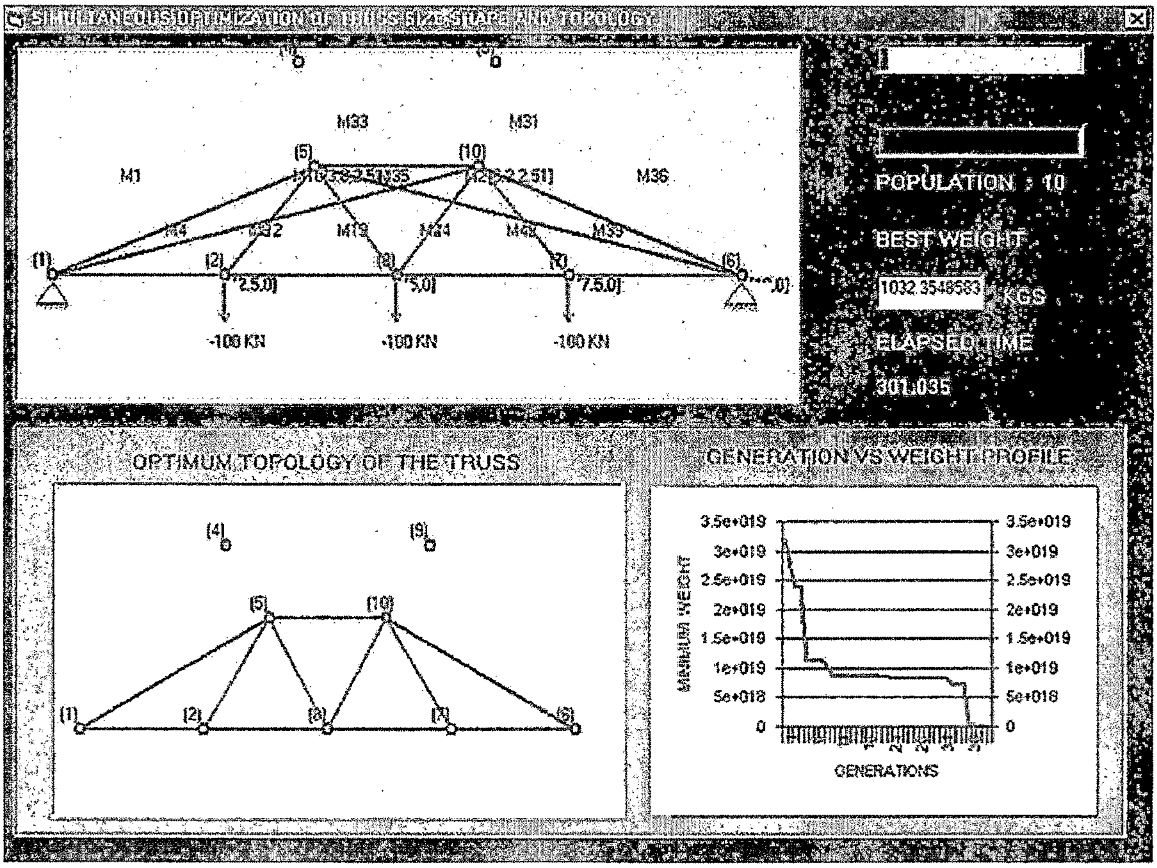
CROSSOVER PROBABILITY : 0.9

MUTATION PROBABILITY : 0.02

APPLY ELITISM : ● YES ○ NO

APPLY DYNAMIC MUTATION ? ○ YES ● NO

7). Run the program and view the evolution of optimum shape of truss



SIMULTANEOUS OPTIMIZATION OF TRUSS SIZE,SHAPE AND TOPOLOGY

POPULATION : 10

BEST WEIGHT 1032.3548583 KGS

ELAPSED TIME 301.035

8.18.2.2 Example of direct topology optimization

Obtain optimum shape of a truss enclosed in a search space of 10 m x 5m. The maximum number of free joints is 10. The truss is loaded with three point loads of 100 kN each, acting at quarter points of the bottom panel of the truss. The truss has hinged supports at both the ends (Fig. 8.75).

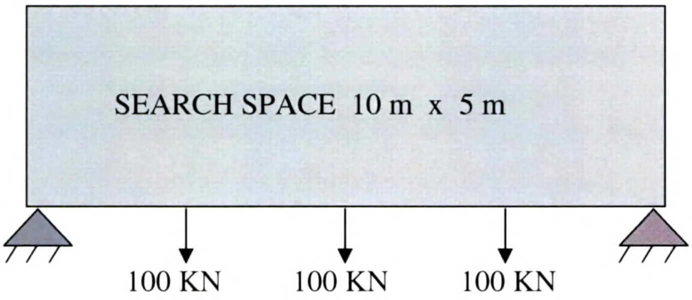
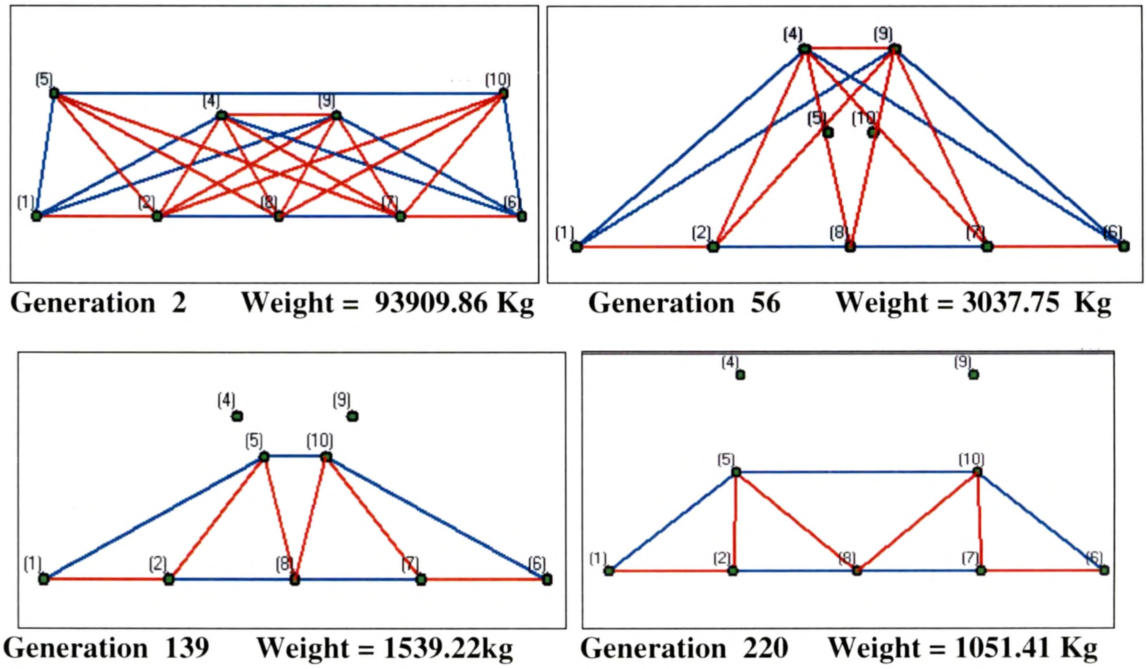


Fig. 8.75 Search Space for Optimization

The following parameter settings have been used to solve the problem: Maximum allowable length of member = 7 m, Minimum allowable length of member = 2 m, Allowable limit of displacement = 1/1000 of span, Number of sector discretizations = 16, Threshold factor = 0.5, Critical force Fcr = 2 kN, Penalty factor for slenderness = 3.5, Penalty factor for stress = 2, Penalty factor for minimum length = 2, Penalty factor for maximum length = 2.5, Penalty factor for nodal displacement = 2, Total generations = 1000, Population size = 10, Probability of mutation = 0.0333, Probability of crossover = 0.6, String length = 8 bits and Elitism = true.

Progressive Evolution of Shape and Reduction in Total Weight of The Truss

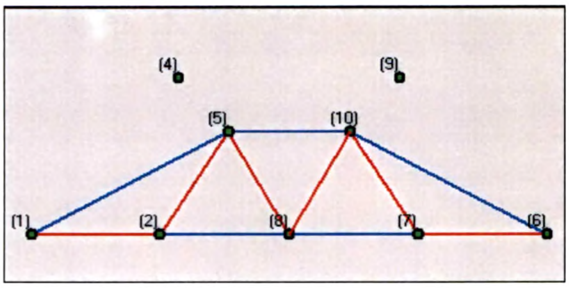


Generation 2 Weight = 93909.86 Kg

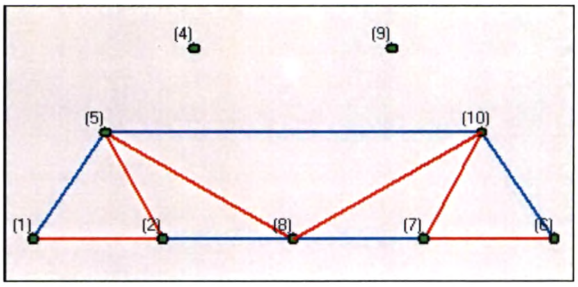
Generation 56 Weight = 3037.75 Kg

Generation 139 Weight = 1539.22kg

Generation 220 Weight = 1051.41 Kg



Generation 474 Weight = 1031.31 Kg



Generation 769 Weight = 1019.41 Kg

Final result at 1000th generation gives best weight of the truss = 940.06 kg. The Final topology is as shown in Fig. 8.76. Figure 8.77 shows deflected shape of the final topology. In the above figures, red colour represents tension, blue colour represents compression and green colour is for zero force members.

