CHAPTER 4

GA BASED SOLUTION PROCEDURE FOR
CONDUCTOR GRADATION PROBLEM

4.1 INTRODUCTION

In a radial feeder with the loads distributed along the feeder main, the power flows in different sections vary. The power flow in sections closer to the substations is high while the same in the sections away from the substations is low the maximum power flow being in the closest section and the lowest power flow being in the farthest section. For the sections carrying low power, the conductor can be of lower size, just adequate to carry the power whereas higher conductor sizes are required for the sections carrying large power. Use of the highest size of conductor throughout the feeder main will mean more investment than required. But use of lower conductor size in the farthest sections will increase the power and energy loss. However there lies an optimum level that could be achieved through conductor gradation along the feeder main. This chapter presents a solution procedure to solve this problem of grading the conductor sizes in the feeder sections of radial feeders. The mathematical models and problem formulation has already been discussed in the section-2.2.2 of Chapter-2. The objective of this work is to optimize the use of graded conductors in the radially operated feeders in order to achieve a reduction in system cost, while maintaining the customer voltages within the declared limits. This problem has been studied by a few researchers in the past using conventional methods. However, from the beginning of the last decade, the researchers are concentrating on developing fast, effective and efficient optimization solution procedures. Genetic Algorithm (GA) and Simulated Annealing (SA) algorithm are two examples of methods derived from nature. A GA based solution procedure developed for the problem is presented in this chapter.

4.2 EXISTING GA BASED METHODOLOGIES

As discussed in chapter-2, this problem of conductor grading for radial feeders was first studied by M.Ponnavaikko and K.S.P Rao, using Dynamic Programming and Method
of Local Variations methods. No solution procedures using Genetic Algorithmic approach, to solve this type of problems have been developed so far. In this research, a GA based solution procedure has been developed to solve this problem. The proposed genetic algorithmic approach is presented in this chapter.

4.3 PROPOSED GENETIC ALGORITHMIC APPROACH

4.3.1 Introduction

Let us consider a distribution feeder main with all sections using ‘Rabbit’ conductor as shown in fig-4.1. The problem is to determine the optimal conductor sizes for each of the feeder sections from among the fixed set of conductor sizes considered. The number of alternatives that can be considered for the problem are large, which increases exponentially with the size of the problem. The objective of this research is to determine the optimal set of conductor sizes for the feeder. The mathematical models developed by Ponnavaikko et. al is considered and a GA based solution approach has been developed in this research. For this purpose, the problem has to be formulated within the framework of Genetic Algorithm.

![Fig-4.1 A radial distribution feeder with single conductor size](image)

As discussed earlier, the solution procedure starts with definition of chromosomes [77, 80-83, 85-86] and the generation of an initial population. The initial population is improved through a reproduction process. Thus, the population reproduced is placed in the mating pool for operations like cross-over and mutation. At every stage the feasibility of
the solutions are checked and validated. The procedure is repeated for a given number of
generations or until the solutions gets stabilized.

4.3.2 Mathematical Formulation of the Problem

The objective function [91] of the problem as discussed in chapter-2 is stated as

Minimize

\[ Z = \sum_{i=1}^{ND} \left( \frac{K_{4i}}{a_{c_i}} + K_{5i} a_{c_i} \right) \] (2.21)

Subject to,

\[ \sum_{i=1}^{ND} \left( \frac{K_{2i}}{a_{c_i}} \right) \leq \Delta v' \] (2.22)

and \( a_{c_1}, a_{c_2}, ..., a_{c_{ND}} \) (2.23)

Where, \( K_{2i}, K_{3i}, K_{4i}, \) and \( K_{5i} \) as described in Appendix-C are reproduced below.

\[ \Delta v'_m = \frac{\Delta v_m}{(1 + g)^M} - \sum_{i=1}^{ND} K_{3i} \] (2.24)

\[ K_{2i} = \sqrt{3} l_{m_i} \ell_i \rho \cos \theta \] (2.25)

\[ K_{3i} = \sqrt{3} l_{m_i} \ell_i x \sin \theta \] (2.26)

\[ K_{4i} = 26.28 \rho \ell_i l_{m_i}^2 \left[ \sum_{k=1}^{M} \frac{(1 + g)^{2k} (LLF_k) c_{e_k}}{(1 + u)^k} + (1 + g)^{2M} \sum_{k=M+1}^{NLF} \frac{(LLF_k) c_{e_k}}{(1 + u)^k} \right] \] (2.27)

\[ K_{5i} = f_i \ell_i \] (2.28)

To define the problem within the framework of GA, we have to define the
individual members of the population of the solution space. This member is called a
chromosome. A chromosome represents a solution of the problem. In electrical terms this chromosome consists of a set of elements representing the feeder sections in terms of conductor sizes. In GA terms a chromosome is a string of integers. The problem considers different conductor sizes for the feeder segments of a given radial distribution feeder which follows a tree configuration. A tree consists of ND consumer nodes fed from a source node (substation node). Thus a tree has a total number of feeder segments equal to ND. Hence, a chromosome representing a solution has ND integers. Each chromosome is decoded to obtain the value of the cost function. That chromosome which results in a minimum value of the cost function (maximum value of the fitness function) is selected as the optimum solution from the final population arrived at. The data structure adopted for the GA procedure of the problem is given in Appendix-K.

Simple power flow algorithm using moment method is run at the beginning of the algorithm to determine $I_m$, $i = 1,...,ND$. Flow in the infeed edge of the tail node is the same as its load demand. Flow in all other edges is the sum total of the flows in infeed edges of its children nodes. Nodes that have no children are identified as tail nodes and the tree search routine starts updating the flow variables when it encounters these nodes.