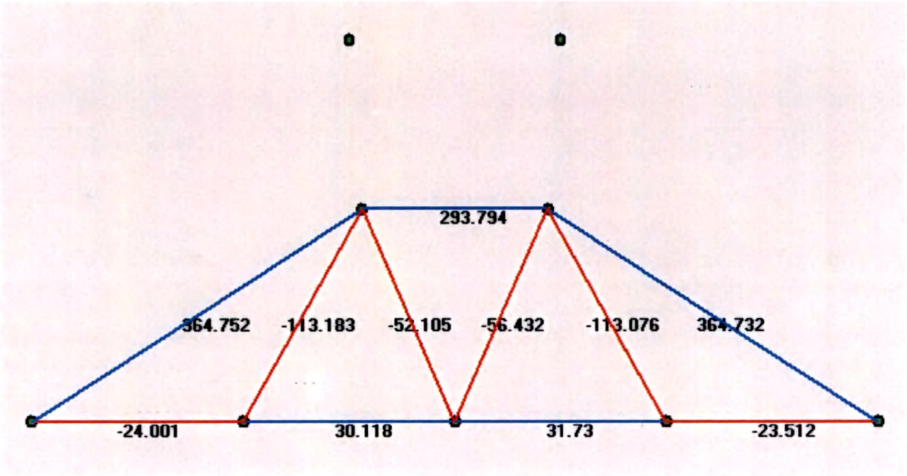


Fig. 8.76 Force Distribution in the Final Geometry of the Truss

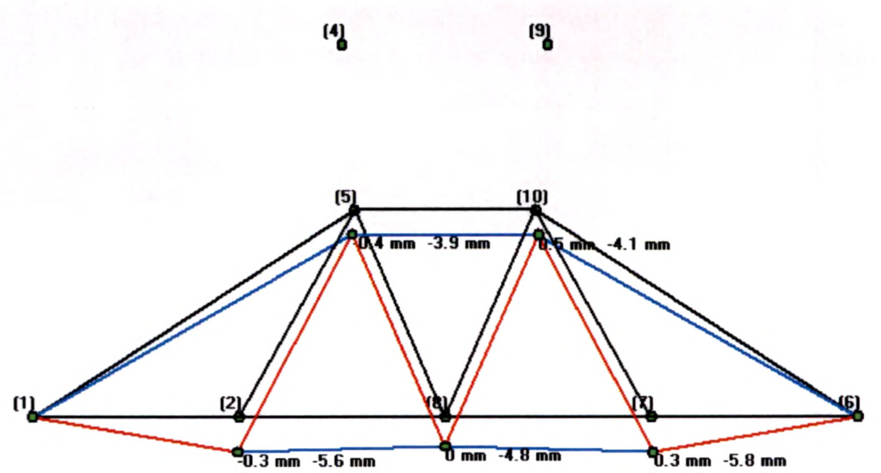


Fig. 8.77 Deflected Shape of the Final Truss Geometry

PRINTABLE OUTPUT GENERATED BY THE SOFTWARE

```
TOPOLOGY OPTIMIZATION OF PLANE TRUSSES USING GENETIC ALGORITHM

 # TOPOLOGY DATA #

 [1] PHYSICAL SEARCH SPACE : 10  m  X  5  m
 [2] MAXIMUM NUMBER OF NODES : 10
 [3] TOTAL NUMBER OF FIXED NODES : 3
 [4] NUMBER OF SUPPORT NODES : 2
 [5] NUMBER OF LOADED NODES: 2
 [6] THRESHOLD FACTOR : 0.5
 [7] LIMITING PARAMETERS :
      MAXIMUM LENGTH : 7 m
      MINIMUM LENGTH : 2 m
      ALLOWABLE DEFLECTION : 1 / 1000  OF SPAN

 [8] PENALTY FACTORS :
      DEFICIT LENGTH : 2
      EXCESS LENGTH : 2.5
      SLENDERNESS RATIO : 3.5
      STRESS : 2
      DISPLACEMENT : 3.5

 # GENETIC DATA #

  [1] TOTAL GENERATIONS : 1000
  [2] POPULATION SIZE : 10
  [3] PROBABILITY OF CROSSOVER : 0.6
  [4] PROBABILITY OF MUTATION : 0.033
  [5] LENGTH OF STRING [ACCURACY] : 8

  RESULTS :

   FITNESS OF THE OPTIMUM TRUSS = 9.93977E-04
   TOTAL WEIGHT OF THE OPTIMUM TRUSS = 940.059  Kgs

    ****************************************
     MEMBER RESULTS FOR THE GENERATED TRUSS (Sample Output)
    ****************************************
    -----------------------------------------------------------------------------------------------
    |MEMBER-NO |J-END| X  | Y  |KEND| X    | Y |MEMBER-STAT|FORCE     |SECTION-TYPE
    -----------------------------------------------------------------------------------------------
    | 1          | 1    | 0  | 0  | 2          | 2.5 | 0   |True       |-24.001   |ISA 70x70x8 @ 8.3 KG/M
    | 2          | 1    | 0  | 0  | 3          | 5   | 0   |False      |-0.039    |ISA 70x70x8 @ 8.3 KG/M
    | 3          | 1    | 0  | 0  | 4          | 3.1 | 4.4 |False      | 1.152    |non-active @ 0.1 KG/M
    | 4          | 1    | 0  | 0  | 5          | 3.9 | 2.5 |True       | 364.752  |ISA 200x200x12 @ 36.6 KG/M
    | 5          | 1    | 0  | 0  | 6          | 10  | 0   |False      | 0        |non-active @ 0.1 KG/M
    | 6          | 1    | 0  | 0  | 7          | 7.5 | 0   |False      | 0.074    |ISA 70x70x8 @ 8.3 KG/M
```

8.19 Optimization of Space Trusses in General

Structural design of transmission towers is an area most suited for computer application to optimization. Since the towers of a particulars type have to be fabricated and erected in large numbers, a big saving in the overall cost of a project justifies the computational time and effort utilized for minimizing the weight of a particular tower.

The algorithm developed for optimization of space truss, is flexible and does not suffer the draw back of using only feasible initial solutions. It operates equally from both feasible and infeasible starting points. The size optimization of any pin jointed structure is a discrete optimization problem, where variable represents the area of the corresponding member and assumes value from a list of standard sections. Whereas in configuration optimization the design variables, which are joint coordinates, are continuous.

8.19.1 Size and Configuration Optimization

The space truss is assumed to be loaded by nodal point loads only hence the members will be subjected to axial compressive or tensile force without any bending induced in them. In the size and optimization member cross sectional areas are considered as design variables. In pure configuration optimization x, y and z coordinates of some of the movable joints are taken as design variables. In the transmission towers, for the joints representing supports, only x and y coordinates are varied while z coordinates are kept constant. Moreover, nodes supporting the conductors are also not allowed to vary. Due to symmetrical shape of the truss design variables can be further reduced.

❖ **Constraints**

1. Slenderness ratio $\lambda_a - \lambda_{per} \leq 0$ ... (8.97)
2. Stress in members $\sigma_a - \sigma_{per} \leq 0$ ... (8.98)
3. Displacement at the node $\delta_a - \delta_{per} \leq 0$ ... (8.99)

where λ_{per} and λ_a are permissible and actual slenderness ratios of the member, σ_{per} and σ_a are permissible and actual stresses in the truss member and δ_{per} and δ_a permissible and actual nodal displacements.

❖ **Objective Function**

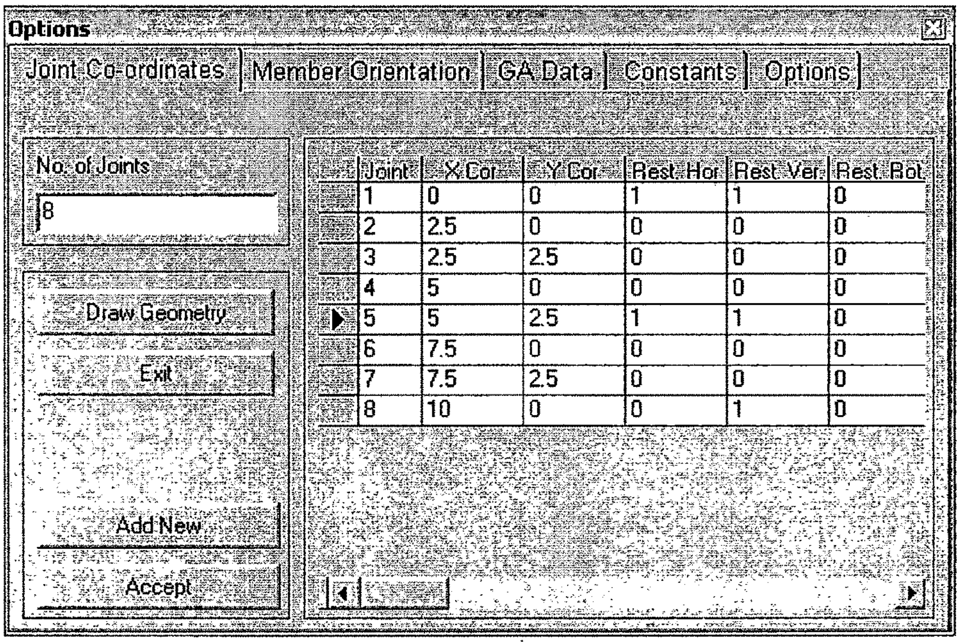
The weight of the truss is considered as objective function and to tackle the problem of negative fitness the fitness function used for the plane truss is adopted here.

8.19.2 Analysis and Design

The stiffness matrix method [92] is used for analysis. For every new configuration generated, the analysis routine is called and the force distribution in each member is tabulated. The member forces are then used for the design of individual members based on the recommendations of IS: 800 [91]. A suitable section is selected based on the calculated requirements of area and radius of gyration. The section is then checked for slenderness and permissible stress constraints. If found insufficient, next higher section is selected till all the requirements are met with. Thus here while finding optimum configuration the member sizes are obtained by normal design method to reduce number of variables and to speed up the search process. However, software provides the facility for simultaneous size and configuration optimization.

8.19.3 Step by Step Procedure and Developed Processors

The developed software is not only capable of giving minimum weight design of any space truss configuration but also capable of optimizing configuration for a given topology. One can adopt the symmetry concept, if applicable, to the problem which may result in a large saving in computational time. Various forms developed under pre processor for various input required for optimization using GA are described one by one. Fig. 8.78 shows the form which enables user to define the initial truss geometry by supplying number of joints, x, y and z coordinates of joints, support conditions and loading conditions.

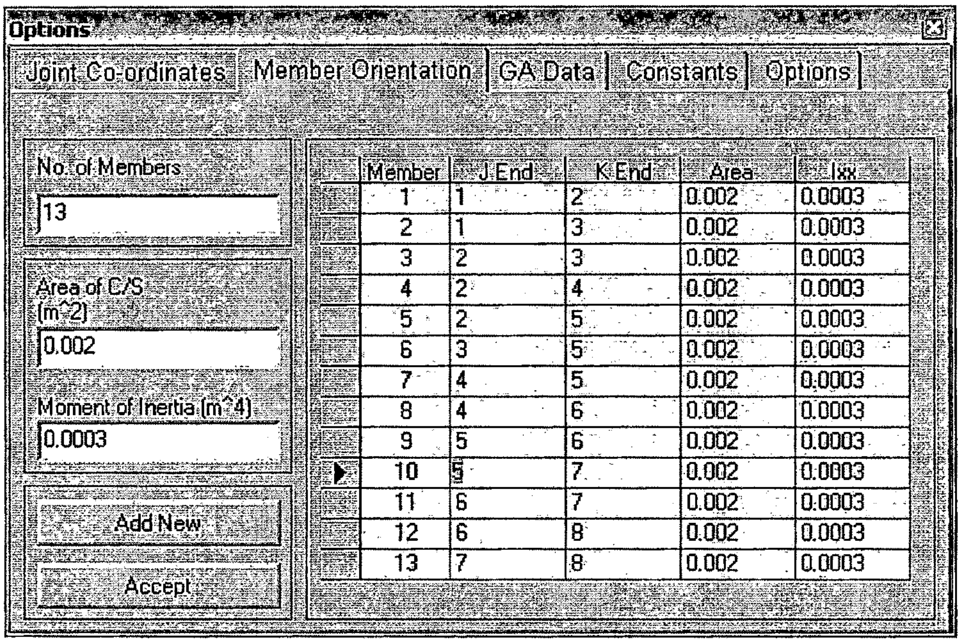


No. of Joints: 8

Joint	X Cor	Y Cor	Rest Hor	Rest Ver	Rest Rot
1	0	0	1	1	0
2	2.5	0	0	0	0
3	2.5	2.5	0	0	0
4	5	0	0	0	0
5	5	2.5	1	1	0
6	7.5	0	0	0	0
7	7.5	2.5	0	0	0
8	10	0	0	1	0

Fig. 8.78 Form for Joint Information for a Space Truss

The information regarding members such as jth end kth end, cross sectional area, etc. is to be entered in the data grid provided in the form shown in Fig. 8.79. The data related to GA such as population size, number of generations, cross over and mutation probabilities, type of sections to be used etc. are to be supplied using form shown in Fig. 8.80.



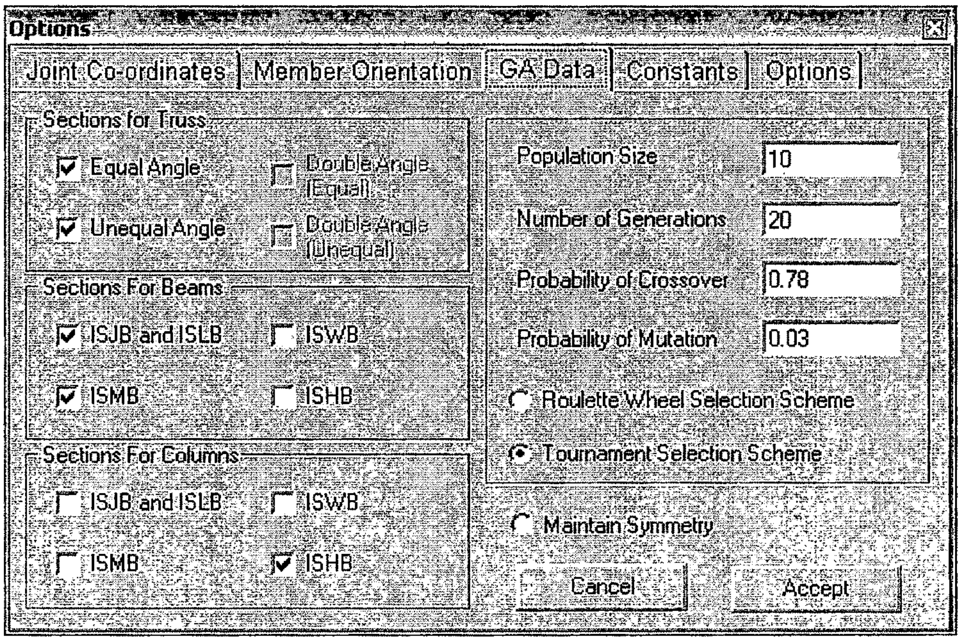
No. of Members: 13

Area of C/S (m^2): 0.002

Moment of Inertia (m^4): 0.0003

Member	J End	K End	Area	Ixx
1	1	2	0.002	0.0003
2	1	3	0.002	0.0003
3	2	3	0.002	0.0003
4	2	4	0.002	0.0003
5	2	5	0.002	0.0003
6	3	5	0.002	0.0003
7	4	5	0.002	0.0003
8	4	6	0.002	0.0003
9	5	6	0.002	0.0003
10	5	7	0.002	0.0003
11	6	7	0.002	0.0003
12	6	8	0.002	0.0003
13	7	8	0.002	0.0003