4.3.3 Encoding of a Chromosome

In a GA approach, chromosome is the basic unit which represents a solution of the problem. It is essential to device a coding mechanism that represents all possible solutions in terms of chromosomes. There are two different encoding techniques popularly used, from the viewpoint of numeric representation namely (1) Binary Encoding and (2) Integer Permutation Encoding. In most of the applications a chromosome is a string of bits. For the problem considered, a chromosome has to represent the conductor sizes with respect to specific locations. Considering the discrete nature of conductor sizes and the feeder segment locations integer encoding is adopted. The problem considers different conductor sizes for the feeder segments of a given radial distribution feeder which is a tree configuration. If the locations of the feeder segments for codification into any chromosome can be fixed in an order, then a chromosome can simply be a string of integers representing a set of conductor sizes. Each element in a chromosome is a decision variable called as a gene in GA terminology. Consumer nodes can be listed in random order and are ordered
after the substation node. Each chromosome is decoded to obtain its value for the cost function defined by (2.21) to (2.28). The data structure adopted for the GA procedure of the problem is given in Appendix-K.

4.3.3.1 Integer Encoding

Consider the distribution feeder shown in fig-4.1. The feeder consists of nodes and links. The set of nodes includes one source node and the rest load nodes. The source nodes will not have infeeds. But each of the load nodes will have an infeed. The identity of a node and its infeed edge are one and the same. Conductor size of infeed edge of each node corresponds to a decision variable which is called as a gene. The conductor sizes considered for grading are coded as integers. A chromosome is constructed in terms of the code values that represent the conductor size of the infeed edge of each node. The elements of a chromosome representing the load nodes are listed along its length in numerical order of the nodes. The radial feeder shown in fig-4.1 consists of 13 load nodes from nodes 1 to 13. Hence the chromosome consists of 13 positions representing the nodes 1 to 13. The conductor data are given in Table-b.1 of Appendix-B. Three conductor sizes namely Squirrel (13mm$^2$ Cu, ACSR conductor), Weasel (20mm$^2$ Cu, ACSR conductor) and Rabbit (30mm$^2$ Cu, ACSR conductor) are considered for grading which are represented by integers of a set $B = \{1, 2, 3\}$ in order. Thus, each element $c_i$ of the chromosome representing the feeder section $i$ can assume a code between the integers 1 and 3. A chromosome representing a solution to the gradation problem is shown in fig-4.2.

$$
\begin{array}{ccccccccccc}
  c_1 & c_2 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & c_{13} \\
  3 & 3 & 3 & 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 & 1
\end{array}
$$

Fig-4.2 A chromosome of conductor gradation problem

For this problem, the chromosome will have 13 elements, since the feeder has 13 sections. To select values for the 13 elements of the chromosome randomly, random
number in the range of 1 to 3 are generated, since one of the three could be considered for each section. Since the feeder considered has no laterals branching, the gradation is obvious from the order of the example chromosome shown in fig-4.2. The nodes are numbered in sequential order from source end to the tail end or vice-versa. Since each node is identified by its parent and children nodes, the procedure can very well be used to study feeders with multiple laterals. This has been made possible because any transformation on a chromosome during crossover or mutation is performed after verifying the conductor sizes of the infeed of its parent node and/or that of its children nodes.

4.3.4 Decoding of a Chromosome

The solution is obtained in the form of a chromosome which consists of a string of integers each representing a decision variable. Each of these integer values need to be decoded to realize the conductor size of the corresponding feeder segment. Decoding refers to the realization and recording of the conductor cross-section of infeed segment of each of the nodes of the feeder represented by the corresponding integer code into the data module of nodes. Once a chromosome is decoded, the conductor sizes of the infeed edge of all nodes are recorded in their respective data modules. Now the feeder costs, the loss costs and voltage drops of all the ND segments required to determine the objective function value, the value for fitness function and the feeder voltage drop corresponding to the chromosome are obtained.

4.3.5 Solution Approach

4.3.5.1 Generation of Initial Population

The algorithm is designed to generate initial population, which are a set of alternate solutions (chromosomes), by using a heuristic constructive algorithm. As discussed earlier, each chromosome includes all the feeder sections as genes, each gene represented by an integer defining the conductor size of its infeed edge. An initial population is generated randomly from the conductor sizes considered. The size of the population should be fixed judiciously. Larger the population size, faster will be the convergence to the solution and lesser will be the generations of the population. Lesser the population size slower will be the convergence to the solution and greater will be the number of generations.
Initially the conductor sizes for each of the feeder segments are selected randomly from the set $B$. The first element of the chromosome which represents the feeder section connected to the substation must be the integer representing the largest size of the set of conductor sizes considered. The sizes considered for the set $B$ should be of appropriate sizes that are used in the power industry as per the standards and pertaining to the class of distribution voltage and to the range of load levels in the feeder. When the size randomly selected for an infeed edge of a node changes by one step lower than that of its parent, a subset $B'$ of conductor types is considered for the rest of the downstream segments of the feeder. The largest size of the set $B'$ is updated to that of the new size selected. Let us consider the chromosome shown in fig-4.2 representing a solution to the feeder shown in fig-4.1. The code corresponding to the biggest size conductor in the set considered, ('Rabbit' represented by integer 3) is chosen for the Node1 and the same is maintained upto Node6. Random selection picks up the next lower size ('Weasel' represented by integer 2) at Node7. Hence for random selection at Node6, the set $B$ is updated to its subset $B'$ consisting of codes [1, 2]. The same procedure is repeated at every changeover of selected code. The initial population obtained during a typical run of the GA procedure for the feeder shown in fig-4.1 is given in Table-4.1.

4.3.5.2 Selection (Reproduction)

Reproduction operation was performed on the original population (referred as [OLD-POP]$_{p-m}$ in the GA algorithm) by Tournament selection method as discussed in section 3.3.5.2 of chapter-3. The selected individuals form a new population called as mating pool (referred as [POOL]$_{p-m}$ in the GA algorithm). The mating pool obtained after the reproduction operation on the initial population during a typical run of the GA procedure for the feeder shown in fig-4.1 is given in Table-4.2.

4.3.5.3 Crossover

Crossover is an operation by which the individuals of the mating pool exchange information. The individuals of the mating pool are transformed so as to explore new search spaces and bring in new structures into the population. New solutions are produced through crossover operation. At each instance of crossover operation, two chromosomes are picked up from the mating pool. They are the parent chromosomes. Generate a random
number to check whether the parents to mate or not. The parent chromosomes are permitted to mate if the cross-over probability $p_{cross}$ is greater than the random number generated. The parent couple is made to swap their bit string beyond a site called crossover site. The new ones are called the children chromosomes. Different crossover operators such as single-point crossover, multipoint crossover, uniform crossover using a mask etc are being used in the GA literature.

The proposed GA adopts single-point crossover due to its simplicity, less computation and also due to the generation of lesser amount of infeasible solutions. The crossover operation changes the conductor types of infeed edge of all the nodes in the mate-1 chromosome beyond the crossover site to assume those of the mate-2 chromosome beyond the same site and vice-versa. The crossover operation was performed by selecting a random number between 1 and ND to be the crossover site. Heuristics are included to result in feasible children chromosomes. i.e. the conductor size of mate-2 to the left of the crossover site should be greater than or equal to that of mate-1 at its site and vice-versa. Let us consider crossover operation between chromosomes Parent-1 and Parent-2 shown in fig-4.3(a) representing solutions of the feeder shown in fig-4.1. If the crossover site is 5, then the genes 6-13 are interchanged. The resulting child-2 is infeasible as the gradation is not in decreasing order as seen from fig-4.3(b), while swapping of genes 7-13 results in feasible chromosomes as shown in fig-4.3(c) for which case the crossover site chosen must be 6. It can be observed that the crossover operation illustrated has produced one lower cost child (chromosome better then the parents), even though a higher cost child is also produced.

In the case of feeders having multiple laterals, the crossover operation can be made following the same procedure illustrated above in fig-4.3. The parent chromosomes for crossover are picked up from the mating pool by Stochastic Remainder selection mechanism. The mates once selected for crossover are removed from the mating pool and the mates for the next generation crossover are chosen among the individuals remaining in the mating pool. The mating pool consists of more than one copy of the high fitness chromosomes. The operation has been performed at a fixed probability rate $p_{cross}$. The operation between the chosen mates is performed only if the outcome of a random event at the probability $p_{cross}$ turns out to be TRUE. Otherwise the mates are copied onto the
new population (referred as \([\text{INTERPOP}]_{p-m}\) in the GA algorithm) without being crossed. In this way all individuals in the mating pool are absorbed into the next population, either in their original form or as transformed ones. The rate is always chosen to be less than 1, in order to preserve some of the good strings that are already present in the mating pool. Best value of crossover probability for test feeder has been obtained through a graphical study. The population obtained after the crossover operation on the mating pool during a typical run of the GA procedure for the feeder shown in fig-4.1 is given in Table-4.3.

<table>
<thead>
<tr>
<th>Parent-1</th>
<th>Parent-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 3 3 3 3 3 2 2 2 1 1 1 1</td>
<td>3 3 3 3 2 2 2 2 2 1 1 1</td>
</tr>
<tr>
<td>955648.39</td>
<td>983857.02</td>
</tr>
</tbody>
</table>

(a) Parent chromosomes

<table>
<thead>
<tr>
<th>Child-1</th>
<th>Child-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 3 3 3 3</td>
<td>3 3 3 3 2</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

(b) Infeasible crossover operation

<table>
<thead>
<tr>
<th>Child-1</th>
<th>Child-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 3 3 3 3</td>
<td>3 3 3 3 2</td>
</tr>
<tr>
<td>947687.92</td>
<td>991817.49</td>
</tr>
</tbody>
</table>

(c) Feasible crossover operation

**Fig-4.3** Feasible and infeasible crossover operations

### 4.3.5.4 Mutation

A mutation operation changes (increments or decrements) the value of the randomly selected feeder section/node to a size next to its current size after verifying the feasibility so as not to disturb the ordering of the conductor sizes towards both the ends of the feeder. Considering the example chromosome shown in fig-4.2, the element \(c_5 = 3\) cannot be mutated to 2 while \(c_6\) can be mutated to 2. To state another example, \(c_{11} = 1\) cannot be mutated to 2 while \(c_{10}\) can be mutated to 2. Whenever a gene is to be mutated, the code representing a conductor size is changed to the one
bigger than or at the least equal to that of its child of maximum size and
- lower than or at the most equal to that of its parent.