Fig 8.79 Form for Member Information for Space Truss



Sections for Truss

☑ Equal Angle ☐ Double Angle (Equal)

☑ Unequal Angle ☐ Double Angle (Unequal)

Sections For Beams

☑ ISJB and ISLB ☐ ISWB

☑ ISMB ☐ ISHB

Sections For Columns

☐ ISJB and ISLB ☐ ISWB

☐ ISMB ☑ ISHB

Population Size: 10

Number of Generations: 20

Probability of Crossover: 0.78

Probability of Mutation: 0.03

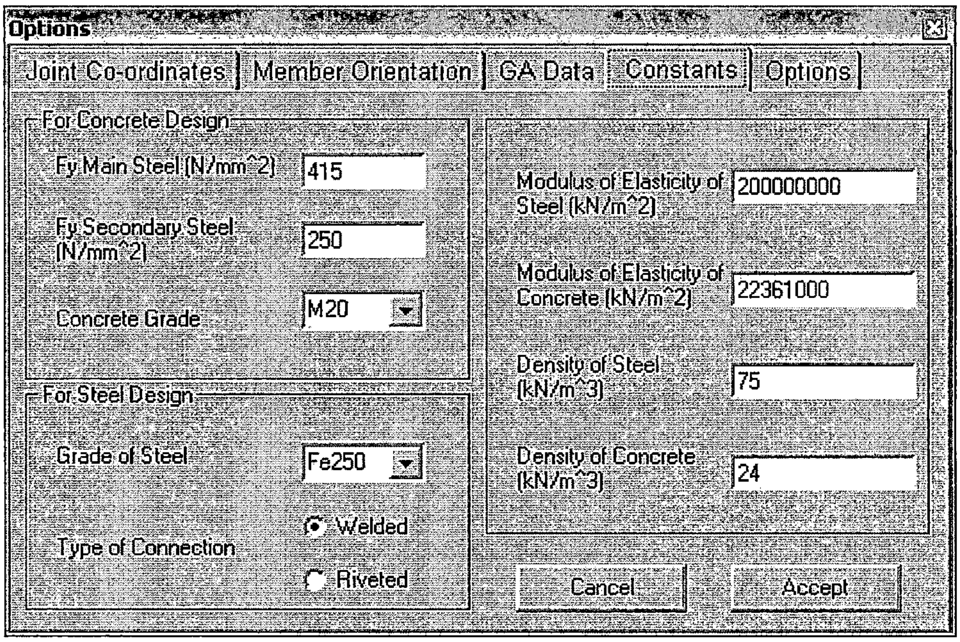
○ Roulette Wheel Selection Scheme

● Tournament Selection Scheme

○ Maintain Symmetry

Fig. 8.80 Form for Entering GA Data for Space Truss

The data such as permissible stresses, density of material, modulus of elasticity etc. is to be entered through the form depicted in Fig. 8.81.



For Concrete Design

Fy Main Steel (N/mm^2): 415

Fy Secondary Steel (N/mm^2): 250

Concrete Grade: M20

For Steel Design

Grade of Steel: Fe250

Type of Connection: ● Welded ○ Riveted

Modulus of Elasticity of Steel (kN/m^2): 200000000

Modulus of Elasticity of Concrete (kN/m^2): 22361000

Density of Steel (kN/m^3): 75

Density of Concrete (kN/m^3): 24

Fig. 8.81 Form for Entering Material Constants for Space Truss

After supplying all the data as specified above and clicking the design button the execution of program starts. The roll of the main processor starts here which calculates the member forces and joint displacements for evaluation of constraint functions and penalty. It also evaluates the objective function and fitness of the solution strings. Finally it activates the GA module to find the optimum solution.

The post-processor provides the results in tabular and graphical form. It includes the tabulated generation history showing selected sections for each solution string and fitness of the string. The final optimized solution is shown in the separate form in table format.

8.20 Configuration Optimization of Space Truss Example

A 25-bar transmission tower space truss of Fig. 8.82 with an initial geometry and topology that have been investigated in combination of geometry and sizing optimization using GA approach by Hansen and Vanderplaats [93] and Soh and Yang [94]. The same truss is considered here for the sake of comparison. The load case considered is same as that used by Soh and Yang [94] and is given below.

Loading (KN)	Force at joints	
	1	2
F_x	0.0	0.0
F_y	88.9	-88.9
F_z	-22.225	-22.225

The truss is also required to remain symmetric with respect to both x-z plane and y-z plane. Thus, only 5 coordinates i.e. x_4, y_4 and z_4 of joint 4 and x_8 and y_8 of joint 8 are considered as geometric variables [94]. Then coordinates of joints 3, 5 and 6 are assigned the coordinates of joint 4 and coordinates of joints 7, 9 and 10 are assigned the coordinates of joint 8. Various genetic parameters used for the optimization are: population size = 20 number of generations = 50. Single point crossover and bit flip mutation are used. Elitism operator is also employed here.

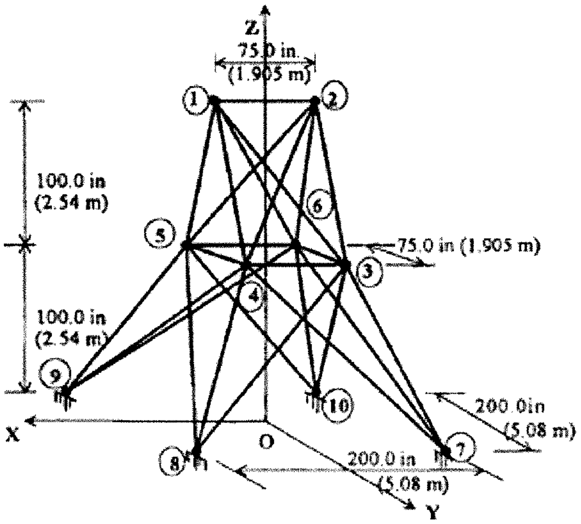


Fig. 8.82 Geometry of 25 bar Space Truss

Flowchart of configuration optimization is depicted in Fig. 8.83. Figure 8.84(a) shows the final results in terms of coordinates of movable joints and selected sections for all the members and Fig. 8.84(b) shows final configuration. Fig. 8.85 shows improvement in the objective function and Fig. 8.86 indicates improvement in the best fitness with generations.

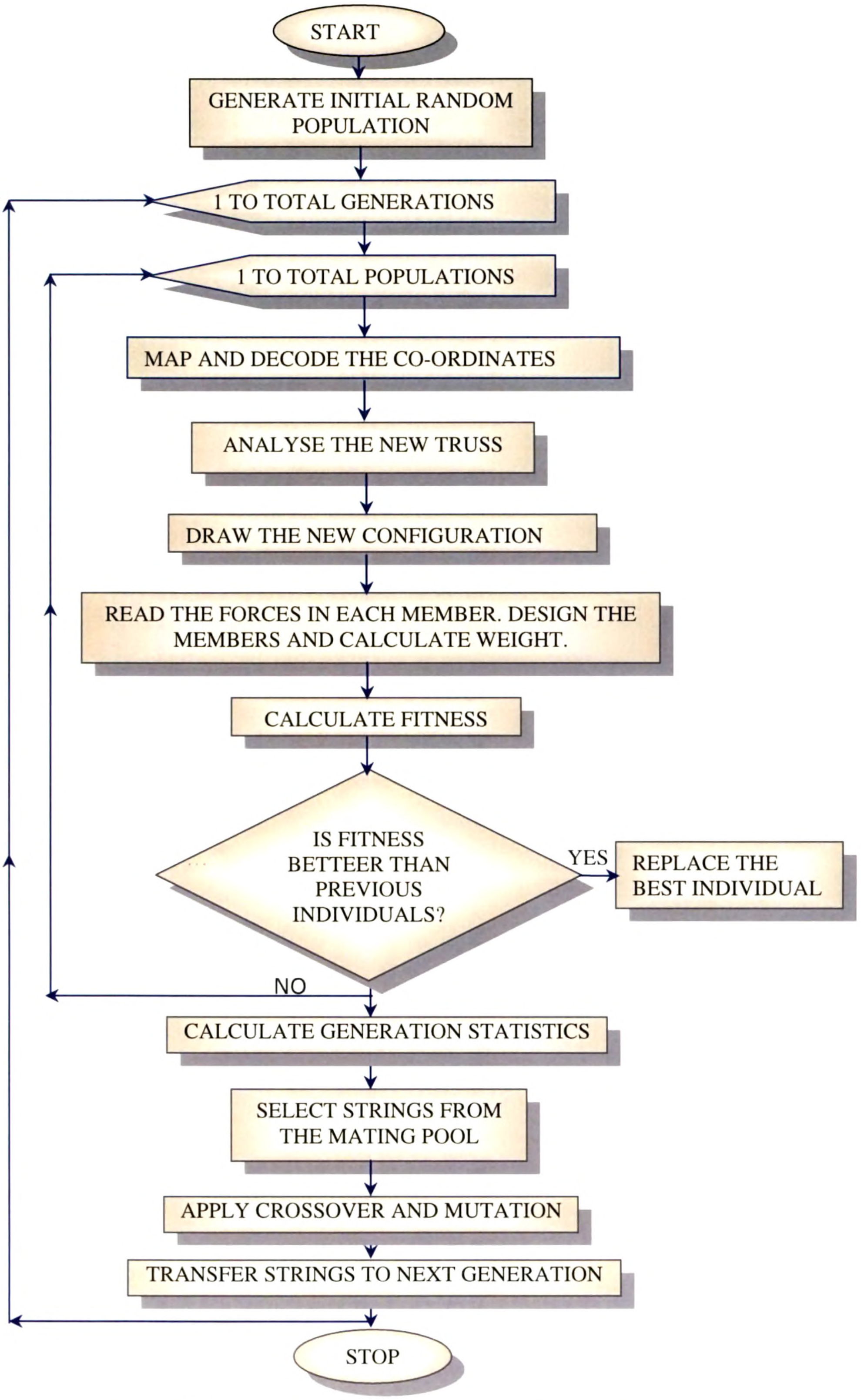
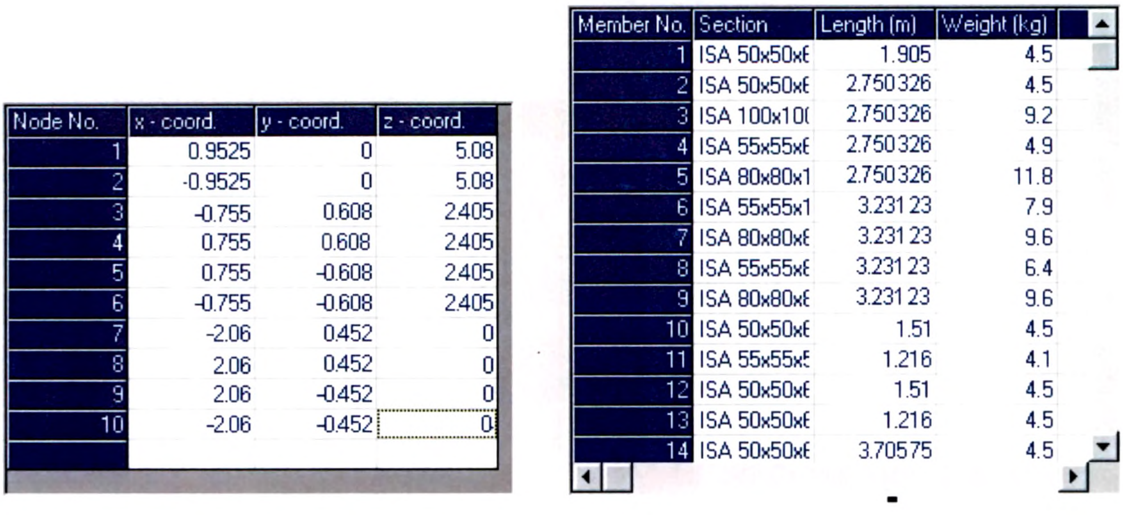