Thus, though the mutation operation is random, it is limited by the above stated conditions so as to produce feasible chromosomes. The operation has been attempted on each gene of a chromosome at a fixed probability rate ‘\( p_{\text{mue}} \)’. The operation on a given gene is performed only if the outcome of a random event at the probability ‘\( p_{\text{mute}} \)’ turns out to be TRUE. Since the mutation operation is meant to introduce a rare local change in a configuration, generally a very small probability rate of about 0.01 to 0.1 is used. Best value of mutation probability for test feeder has been obtained through graphical study. The population resulting after mutation has been designated as \([\text{NEW-POP}]_{p_{\text{m}}} \) in the GA algorithm. The population obtained after the mutation operation at the first generation cycle of a typical run of the GA procedure for the feeder shown in fig-4.1 is given in Table-4.4.

4.3.5.5 Diversification Strategy

As discussed in section 3.3.5.5 of chapter-3, when the diversity of a population after certain number of generations drops to a low value which may be a premature convergence the population diversity is controlled according to a rate called as diversification rate. Diversity is increased by increasing the mutation rate. The control is actuated when the population diversity falls below a prespecified minimum value expressed in percentage.

4.3.5.6 Convergence

Convergence is achieved when the number of generations reaches the maximum number of generations specified and after the diversification is invoked over and above a pre-specified number of times. Due to the randomness of GA method the solution tends to differ for each run even with the same initial population. For this reason it is suggested to perform multiple runs and select the most acceptable solution. Elitist strategy retains the best solution in any population. This strategy replaces the best candidate of population at the beginning of a generation cycle into the worst one of the transformed population obtained after reproduction, crossover and mutation, so as carry the best solution of a generation to the next throughout the GA process. The final population obtained after 50 generations of a typical run of the GA procedure for the feeder shown in fig-4.1 is given in Table-4.5.
The reproduction, crossover and mutation operations [77, 80-83, 85-86] performed in order on a given population constitute a generation cycle of the GA process. Once a cycle of generation is completed, the population shifts from \([\text{OLD-POP}]_{p-m}\) to \([\text{NEW-POP}]_{p-m}\). After evaluation of the mutated population of a generation cycle, which is now in \([\text{NEW-POP}]_{p-m}\), it is copied on to the \([\text{OLD-POP}]_{p-m}\) for further processing through next generation cycle.

### 4.3.6 Fitness function

The score of each chromosome is its fitness value, which is the value assumed by a function called fitness function [77, 80-83, 85-86]. A chromosome can survive through the reproduction and the transformation operations of the GA process based on its fitness function values. Those chromosomes that give higher values for fitness function are copied in multiples into a new population designated as mating pool, which are further taken through crossover operation for exchange of information among the high fitness-chromosomes and then through mutation operation for introducing a fine and local changes in the chromosome as a bid towards turning the search into uncovered regions of the solution space. A fitness function may directly or indirectly represent the objectives of problem. As given by equation (2.21), the objectives of the conductor gradation problem have been formulated as a cost function which needs to be minimized. Since the trend of GA approach is to maximize a function, a corresponding maximization function \(F\) defined by equation (3.2) is used. Each chromosome of the population resulting after mutation \([\text{NEW-POP}]_{p-m}\), is decoded to obtain their values for the fitness function.

### 4.3.7 GA Algorithm

**Definition of GA Parameters and Variables**

- **POPSIZE** - population size
- **pcross** - crossover probability
- **pmute1, pmute2** - mutation probabilities, \(pmute2 > pmute1\)
- **DIV-MIN** - percentage of minimum diversity in the population, when the mutation rate is to be increased to diversify the population.
- **MAX-GEN** - maximum prespecified number of generations before the procedure is stopped
LCHROM - length of the chromosome equal to the number of nodes in the feeder, excluding substation node.

NDIV-MAX - maximum prespecified number of times the diversification can be applied to populations before the procedure is stopped.

BIGM - a large enough constant value for transforming the minimization problem into a maximization problem

igen - generation counter

iDiv - count for invoking diversification into the population

[OLD-POP]_{p,m} - population at the beginning of a generation, a two dimensional vector of size p = 1 to POPSIZE, m = 1 to LCHROM

[NEW_POP]_{p,m} - population at the end of a generation, a two dimensional vector of size p = 1 to POPSIZE, m = 1 to LCHROM

[OLD_FIT]_{p} - vector of fitness values of [OLD_POP]_{p,m}, of size p = 1 to POPSIZE

[NEW_FIT]_{p} - vector of fitness values of [NEW_POP]_{p,m}, of size p = 1 to POPSIZE

[BEST_SOL]_{m} - chromosome that gives maximum fitness function value. It is an array of size m = 1 to LCHROM

BEST_FIT - maximum fitness function value obtained thus far in the GA process

MIN_COST - objective function value (i.e. cost) of the [BEST_SOL]_{m}

iBest - pointer to maximum fitness chromosome of [OLD_POP]_{p,m} vector of a generation

iWorst - pointer to minimum fitness chromosome of [NEW_POP]_{p,m} vector of a generation

[POOL]_{p,m} - population after reproduction operation, a two dimensional vector of size p = 1 to POPSIZE, m = 1 to LCHROM

[INTER_POP]_{p,m} - population after crossover operation, a two dimensional vector of size p = 1 to POPSIZE, m = 1 to LCHROM

Step 1: Initialize the GA parameters, static data and cost constants of the system.

Step 2: Read nodes and edges data and construct the network:
   Initialize each node
with load data
- with its infeed edge data, its parent and children nodes

Step 3: Read conductor data - size-id, area of cross section, cost, reactance per km etc.