START
GENERATE INITIAL RANDOM POPULATION
1 TO TOTAL GENERATIONS
1 TO TOTAL POPULATIONS
MAP AND DECODE THE CO-ORDINATES
ANALYSE THE NEW TRUSS
DRAW THE NEW CONFIGURATION
READ THE FORCES IN EACH MEMBER. DESIGN THE MEMBERS AND CALCULATE WEIGHT.
CALCULATE FITNESS
IS FITNESS BETTEER THAN PREVIOUS INDIVIDUALS?
YES
REPLACE THE BEST INDIVIDUAL
NO
CALCULATE GENERATION STATISTICS
SELECT STRINGS FROM THE MATING POOL
APPLY CROSSOVER AND MUTATION
TRANSFER STRINGS TO NEXT GENERATION
STOP

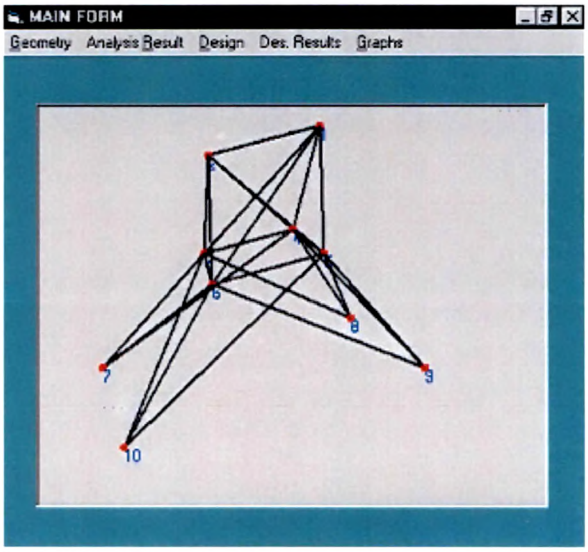
Fig 8.83 Flow Chart for Configuration Optimization



Node No.	x - coord.	y - coord.	z - coord.
1	0.9525	0	5.08
2	-0.9525	0	5.08
3	-0.755	0.608	2.405
4	0.755	0.608	2.405
5	0.755	-0.608	2.405
6	-0.755	-0.608	2.405
7	-2.06	0.452	0
8	2.06	0.452	0
9	2.06	-0.452	0
10	-2.06	-0.452	0

Member No.	Section	Length (m)	Weight (kg)
1	ISA 50x50x6	1.905	4.5
2	ISA 50x50x6	2.750326	4.5
3	ISA 100x10(	2.750326	9.2
4	ISA 55x55x6	2.750326	4.9
5	ISA 80x80x1	2.750326	11.8
6	ISA 55x55x1	3.23123	7.9
7	ISA 80x80x6	3.23123	9.6
8	ISA 55x55x6	3.23123	6.4
9	ISA 80x80x6	3.23123	9.6
10	ISA 50x50x6	1.51	4.5
11	ISA 55x55x5	1.216	4.1
12	ISA 50x50x6	1.51	4.5
13	ISA 50x50x6	1.216	4.5
14	ISA 50x50x6	3.70575	4.5

(a)



(b)

Fig. 8.84 Final results

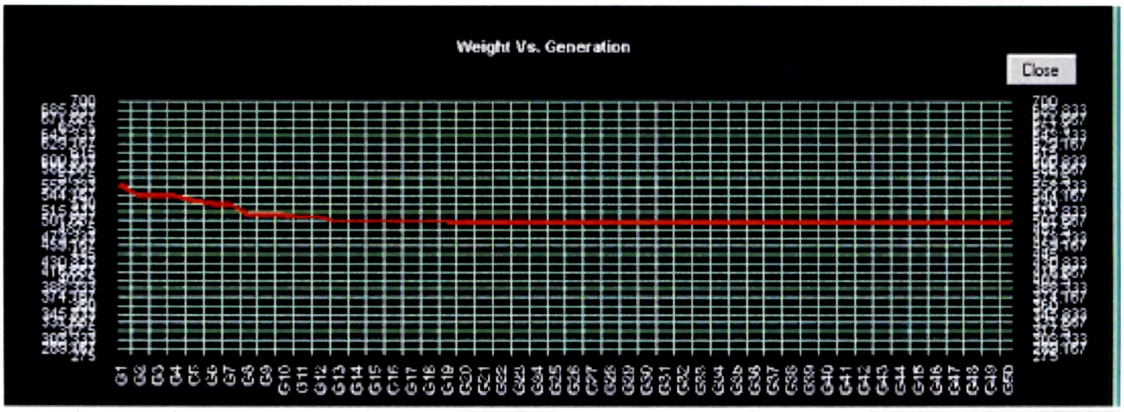


Fig. 8.85 Weight vs. Generations Graph Plotted in the Software for Space Truss

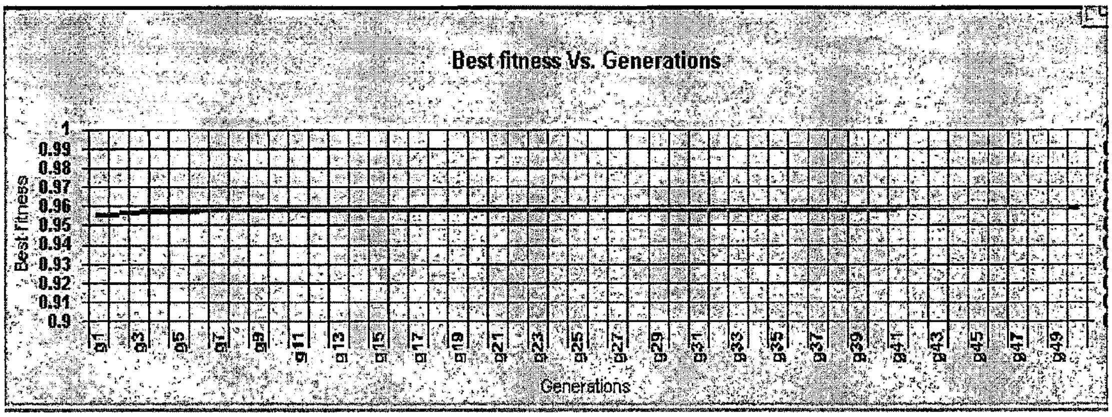


Fig. 8.86 Best fitness vs. Generations Graph for Space Truss

8.21 TOPOLOGY OPTIMIZATION OF SPACE TRUSS EXAMPLE

Truss topology optimization is one of the most interesting problems in structural optimization. This problem is described as one where a ground structure containing many points and members defines the discrete version of structural universe; and from which an optimal structure is derived. The application of the topology optimization method in various fields of engineering can significantly improve design cost and quality, which is important in global competition.

8.21.1 Problem Formulation

The method used in this study is based on the work of Kawamura and Ohmori [95]. The problem considered for the present work is that of a 3-dimensional cantilever truss as shown in Fig. 8.87. The same is also supplied as a ground structure to the algorithm. In this method a tetrahedron is considered as the basic unit of a space structure. Each tetrahedron element is produced one by one in such a manner that newly produced tetrahedrons are formed so as to be connected to the nodes having been generated up to the step directly before. These procedures are repeated until both supporting and loading points are included as one of the selected nodes. The topologies produced by the present method are guaranteed to have neither needless members nor undesirable overlaps between structural members, and also confirmed to be always stable structures. The present method has a strong advantage especially in a 3-dimensional problem because it is more difficult to produce candidates, which have a stable truss topology when we use ordinary production methods. The information made available from the ground structure is as follows:

1. There are a total of 18 nodes the locations of which are fixed.
2. The support nodes are 1, 9 and 11.

3. The loaded nodes are 4 and 7.
4. Thus the total number of free nodes is 13.

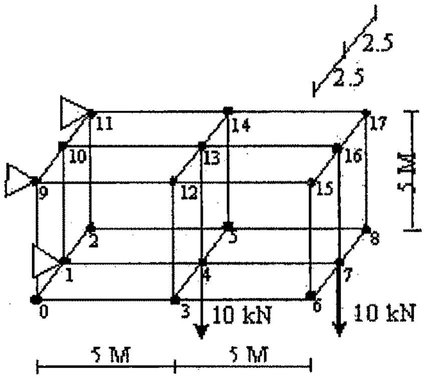


Fig. 8.87 Initial Ground Structure

8.21.2 Generation of Structural Topologies

At first, three points are randomly selected in the available area to make the first triangle in such a way that those points are not on the same line. The fourth point is also selected randomly such that it is not in the plane of the previous three triangles so as to form a tetrahedron. To form the next tetrahedron, three nodes are selected from the points selected up to the present stage and the fourth node is selected from the nodes, which have not yet been selected in the available area and is connected to those points. These newly produced members will be stable. Until all required points namely supporting and loading points are included in the selected nodes, the same operations are continued. The process of generation of initial population is illustrated in Fig. 8.88 in six phases.

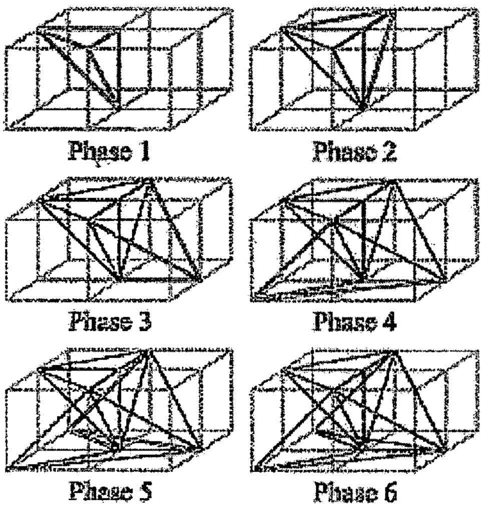


Fig. 8.88 Generation of Initial Truss Topologies

String Representation: A combination of binary and integer coding is used here. The information about the tetrahedral element is hidden in 8 bits, which form a part of the chromosome. The binary part i.e. the number '0' or '1' indicates the nodes in the corresponding plane. The nodes in the ground structure are now identified as shown in Fig. 8.89.

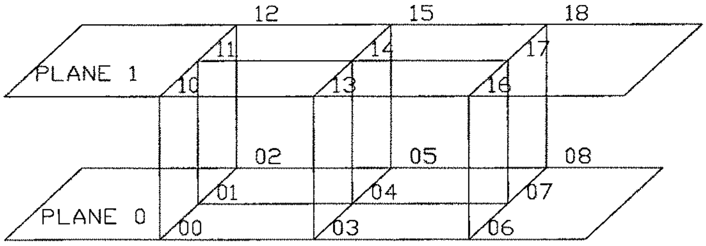


Fig. 8.89 Identification of Nodes

Thus, a chromosome as shown in Fig. 8.90 represents a topology shown in Fig. 8.88.

1	1	0	4	1	3	1	4	1	1	1	4	0	4	1	5
1	5	1	3	0	4	0	7	1	3	0	4	0	7	0	0
0	4	1	5	0	7	0	2	— — —							

← Unit of a chromosome →

Fig. 8.90 Chromosome Structure

Crossover Operator: The crossover used here is a single point crossover. The crossover point is randomly selected on each parent and it is divided into the former and the latter parts. Now crossover is performed. If the latter part of the second parent cannot be joined to the former part of the first parent, nodes, which can be connected to the former part, are used and the remaining nodes are generated randomly. If the produced topology does not include all the required points, new tetrahedrons are produced till all the points are included.

Mutation Operator: Uniform mutation has been used in the present work. A unit of chromosome is selected randomly. From this unit a node is randomly selected and is replaced by another feasible node such that the unit still decodes to a tetrahedron.

8.21.3 Objective Function

For the present work, objective function is formulated as minimizing the weight of the structure. When the aim of optimization is to find a topology whose weight is minimum and which satisfies a certain set of constraints a simple fitness function can be adopted as follows:

$$F = \frac{1}{W} \prod_{i} \gamma_i \qquad \ldots (8.100)$$

where γ_i is the penalty term for ith constraint.