Step 4: Generate initial population into \([OLD\_POP]_{p-m}\)

Step 5: Decode the population \([OLD\_POP]_{p-m}\) and obtain \([OLD\_FIT]_p\).
For each chromosome, i.e., for \(p = 1\) to POPSIZE
- get the decision variables \(b_i\), \(i = 1\) to \(m\).
- register the size of feeder segments coded into each of them into the data modules of the respective nodes.
- compute investment cost and loss cost of lines and feeder voltage drop
- compute objective function value and fitness function values using equations (2.21) and (3.2)

Step 6: Compute the statistics:
- Determine average fitness, maximum fitness and minimum fitness of \([OLD\_FIT]_p\), \(p = 1\) to POPSIZE
- \(iBest \leftarrow \) pointer \(p\) to the maximum fitness chromosome of \([OLD\_POP]_{p-m}\)
- Display the statistics & results of initial population.

Step 7: Initialize the best solution
\([BEST\_SOL]_m \leftarrow [OLD\_POP]_{iBest-m}, iBest \in POPSIZE, m = 1\) to LCHROM.
BEST_FIT \(\leftarrow [OLD\_FIT]_{iBest}, \ iBest \in POPSIZE.\)

Step 8: Initialize \(igen \leftarrow 1\)

Step 9: Determine Diversity of \([OLD\_POP]_{p-m}\)
IF Diversity < DIV_MIN, THEN "pmute" \(\leftarrow pmut2\)
ELSE "pmute" \(\leftarrow pmut1\)

Step 10: Perform reproduction on the \([OLD\_POP]_{p-m}\) using Tournament selection method and form the mating pool \([POOL]_{p-m}\).

Step 11: Perform crossover on the mating pool at the rate "pcross" and form the intermediate population \([INTER\_POP]_{p-m}\).

Step 12: Perform mutation on the intermediate population at the rate "pmute" and form \([NEW\_POP]_{p-m}\)

Step 13: Decode the population \([NEW\_POP]_{p-m}\) and obtain \([NEW\_FIT]_p\).
For each chromosome, i.e., for \(p = 1\) to POPSIZE
- get the decision variables \( b_i, i = 1 \) to \( m \).
- register the size of feeder segments coded into each of them into the data modules of the respective nodes.
- compute investment cost and loss cost of lines and feeder voltage drop
- compute objective function value and fitness function values using equations (2.21) and (3.2) respectively

**Step 14:** Compute the statistics:
- Determine average fitness, maximum fitness and minimum fitness of \( \{OLD\_FIT\}_p, p=1 \) to \( POP\_SIZE \)
- \( iWorst \) \( \leftarrow \) pointer ‘\( p \)’ to the minimum value of fitness vector \( \{NEW\_FIT\}_p \).
  Also points to the \( p \)th chromosome of \( \{NEW\_POP\}_{p-m} \)
- Display the results and statistics of \( \{NEW\_POP\}_{p-m} \)

**Step 15:** Update \( \{NEW\_POP\} \) with elitist strategy;

\[
\begin{align*}
\{NEW\_POP\}_{Worst-m} & \leftarrow \{OLD\_POP\}_{Best-m} \\
\{NEW\_FIT\}_{Worst} & \leftarrow \{OLD\_FIT\}_{Best}
\end{align*}
\]

Update \( iBest; \)

\( iBest; \) \( \leftarrow \) pointer ‘\( p \)’ to the maximum value of fitness vector \( \{NEW\_FIT\}_p \).
  Also points to the \( p \)th chromosome of \( \{NEW\_POP\}_{p-m} \)

**Step 16:** Update and display the best solution.

If \( \{NEW\_FIT\}_{iBest} > BEST\_FIT \), Then

\[
\begin{align*}
\{BEST\_SOL\}_m & \leftarrow \{NEW\_POP\}_{Best-m}, iBest \in POP\_SIZE. \\
BEST\_FIT & \leftarrow \{NEW\_FIT\}_{iBest}, iBest \in POP\_SIZE.
\end{align*}
\]

**Step 17:** \( igen \leftarrow igen + 1 \)

**IF** \( igen > MAX\_GEN \) go to step-19.
**ELSE** go to next step.

**Step 18:** copy

\[
\begin{align*}
\{NEW\_POP\}_{p-m} & \leftarrow \{OLD\_POP\}_{p-m} \\
\{NEW\_FIT\}_p & \leftarrow \{OLD\_FIT\}_p
\end{align*}
\]

And go to step-9.

**Step 19:** Display the results of the optimum solution:

Optimum solution \( \{BEST\_SOL\}_m \),

\( MIN\_COST = BIGM - BEST\_FIT \), and

Voltage drop of optimally graded feeder.

**Step 20:** STOP
Flow chart of the GA Algorithm

START

Initialize GA parameters: POPSIZE, pcross, pmut, MAX_GEN etc.
Initialize system parameters: ND, DIV_MIN, NDIV_MAX, MAX_NEIGH etc.

Read the feeder-conductor data. (area of c.s., cost, parameters etc.)

Read node data (load, infeed edge, parent and children nodes). Initialize the nodes and construct the feeder.

LCHROM = ND

Generate initial population [OLD_POP]_{p,m}

Decode each chromosome of [OLD_POP]_{p,m} to realize the size of feeder lines. Compute the fitness function values, [OLD_FIT]_{p} of each of them.

1
Statistics: Determine the average fitness, maximum fitness, minimum fitness of \([\text{OLD\_POP}]_{p-m}\).

\[ iBest \leftarrow \text{pointer 'p' to maximum fitness chromosome of } [\text{OLD\_POP}]_{p-m} \]

Initialize the best solution.

\[ \text{BEST\_FIT} \leftarrow [\text{OLD\_FIT}]_{iBest} \]
\[ [\text{BEST\_SOL}]_m \leftarrow [\text{OLD\_POP}]_{iBest-m} \]

Display the statistics and results of initial population.

\[ iGen \leftarrow 1 \]

A

\[ \text{DIV\_MIN} < 20\% \]

\[ pmute \leftarrow pmute2 \]

\[ pmute \leftarrow pmute1 \]

Perform Reproduction on \([\text{OLD\_POP}]_{p-m}\) and form \([\text{POOL}]_{p-m}\).

Perform Crossover at the rate 'pcross' on \([\text{POOL}]_{p-m}\) and form \([\text{INTER\_POP}]_{p-m}\).
Perform Mutation at the rate 'pmute' on [INTER_POP]_{p,m} and form [NEW_POP]_{p,m}.

Decode each chromosome of [NEW_POP]_{p,m} to realize the size of feeder lines. Compute fitness function values [NEW_FIT]_{p} of each of them.

Statistics: Determine the average fitness, maximum fitness, minimum fitness of [NEW_POP]_{p,m}.

\( iWorst \leftarrow \text{pointer 'p' to minimum fitness chromosome of [NEW_POP]_{p,m}}. \)

Elitist strategy:
\[
\begin{align*}
[NEW\_POP]_{Worst-m} &\leftarrow [OLD\_POP]_{Best-m} \\
[NEW\_FIT]_{Worst} &\leftarrow [OLD\_FIT]_{Best}
\end{align*}
\]

\( iBest \leftarrow \text{pointer 'p' to maximum fitness chromosome of [NEW_POP]_{p,m}}. \)

\[ [NEW\_FIT]_{Best} > BEST\_FIT \]

NO

YES

Update the best solution.
\[
\begin{align*}
BEST\_FIT &\leftarrow [NEW\_FIT]_{Best} \\
[BEST\_SOL]_{m} &\leftarrow [NEW\_POP]_{Best-m}
\end{align*}
\]
Decode \([\text{BEST\_SOL}]_m\).
Compute objective function value, fitness function value and feeder voltage drop of the \([\text{BEST\_SOL}]_m\).

Display the results: Optimal feeder conductor sizes \([\text{BEST\_SOL}]_m\),
Optimum cost \(\text{MIN\_COST} = \text{BIGM} - \text{BEST\_FIT}\),
and feeder voltage drop

STOP
4.4 APPLICATION

The test system for the Conductor Gradation Problem is a 11kV primary distribution feeder with 13 consumer points taken from Tiruchirappalli district, Tamil Nadu. The feeder is shown in fig-4.1. Length of the feeder is 21.89 km and is operated with a total peak load of 1070.9 kW at a power factor of 0.8. All the feeder segments are of Rabbit conductors. Three different conductor types namely ‘Rabbit, Weasel, Squirrel’ have been considered (see Table-b.1 of Appendix-B for conductor data). Power loss of the feeder before grading was found to be 49.85kW and the energy loss over a planning period of 15 years will be 8443.07kWh. Voltage drop along the feeder has been 7.31%. Other constants specific to the problem are taken from the works reported in [38] and [40].

GA based solution procedure was employed with a population size of 20 and with a crossover and mutation probability of 0.6 and 0.01 respectively. The length of the solution string was equal to the size of the feeder i.e. 13 in this case. The first gene of a chromosome is forced to be the biggest size, since a judicious choice of the conductor sizes have been made for consideration. For this problem an initial population of size 20 was generated as given in Table-4.1 and the average cost of the population is Rs.1133199.41. The mating pool obtained using Tournament selection mechanism with subpopulation size of 2 is given in Table-4.2 and the average cost of the population is Rs.979981.71. Table-4.3 shows the population obtained after cross-over (single point) operation and the average cost of the population is Rs.978935.16. Table-4.4 shows the population obtained after mutation operation and the average cost of the population is Rs.983314.47. The final population obtained after 50 generations is given in Table-3.5 whose average cost is Rs.928523.84. Cost of the optimal solution is Rs.925530.95. The average cost of the populations depicts improvements in solution. During the run the diversification procedure was found to have been invoked for 21 times (generations). The genes chosen and mutated at random are shown underlined in Table-4.3 and Table-4.4.
Table -4.1
INITIAL POPULATION

<table>
<thead>
<tr>
<th>S.No</th>
<th>Chromosome</th>
<th>Cost (Rs)</th>
<th>S.No</th>
<th>Chromosome</th>
<th>Cost (Rs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>333222 2222222</td>
<td>1033088.24</td>
<td>11</td>
<td>333333 1111111</td>
<td>1008073.01</td>
</tr>
<tr>
<td>2</td>
<td>322111 1111111</td>
<td>1315693.00</td>
<td>12</td>
<td>321111 1111111</td>
<td>1454021.83</td>
</tr>
<tr>
<td>3</td>
<td>333322 2221111</td>
<td>991817.49</td>
<td>13</td>
<td>333332 2211111</td>
<td>971054.45</td>
</tr>
<tr>
<td>4</td>
<td>332211 1111111</td>
<td>1221482.85</td>
<td>14</td>
<td>332221 1111111</td>
<td>1185127.75</td>
</tr>
<tr>
<td>5</td>
<td>333333 3333333</td>
<td>1013520.73</td>
<td>15</td>
<td>322221 1111111</td>
<td>1247499.20</td>
</tr>
<tr>
<td>6</td>
<td>322211 1111111</td>
<td>1283854.31</td>
<td>16</td>
<td>322222 1111111</td>
<td>1224318.65</td>
</tr>
<tr>
<td>7</td>
<td>333333 3321111</td>
<td>933788.81*</td>
<td>17</td>
<td>311111 1111111</td>
<td>1540037.23</td>
</tr>
<tr>
<td>8</td>
<td>333333 3333322</td>
<td>971154.16</td>
<td>18</td>
<td>333333 2222222</td>
<td>975482.73</td>
</tr>
<tr>
<td>9</td>
<td>322222 2222221</td>
<td>1177386.91</td>
<td>19</td>
<td>333333 3333321</td>
<td>956812.71</td>
</tr>
<tr>
<td>10</td>
<td>333333 3333332</td>
<td>984567.27</td>
<td>20</td>
<td>322222 2211111</td>
<td>1175206.90</td>
</tr>
</tbody>
</table>