8.21.4 Constraints

The various constraints considered are as under:

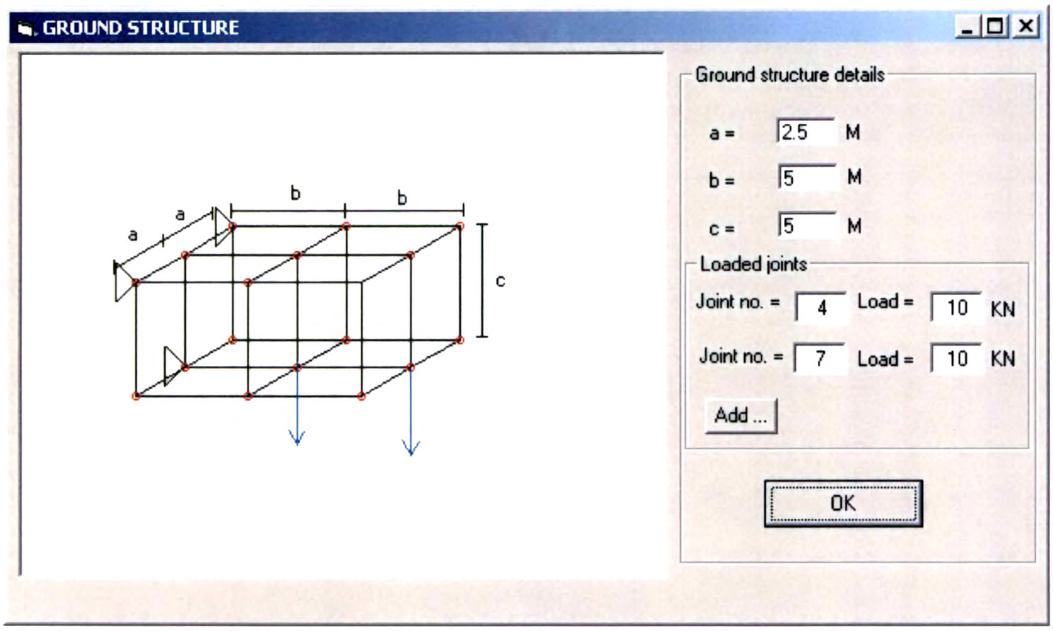
1. Permissible stress for compression and tension members as per IS: 800 [83] = 0.6 f_y = 150 N / MM^2 where f_y is the yield strength of steel.
2. The permissible slenderness ratio considered for compression and tension members is 250 and 400 respectively.
3. Allowable nodal displacement = 8 MM.

All the above constraints are handled either by using the penalty approach or selecting a higher section every time a constraint is violated.

The selection of genetic parameters is very important for convergence to occur. The following values of genetic parameters are used here.

1. Generations – 100
2. Populations – 200
3. Crossover probability – 0.7
4. Mutation probability – 0.003.

The details of the ground structure are supplied through the form shown in Fig. 8.91. The algorithm is then run and the gradual convergence towards optimum can be seen in Figures 8.92 to 8.96. The structure to the right in each case indicates the best-so-far solution in the run. The structure to the left indicates the current population. In figures, Blue, orange and green colours indicate respectively, compression, tension member and null members.