* Best solution

Table -4.2
POPULATION AFTER REPRODUCTION: MATING POOL

<table>
<thead>
<tr>
<th>S.No</th>
<th>Chromosome</th>
<th>Cost (Rs)</th>
<th>S.No</th>
<th>Chromosome</th>
<th>Cost (Rs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>333333 3333332</td>
<td>984567.27</td>
<td>11</td>
<td>333333 3333321</td>
<td>956812.71</td>
</tr>
<tr>
<td>2</td>
<td>333333 3333321</td>
<td>956812.71</td>
<td>12</td>
<td>333333 3333321</td>
<td>956812.71</td>
</tr>
<tr>
<td>3</td>
<td>333333 3333322</td>
<td>971154.16</td>
<td>13</td>
<td>322222 2211111</td>
<td>1175206.90</td>
</tr>
<tr>
<td>4</td>
<td>333333 3321111</td>
<td>933788.81*</td>
<td>14</td>
<td>333333 3321111</td>
<td>933788.81*</td>
</tr>
<tr>
<td>5</td>
<td>333333 2222222</td>
<td>975482.73</td>
<td>15</td>
<td>333333 3333321</td>
<td>956812.71</td>
</tr>
<tr>
<td>6</td>
<td>333333 1111111</td>
<td>1008073.01</td>
<td>16</td>
<td>333333 3321111</td>
<td>933788.81*</td>
</tr>
<tr>
<td>7</td>
<td>333333 3321111</td>
<td>933788.81*</td>
<td>17</td>
<td>333332 2211111</td>
<td>971054.45</td>
</tr>
<tr>
<td>8</td>
<td>333322 2221111</td>
<td>991817.49</td>
<td>18</td>
<td>333333 3333333</td>
<td>1013520.73</td>
</tr>
<tr>
<td>9</td>
<td>333332 2211111</td>
<td>971054.45</td>
<td>19</td>
<td>333333 3333322</td>
<td>971154.16</td>
</tr>
<tr>
<td>10</td>
<td>333322 2222222</td>
<td>1033088.24</td>
<td>20</td>
<td>333332 2211111</td>
<td>971054.45</td>
</tr>
</tbody>
</table>

* Best solution
### Table -4.3
POPULATION AFTER CROSSOVER

<table>
<thead>
<tr>
<th>S.No</th>
<th>Chromosome</th>
<th>Cost (Rs)</th>
<th>S.No</th>
<th>Chromosome</th>
<th>Cost (Rs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>333332 2211111</td>
<td>971054.45</td>
<td>11</td>
<td>333333 3321111</td>
<td>933788.81*</td>
</tr>
<tr>
<td>2</td>
<td>333333 1111111</td>
<td>1008073.01</td>
<td>12</td>
<td>333333 3333321</td>
<td>956812.71</td>
</tr>
<tr>
<td>3</td>
<td>333333 3333322</td>
<td>971154.16</td>
<td>13</td>
<td>322222 2211111</td>
<td>1175206.90</td>
</tr>
<tr>
<td>4</td>
<td>333333 3321111</td>
<td>933788.81*</td>
<td>14</td>
<td>333333 3333322</td>
<td>971154.16</td>
</tr>
<tr>
<td>5</td>
<td>333333 3333322</td>
<td>984567.27</td>
<td>15</td>
<td>333333 3333321</td>
<td>956812.71</td>
</tr>
<tr>
<td>6</td>
<td>333333 3333333</td>
<td>1013520.73</td>
<td>16</td>
<td>333333 3321111</td>
<td>933788.81*</td>
</tr>
<tr>
<td>7</td>
<td>333333 3333321</td>
<td>956812.71</td>
<td>17</td>
<td>333322 2222222</td>
<td>1011651.83</td>
</tr>
<tr>
<td>8</td>
<td>333322 2221111</td>
<td>991817.49</td>
<td>18</td>
<td>333322 2221111</td>
<td>1013253.91</td>
</tr>
<tr>
<td>9</td>
<td>333333 3311111</td>
<td>937101.68</td>
<td>19</td>
<td>333333 3321111</td>
<td>933788.81*</td>
</tr>
<tr>
<td>10</td>
<td>333333 2221111</td>
<td>967741.58</td>
<td>20</td>
<td>333333 3333321</td>
<td>956812.71</td>
</tr>
</tbody>
</table>

* Best solution
Genes chosen for mutation are shown underlined.

### Table -4.4
POPULATION AFTER MUTATION

<table>
<thead>
<tr>
<th>S.No</th>
<th>Chromosome</th>
<th>Cost (Rs)</th>
<th>S.No</th>
<th>Chromosome</th>
<th>Cost (Rs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>333332 2211111</td>
<td>971054.45</td>
<td>11</td>
<td>333333 3321111</td>
<td>933788.81</td>
</tr>
<tr>
<td>2</td>
<td>333332 1111111</td>
<td>1020166.19</td>
<td>12</td>
<td>333333 3333321</td>
<td>956812.71</td>
</tr>
<tr>
<td>3</td>
<td>333333 3333322</td>
<td>971154.16</td>
<td>13</td>
<td>322222 2211111</td>
<td>1193724.88</td>
</tr>
<tr>
<td>4</td>
<td>333333 3321111</td>
<td>933788.81</td>
<td>14</td>
<td>333333 3333322</td>
<td>971154.16</td>
</tr>
<tr>
<td>5</td>
<td>333333 3333333</td>
<td>984567.27</td>
<td>15</td>
<td>333333 3333321</td>
<td>956812.71</td>
</tr>
<tr>
<td>6</td>
<td>333333 3333333</td>
<td>1013520.73</td>
<td>16</td>
<td>333333 3321111</td>
<td>933491.43*</td>
</tr>
<tr>
<td>7</td>
<td>333333 3333322</td>
<td>940118.41</td>
<td>17</td>
<td>333333 333322</td>
<td>997310.38</td>
</tr>
<tr>
<td>8</td>
<td>333322 2221111</td>
<td>991817.49</td>
<td>18</td>
<td>333322 2221111</td>
<td>1009522.57</td>
</tr>
<tr>
<td>9</td>
<td>333333 3311111</td>
<td>937101.68</td>
<td>19</td>
<td>333333 3321111</td>
<td>933788.81</td>
</tr>
<tr>
<td>10</td>
<td>333332 2221111</td>
<td>959781.10</td>
<td>20</td>
<td>333333 3333321</td>
<td>956812.71</td>
</tr>
</tbody>
</table>

* Best solution
Genes mutated are shown underlined.
Table -4.5

<table>
<thead>
<tr>
<th>S.No</th>
<th>Chromosome</th>
<th>Cost (Rs)</th>
<th>S.No</th>
<th>Chromosome</th>
<th>Cost (Rs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>333333 3321111</td>
<td>933788.81</td>
<td>11</td>
<td>333333 3332111</td>
<td>925530.95*</td>
</tr>
<tr>
<td>2</td>
<td>333333 3332111</td>
<td>925530.95*</td>
<td>12</td>
<td>333333 3332111</td>
<td>925530.95*</td>
</tr>
<tr>
<td>3</td>
<td>333333 3322111</td>
<td>925828.33</td>
<td>13</td>
<td>333333 3332111</td>
<td>925530.95*</td>
</tr>
<tr>
<td>4</td>
<td>333333 3332111</td>
<td>925530.95*</td>
<td>14</td>
<td>333333 3222221</td>
<td>946486.23</td>
</tr>
<tr>
<td>5</td>
<td>333333 3332111</td>
<td>925530.95*</td>
<td>15</td>
<td>333333 3332111</td>
<td>925530.95*</td>
</tr>
<tr>
<td>6</td>
<td>333333 3222111</td>
<td>93032.87</td>
<td>16</td>
<td>333333 3331111</td>
<td>933491.43</td>
</tr>
<tr>
<td>7</td>
<td>333333 3332111</td>
<td>925530.95*</td>
<td>17</td>
<td>333333 3322221</td>
<td>939281.69</td>
</tr>
<tr>
<td>8</td>
<td>333333 3331111</td>
<td>926665.06</td>
<td>18</td>
<td>333333 3332111</td>
<td>925530.95*</td>
</tr>
<tr>
<td>9</td>
<td>333333 3332111</td>
<td>925530.95*</td>
<td>19</td>
<td>333333 3332111</td>
<td>925530.95*</td>
</tr>
<tr>
<td>10</td>
<td>333333 3332111</td>
<td>925530.95*</td>
<td>20</td>
<td>333333 3332111</td>
<td>925530.95*</td>
</tr>
</tbody>
</table>

* Best solution

To illustrate the selection of best GA parameter values, the algorithm was run 10 times with crossover rates fixed at 0.6 and 0.8 for the following values of mutation rates: 0.001, 0.005, 0.02 and 0.08. Fig-4.4 to fig-4.5 illustrate the graph of the average costs over 10 ten runs during the execution of the algorithm which produced consistent results for the mutation rates: 0.005 and 0.02. As shown in fig-4.5 crossover probability of 0.6 and mutation probability of 0.02 gave lowest cost and hence are chosen to be the best parameter values for the feeder. The feeder of fig-4.1 with optimally graded conductor sizes is shown in fig-4.6.
Fig-4.4 Effect of Crossover probability on Convergence

Fig-4.5 Effect of Crossover probability on Convergence
4.5 CONCLUSION

The proposed GA procedure for solving the problem of grading of feeder is found to be effective with the use of efficient coding method and specifically manipulated crossover and mutation operators. Optimal solution with a population size of 20 has been obtained in about 140 generations without the use of diversification strategy. Use of diversification strategy produced the same results in 40-50 generations. With a population size equal to 50 the same optimum has been obtained in less than 10 generations.

Grading of conductors had produced a maximum reduction in the system cost of 8.9%. Power loss of the feeder after grading increased to 51.23kW (an increase of 2.8% with respect to the loss before grading) and the energy loss over a planning period of 15 years increased to 8676.01 kWh. Voltage drop along the feeder after conductor gradation has increased to 7.91% from 7.31% but has resulted in a cost saving of Rs.87,989.78.