Ground structure details

a = 2.5 M

b = 5 M

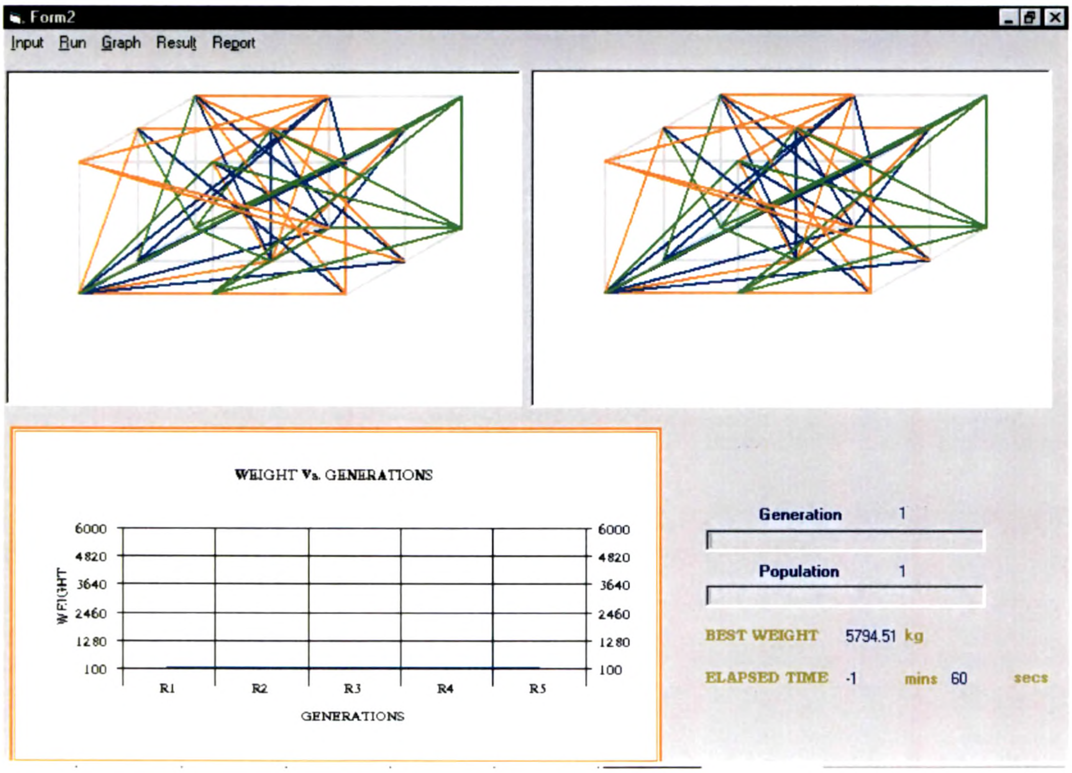
c = 5 M

Loaded joints

Joint no. = 4 Load = 10 KN

Joint no. = 7 Load = 10 KN

Fig. 8.91 Input Data Form to Define Ground Structure



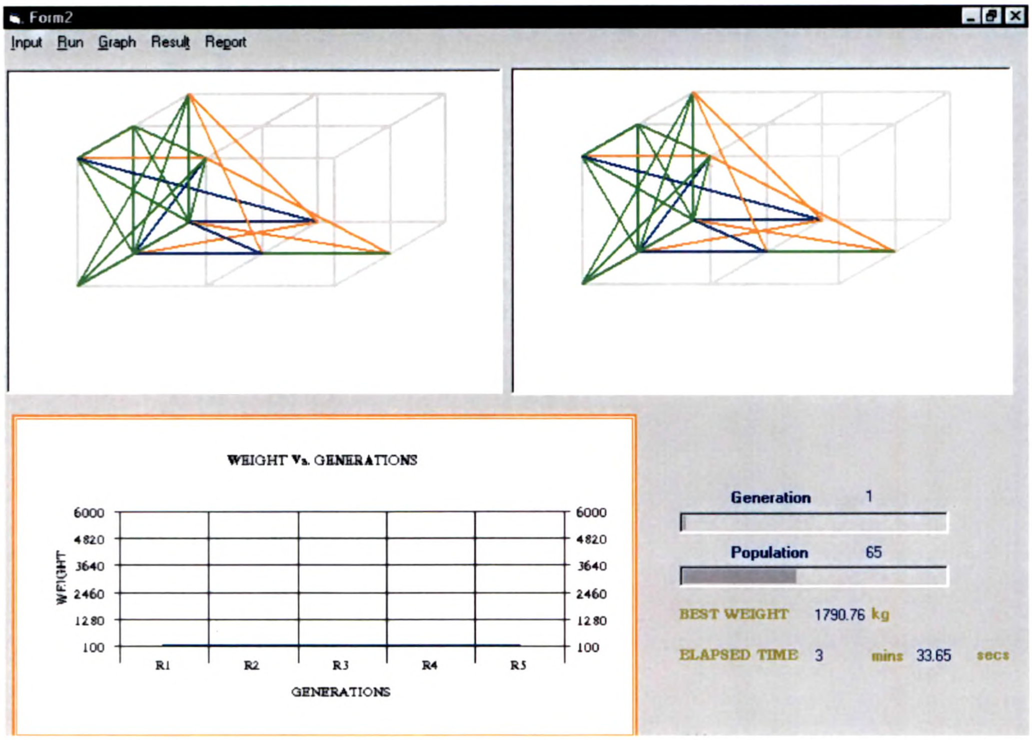
Generation 1

Population 1

BEST WEIGHT 5794.51 kg

ELAPSED TIME -1 mins 60 secs

Fig. 8.92 Solution in First Generation



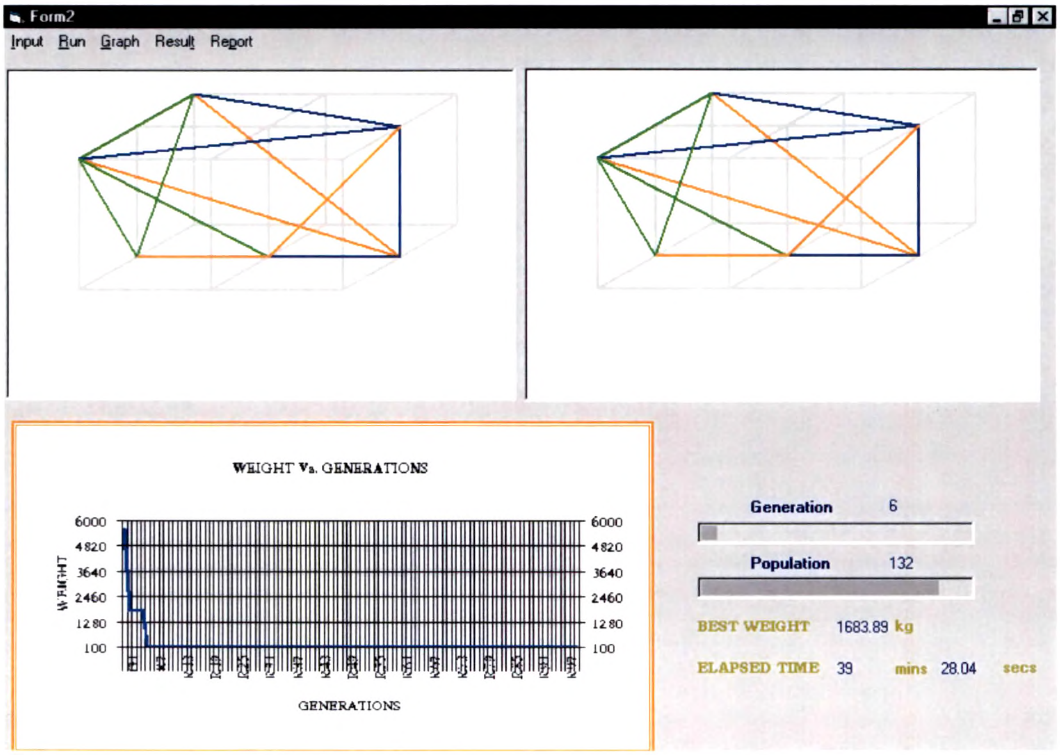
Generation 1

Population 65

BEST WEIGHT 1790.76 kg

ELAPSED TIME 3 mins 33.65 secs

Fig. 8.93 Solution in 65th Population of First Generation



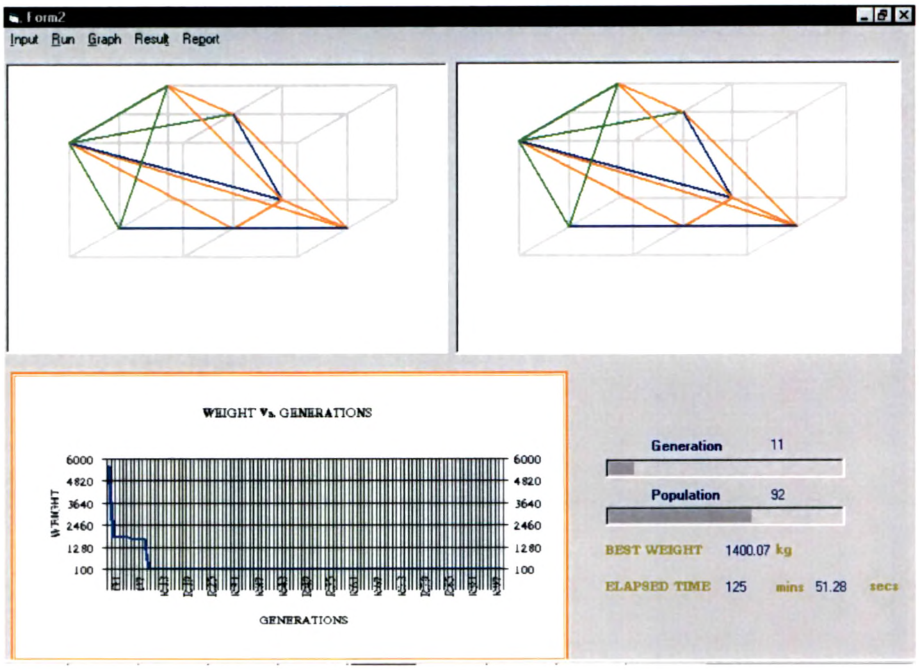
Generation 6

Population 132

BEST WEIGHT 1683.89 kg

ELAPSED TIME 39 mins 28.04 secs

Fig. 8.94 Solution in 6th Generation



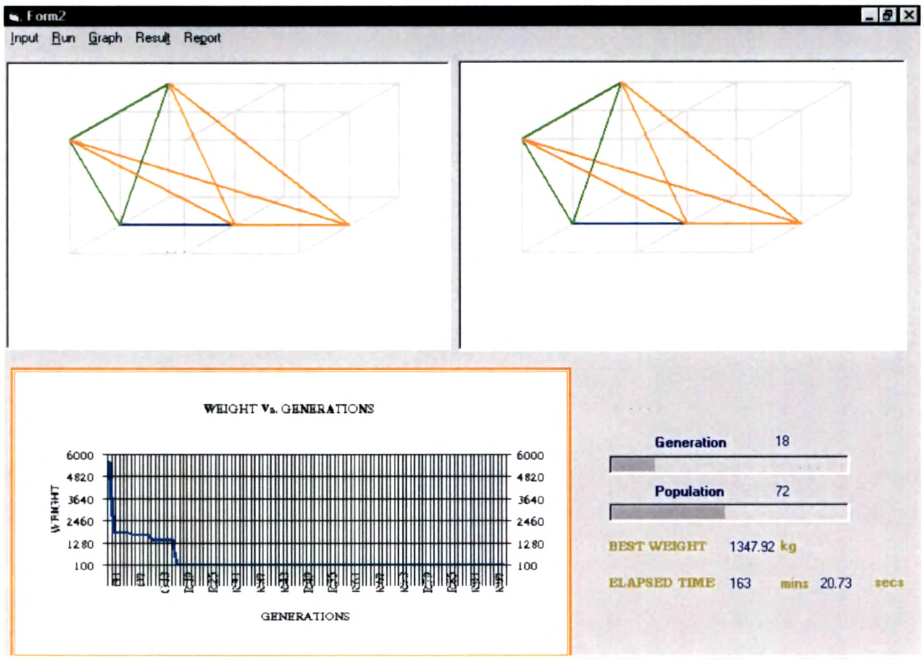
Generation 11

Population 92

BEST WEIGHT 1400.07 kg

ELAPSED TIME 125 mins 51.28 secs

Fig. 8.95 Solution in 11th Generation



Generation 18

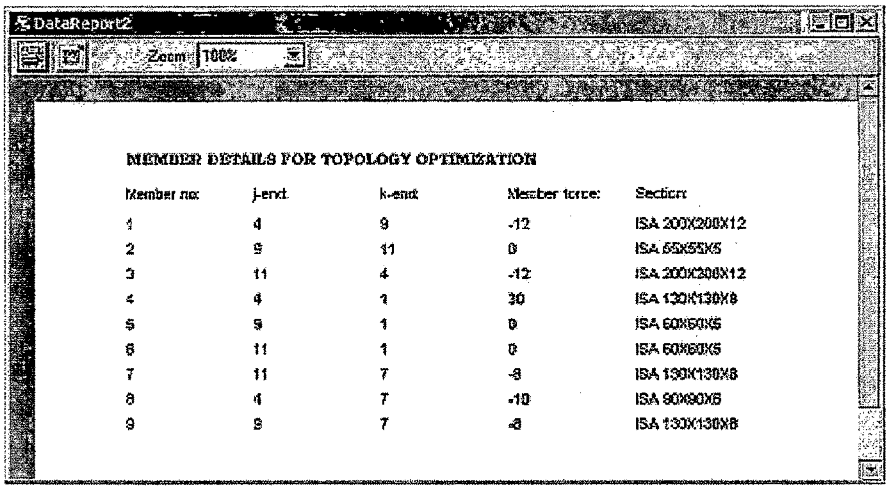
Population 72

BEST WEIGHT 1347.92 kg

ELAPSED TIME 163 mins 20.73 secs

Fig. 8.96 Final Solution

The selected sections from the discrete data set in the final solution are displayed in Fig. 8.97. The graph of best weight versus generations is also shown in all the figures and indicates gradual convergence. Elitism has also been used in the present work to improve the efficiency of the algorithm. The software provides a report of topology data and genetic data supplied as input. It also facilitates the user with the display of forces developed in the members. It can be verified that the structure obtained can be represented by two tetrahedrons and is thus stable.



MEMBER DETAILS FOR TOPOLOGY OPTIMIZATION

Member no:	j-end	k-end	Member force:	Section:
1	4	9	-12	ISA 200X200X12
2	9	11	0	ISA 65X55X5
3	11	4	-12	ISA 200X200X12
4	4	1	30	ISA 130X130X8
5	9	1	0	ISA 60X60X5
6	11	1	0	ISA 60X60X5
7	11	7	-8	ISA 130X130X8
8	4	7	-10	ISA 90X90X6
9	9	7	-8	ISA 130X130X8

Fig. 8.97 Final Sections Selected for Each Member of Space Truss

8.21.5 Results and Discussions

The results obtained by Kawamura et al [95] are shown in Fig. 8.98 and the same is reproduced in Fig.8.99 (b). The optimized truss topology is as shown in Fig. 8.96. It can be seen that the structure indicated above has three null members indicated in green. These members do not carry any load and also do not contribute to the stability of the structure. Hence, may be removed. The structure after the removal of null members is reproduced in Fig. 8.99 (a). The stability of the two structures is checked in Table 8.14.

Sectional Area	5 to 20 cm^2 ($\Delta = 1.0$ cm^2)
Young's Modulus	21 GPa
Displacement Limit	2 mm
Stress Limit	15 MPa

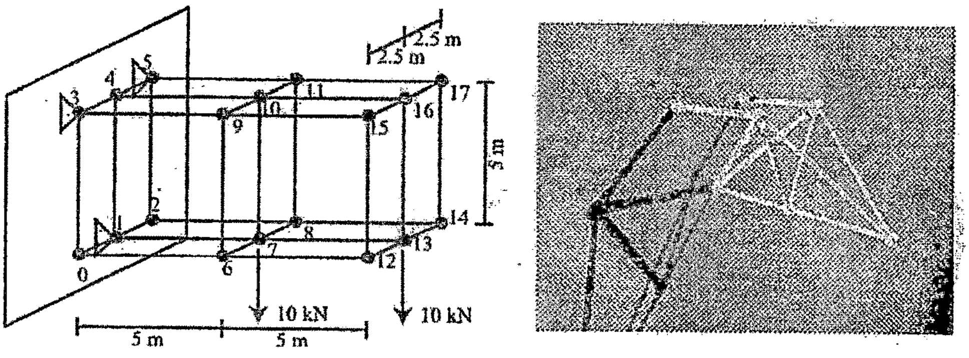


Fig. 8.98 3D Cantilever Truss Topology Optimization

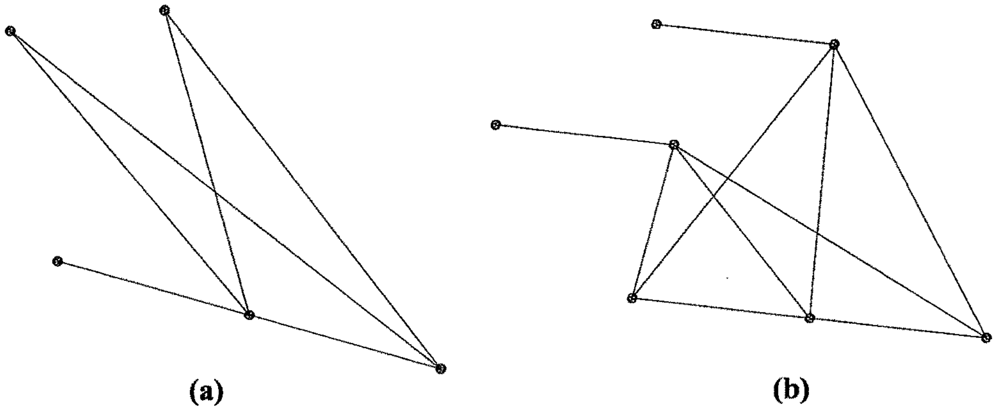


Fig. 8.99 Truss after Removal of Null Members

Table 8.13 Stability check

Degree of indeterminacy	Fig. 8.93 (a)	Fig. 8.93 (b)
$d_{se} = R - 6$	$= 9 - 6 = 3$	$= 9 - 6 = 3$
$d_{si} = m - (3j - 6)$	$= 6 - (3(5) - 6) = -3$	$10 - (3(7) - 6) = -5$
$d_T = d_{se} + d_{si}$	0	-2
d_{se} – external degree of static indeterminacy, d_{si} – internal degree of static indeterminacy, d_T – total degree of static indeterminacy, R – number of reactions, m – number of members, j – number of joints.		

Thus it is seen that the topology obtained in the present work is stable whereas that of reference [95] is unstable.

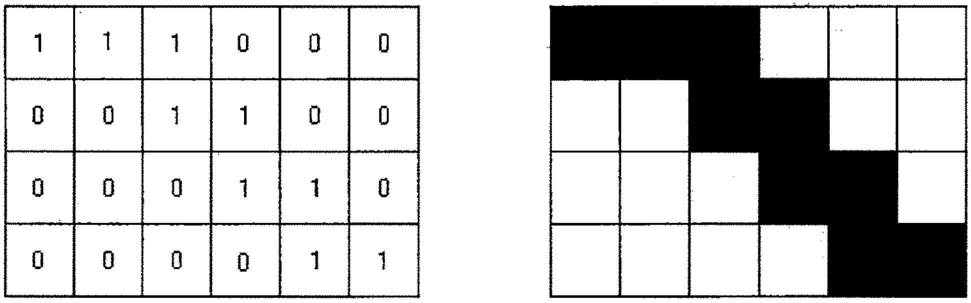
8.22 TOPOLOGY OPTIMIZATION OF CONTINUUM STRUCTURES

Design connectivity is an important issue for structural topology design optimization. A final structural topology must be a single connected object so that it can bear external loadings and its finite element analysis can be carried out for function and constraint evaluation in the optimization. The importance of structural connectivity in design is emphasized by considering the total number of connected objects of each individual explicitly in an equality constraint. A violation penalty method is used to reduce number of connected objects to one and to drive the GA search towards the topologies with higher structural performance and less unusable material.

A bit-array representation scheme is implemented to describe the topology as it is an intuitive and straight forward method to represent the two-dimensional topology for the optimum design problems using the GAs. A bit-array or binary-string is mapped into the two-dimensional design domain discretized by a fixed regular mesh, where each of the small, rectangular elements contains either material or void and is thus treated as a binary design variable. An identical population initialization method is used as a supplement to the commonly adopted procedure of randomly generating the initial population to improve the convergence of GA towards optimum solution.

8.22.1 Design Variables

The bit-array representation is adopted as the chromosome representation method to define the distribution of material in a two-dimensional design domain. This representation method is based on a given regular mesh and the length of each chromosome is thus fixed, where elements with allele values of 1 become material while those with allele values of 0 become void as shown in Fig. 8.100. The two-dimensional design domain discretized by a fixed regular mesh, where each of the small, square elements contains either material or void is thus treated as a design variable.



1	1	1	0	0	0
0	0	1	1	0	0
0	0	0	1	1	0
0	0	0	0	1	1

(a) Bit-array representation **(b) Resulting topology**

Fig. 8.100 Coding of Structural Topology

8.22.2 Objective Function

Being a steel continuum structure weight of the steel plate is taken as an objective function as the cost is directly proportional to the weight in the steel structure. The weight of the plate is calculated by adding weights of all the elements in the topology.

8.22.3 Design Constraints

In the present study maximum principal stress constraint is imposed as an inequality constraint.

$$\sigma_{\text{actual}} - \sigma_{\text{perm}} \leq 0 \qquad \dots (8.101)$$

where σ_{actual} is resultant stress in the plate element under loading and σ_{perm} is permissible resultant stress in the plate element.

The corresponding constraint function is:

$$\max\left(\frac{\sigma_{actual}}{\sigma_{perm}} - 1, 0\right) \qquad \dots (8.102)$$

The number of connected objects of each topology in the design domain is explicitly adopted as an equality constraint function.

$$N_c - 1 = 0 \qquad \dots (8.103)$$

where N_c is the total number of connected objects in the topology.

Figure 8.101 shows an example in which three different connected objects can be found and thus violates the equality constraint.

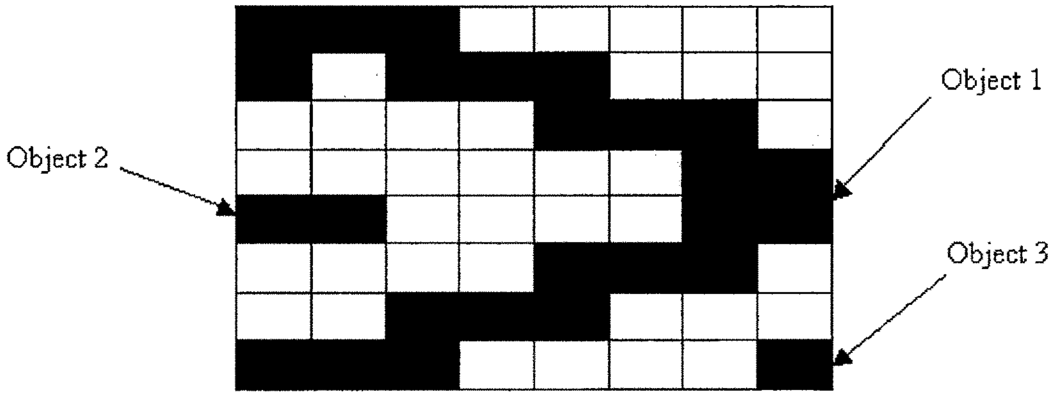


Fig. 8.101 Connected Objects in Design Domain

An artificial unconstrained objective function of the constrained optimization can be constructed as:

$$F(x) = \begin{cases} W(x) & \text{if } x \in F' \\ \widetilde{W} + viol(x) & \text{otherwise,} \end{cases} \qquad \ldots (8.104)$$

where F(x) is the artificial unconstrained objective function, F' is the feasible region of the design domain R, $\widetilde{W}$ is the objective function value of the infeasible solution in the population and $viol(x)$ is the summation of all the violated constraint function values.

Design connectivity is an important issue that must be handled in the structural topology design to ensure the success of the GA. To satisfy the constraint on the connectivity, the final topology must be a single connected object. According to the violation penalty method, if the total number of connected objects of a topology in the design domain is greater than 1, the summation of all violation values viol(x) is formulated as follows:

$$viol(x) = \Gamma c * (Nc - 1) + \Gamma a * \sigma \qquad \ldots (8.105)$$

Here Γc is the penalty multiplier for the total number of connected objects of the topology and Γa is the penalty multiplier for the total stress σ of the objects in which stress values are more than allowable value. The values of these two penalty multipliers are problem dependent, but the former should be much larger than the latter to ensure that the constraint on the number of connected objects is more heavily penalized to help in reducing the total number of connected object more significantly.

Fitness function is derived from the objective function and used in successive genetic operations. As the problem of minimization, the fitness has been worked out using following formula:

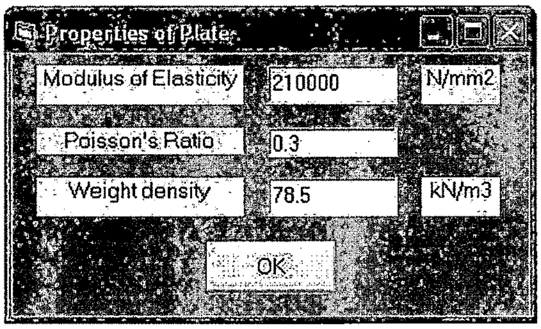
$$\text{Fitness} = \frac{1}{(1+F(x))} \qquad \ldots (8.106)$$

8.22.4 Processors Developed

A software based on Visual Basic 6.0 is developed with number of forms, menus etc. for topology optimization of plates. The software is developed in modular form with three processors i.e. Pre-processor, Main-processor and Post-processor.

Pre-processor: This processor provides user an easy input data entry and displays the data entered in a graphical form. It also has the facility of editing the data. Figures 8.102 to 8.106 show various forms belonging to this processor.

Main-processor: In this processor, the data supplied by the user are assigned to variables and used for further calculation. Some of the subroutines and/or functions developed under this processor are Draw, Discretize, KeMatrix1, ChromAssign, ChromCheck, GridDraw, KMatrix1, DeqKinR, StringFitness, Statistics, Maximum1, NoConObj, etc.

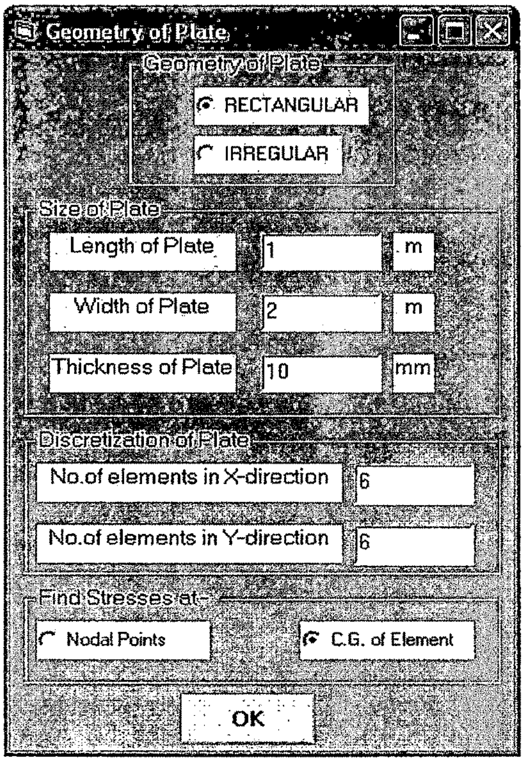


Modulus of Elasticity 210000 N/mm2

Poisson's Ratio 0.3

Weight density 78.5 kN/m3

Fig. 8.102 Material Property Form



Geometry of Plate

● RECTANGULAR

○ IRREGULAR

Size of Plate

Length of Plate 1 m

Width of Plate 2 m

Thickness of Plate 10 mm

Discretization of Plate

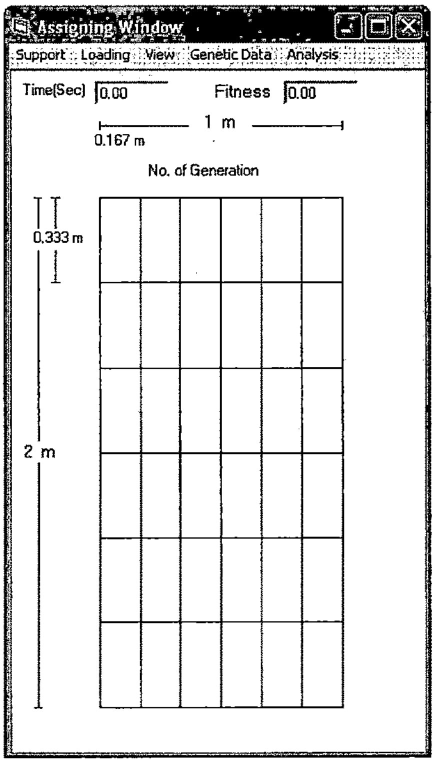
No.of elements in X-direction 6

No.of elements in Y-direction 6

Find Stresses at

○ Nodal Points ● C.G. of Element

Fig. 8.103 Form for Defining Geometry and Grid

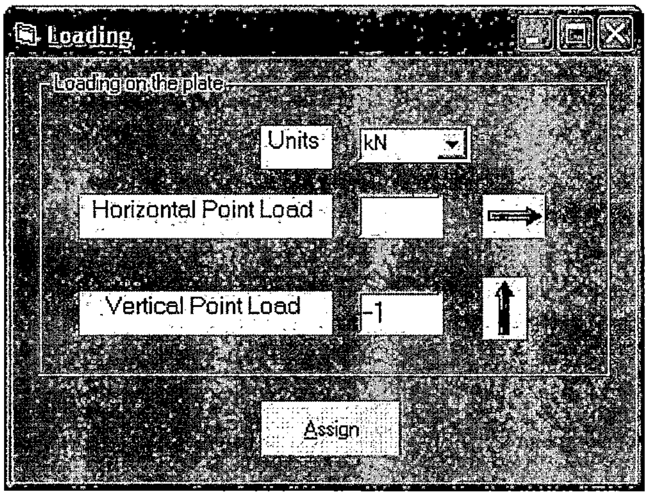


Time(Sec) 0.00

Fitness 0.00

Fig. 8.104 Display of Geometry

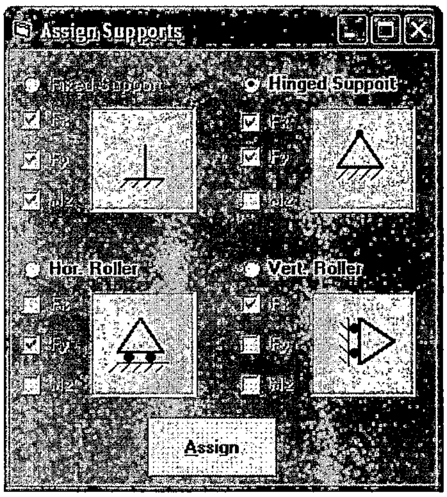
Post-processor: The results received from the main-processor are displayed on the screen in graphical form using this processor. Figures 8.107 and 8.110 are the screens shots showing output results developed by this processor.



Units: kN

Horizontal Point Load:

Vertical Point Load: -1



○ Fixed Support
☑ Fx
☑ Fy
☑ Mz

● Hinged Support
☑ Fx
☑ Fy
□ Mz

○ Hor. Roller
□ Fx
☑ Fy
□ Mz

○ Vert. Roller
☑ Fx
□ Fy
□ Mz

Fig. 8.105 Form for Load Data

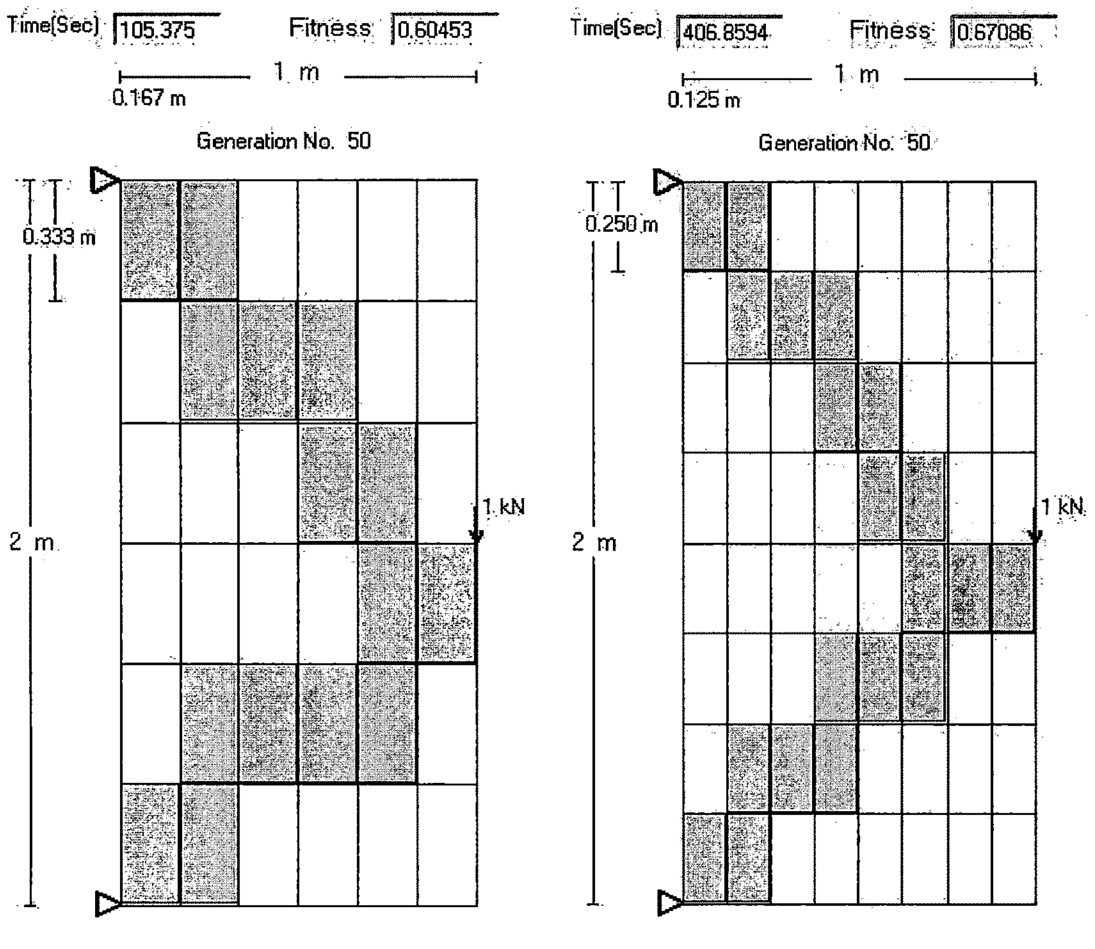
Fig. 8.106 Form for Assigning Support

8.22.5 Illustrative Example

A cantilever steel plate of size 1 m length x 2 m width with the supports on the left boundary and a unit point force applied vertically downward at half-height of the right boundary is studied. The thickness of the plate is taken as 10 mm. The plate is analyzed using finite element method considering it as a plane stress problem. The plate is discretized into 6 x 6 mesh, 8 x 8 mesh and 10 x 10 mesh. The plate is considered of steel with Young's modulus of elasticity = 2.1 x 10^5 N/mm^2, Poisson's ratio = 0.3 and Weight density = 78.5 kN/m^3.

GA is used with a population size of 15, crossover probability of 0.70, mutation probability of 0.01 and maximum generations of 50. Combined roulette wheel-tournament selection scheme is used. To guarantee the structural connectivity, an identical population initialization is taken into consideration. Also, in each discretization case, boundary and loading points are not disturbed during topology optimization.

Figure 8.107 shows resulting topologies and time taken for 6 x 6 mesh and 8 x 8 mesh. Figures 8.108 and 8.109 show the generation history for 6 x 6 mesh and 8 x 8 mesh respectively. Figure 8.110 shows resulting topology for 10 x 10 mesh and Fig. 8.111 shows its generation history.



Time(Sec) 105.375 Fitness 0.60453

Generation No. 50

Time(Sec) 406.8594 Fitness 0.67086

Generation No. 50

a) Weight = 0.65 kN

b) Weight = 0.48 kN

Fig. 8.107 Final Topology for 6 X 6 Mesh and 8 X 8 Mesh

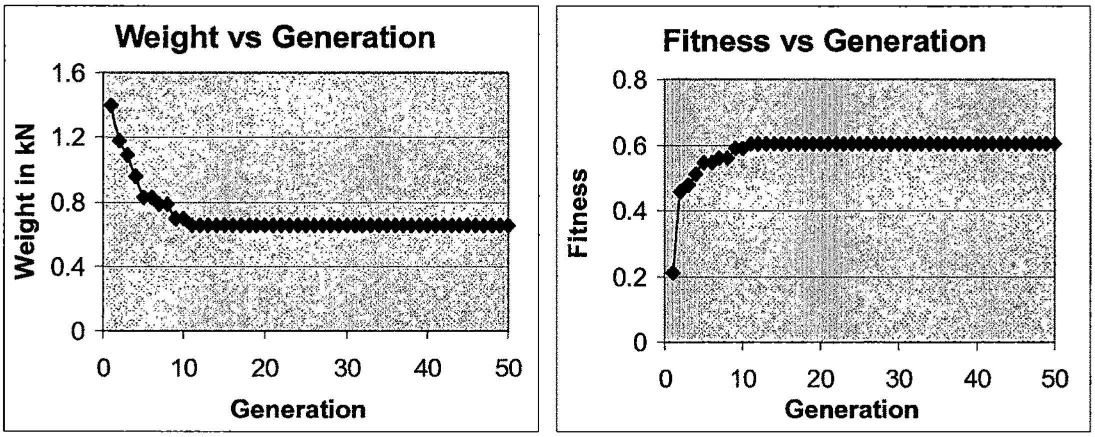


Fig. 8.108 Generation History for 6 X 6 Mesh

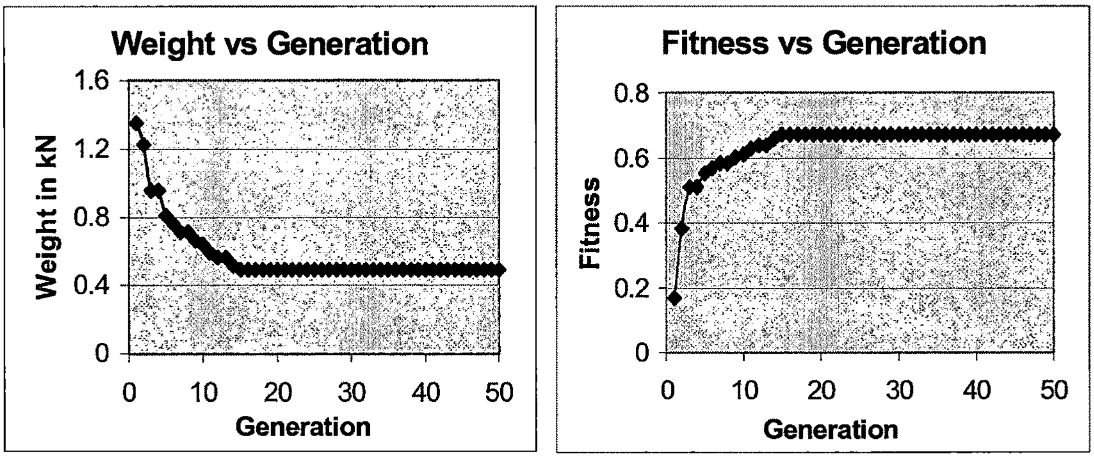
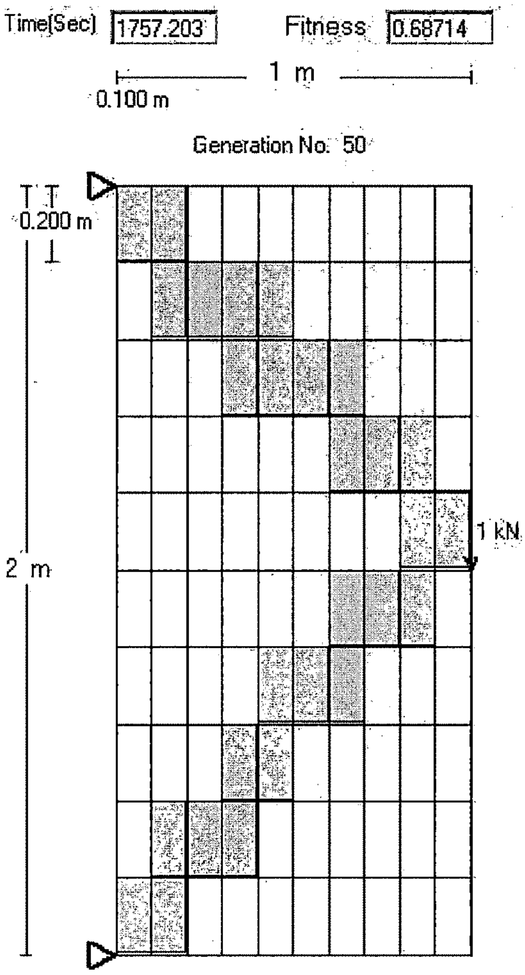


Fig. 8.109 Generation History for 8 X 8 Mesh



Time[Sec] 1757.203

Fitness 0.68714

Generation No: 50

Weight = 0.43 kN

Fig. 8.110 GA Based Final Topology for 10 X 10 Mesh

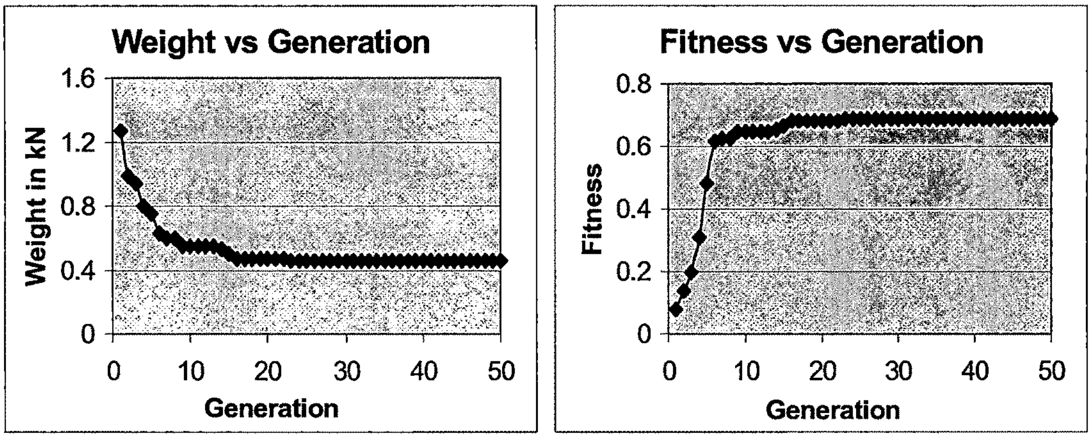


Fig. 8.111 Generation History for 10 X 10 Mesh

8.22.6 Comparison of Results

Table 8.14 shows the comparison of the GA results. From the comparison, it can be seen that as the mesh becomes finer, better solution is obtained, but time consumed by software is more.

Table 8.14 Effect of Mesh Size on the Results

Sr. No.	Mesh size	Weight (kN)	Fitness	Time (Sec)
1	6 x 6	0.65	0.605	105.375
2	8 x 8	0.48	0.671	406.856
3	10 x 10	0.43	0.687	1757.203

Optimum topology for 10 x 10 mesh obtained from suggested approach is modeled in commercially available software and the stresses are compared. The results are very nearer to each other. The comparison of GA is shown in Table 8.15

Table 8.15 Comparison of Maximum Stress in kPa

Sr. No.	Stage	GA	STAAD.Pro
1	Initial	53.622	52.859
2	Final	84.388	88.226

8.23 CLOSING REMARKS

In this chapter GA was selected as a tool for optimization of engineering structures. The choice of programming language is an important phase of any algorithm to be effective, attractive and user friendly. The latest version of Microsoft Visual Basic was used to develop GA based software in this chapter and a wide variety of RCC and steel structure have been addressed to check the suitability and performance of GA considering different design variable types and different constraints. Sizes of the problems considered in the present study varied form small having few variables (up to 10) and moderate having more variables (up to 25) to large having many variables such as topology optimization of continuum structures.

During development of the program based on GA, it has been found that once prepared, algorithms of various GA operations (random solution generation, decoding, selection, elitism, crossover, mutation, scaling of fitness etc.) can be reused or called in different optimization problems. Only operation which is problem dependent is objective function and fitness evaluation which requires thorough knowledge of the problem at hand. Thus some of the modules of GA based software are problem dependent whereas some of the modules are universal.

The graph showing generation history in Fig. 8.5 obtained in optimization of water tank clearly indicates that during the initial generations the convergence is fast but takes more time to converge in the later generations. This is due to the reason that during initial generations the population contains diverse solutions from various corners of the search space and hence can pick up the right hill very fast. Once near optimal solution is found on the right peak more and more solutions will start concentrating in that region and thus reduces the diversity of population. This nature was noticed in almost all the convergence graphs.

In the size and configuration optimization of trusses as number of joints and members do not change structural stability the constraints violation issues could be handled easily. Whereas in topology optimization problem as number of joints and number of members may change during search the search space increases significantly and therefore it becomes very difficult for the optimization algorithms to find the optimal solution. The constraint handling strategy adopted in the work was so effective that the constraint of unstable structure, excessive member length etc. could be easily resolved.

In the constraint handling techniques used in the topology optimization penalty is applied by increases the weight of only those members which violate the constraints and does not penalize the solution string severely by increasing the weight of whole structure if only few members violate the constraints. This method does not kill the penalized string suddenly but gives a chance of getting selected and retains the diversity in the search process. The reason for selecting this method for topology optimization is that number of constraints is quite a large and in the initial generations chances of getting fully feasible string is very limited and if the strings violating constraints by small amount is severely penalized then there are very rare chances of evolving near optimal solution and the algorithm may get struck to local optimum very soon.

The topology optimization problem of plane truss has also been tried by using the configuration optimization algorithm for four different ground structures. The solution of this problem depends to great extent on the initial ground structures and algorithm does not have capability to get innovative shapes. But this problem was satisfactorily resolved in the topology optimization algorithm presented in section 8.18.