CHAPTER - 3

RESEARCH METHODOLOGY
3.1 INTRODUCTION

In this chapter, the information about the research methodology adopted in the present work is described in context of the problem undertaken related to offset printing process. Research Methodology used is based on the concepts namely Design of Experiment and Analytical Hierarchy Process, Genetic Algorithm and Computational Fluid Dynamics.

1. **Design of experiments (DOE)**

   The word “design” in the expression design of experiments, is used in a general sense to convey a mental project or a scheme in which means to an end are laid down. To design the experiment is to develop a scheme or layout of the different conditions to be studied. The Taguchi design of experiments can be used to optimize many designs. An experiment design must satisfy two objectives. First, the number of trials must be determined. Second, the conditions for each trial must be specified. Taguchi’s arrays are versatile recipes that apply to several experimental conditions[14].

   Outcome of design of experiment needs further clarification and validation so as to implement the same for practical application. Analytical Hierarchy Process is used for this purpose.

2. **Analytical Hierarchy Process (AHP)**

   Analytical hierarchy process (AHP) approach is a simple decision-aiding tool with a simple hierarchy decision structure for Prioritizing alternatives having multi objectives. It is based on three principles[15,16]:
   
   a) Decomposition of the problem
   
   b) Comparison of alternative elements
   
   c) Synthesis of priorities

3. **Genetic Algorithm**

   A genetic algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics.
3.2 DESIGN OF EXPERIMENT

Before designing an experiment, knowledge of the product/process under investigation is of prime importance for identifying the factors likely to influence the outcome. In order to compile a comprehensive list of the factors, the input to the experiment is generally obtained from all the people involved in the project. Taguchi found brainstorming to be necessary steps for determining the full range of factors to be investigated.

3.2.1 Taguchi Philosophy

Taguchi exposed an excellent philosophy for quality control in the manufacturing industries. Indeed, his doctrine is creating an entirely different breed of engineers who think, breathe and live quality. The Whole of the technology and techniques arise entirely out of these three ideas. These concepts are:

1. Quality should be designed in to the product and not inspected into it.
2. Quality is best achieved by minimizing the deviation from a target. The product should be so designed that it is immune to uncontrollable environmental factors.
3. The cost of quality should be measured as a function of deviation from the standard and the losses should be measured system-wide.

Taguchi built upon W.E. Deming’s observation that 85% of poor quality is attributable to the manufacturing process and only 15% to the worker. Hence he developed manufacturing system that was “robust” or insensitive to daily and seasonal variations of environment, machine wear, and other external factors.

Taguchi believed that the better way to improve quality was to design and build it into the product. Quality improvement starts at the very beginning i.e. during the design stages of a product or a process, and continues through the production phase. He proposed an “off-line” strategy for developing quality improvement in place of an attempt to inspect quality into a product on the production line. He observed that poor quality cannot be improved by the process of inspection. No amount of inspection can put quality back into the product; it merely treats a symptom. Therefore, quality concepts should be based upon, and developed around the philosophy of prevention. The product design must be so robust that it is immune to the influence of uncontrolled environmental factors on the manufacturing processes. He emphasizes that quality is what one designs into a product.

A second objective of manufacturing products to conform to an ideal value is to reduce the variation or scatter around the target. To accomplish this objective, Dr. Taguchi cleverly incorporates a unique way to treat noise factors. Noise factors, according to his
terminology, are factors which influence the response of a process, but cannot be economically controlled. The noise factors such as weather conditions, machinery wear, etc. are usually prime sources for variations. Through the use of what he calls the “Orthogonal arrays”, Taguchi devised an effective way to study their influence with the least number of repetitions. The end result is a “robust” design affected minimally by noise, i.e. with a high signal to noise value.

To achieve desirable product quality by design, Dr. Taguchi recommends a three stage process.

1. **Systems design**
2. **Parameter design**
3. **Tolerance design**

   The focus of the system design phase is on determining the suitable working levels of design factors. It includes designing and testing a system based on the engineer’s judgment of selected materials, parts, and nominal product process parameters based on current technology. Most often it involves innovation and knowledge from the applicable fields of science and technology. While system design helps to identify the working levels of the design factors, parameter design seeks to determine the factor levels that produce the best performance of the product process under study. The optimum condition is selected so that the influence of the uncontrolled factors (noise factors) causes minimum variation of system performance. Tolerance design is a step used to fine tune the results of parameter design by tightening the tolerance of factors with significant influence on the product.

Taguchi’s approach to enhance quality in the design phase involves two steps;

1. Optimizing the design of the product/process (system approach).
2. Making the design insensitive to the influence of uncontrollable factors (robustness).

   The Taguchi technique is applied in four steps.

   1. Brainstorm the quality characteristics and design parameters important to the product/process.
   2. Design and conduct the experiments.
   3. Analysis the results to determine the optimum conditions.
   4. Run a confirmatory test(s) using the optimum conditions.

### 3.2.2 Quality Characteristics

Every product is designed to perform some intended function. Some measurable characteristic, generally referred to as the quality characteristic, is used to express how well a product performs the
function. No matter how the quality of the product is measured, by a single criterion, or by a combination of multiple criteria, the measure will possess one of the following three characteristics:

- **The bigger the better** - e.g. maximum expected life of a component.
- **The smaller the better** - e.g. minimum shrinkage in a cast iron cylinder block casting.
- **The nominal the better** - e.g. dimension of a part consistently achieved with modest variance.

**Key Observations**

- In designing experiments with interactions, Triangular Tables or Linear Graphs should be used for column assignments. To select the appropriate Orthogonal Array (OA), the types of interactions and their degrees of freedom will have to be considered.
- For the purpose of analysis, interactions are treated as are any other factors; however their presence is ignored for the preliminary determination of the optimum condition. The relative significance of interaction is obtained from an ANOVA study.

**3.2.3 Orthogonal Array (OA)**

Taguchi method is designed to improve the quality of products and processes where the performance depends on many factors. In laying out a test and development strategy, simple logic will usually be sufficient to establish all possible combinations of factors along with allowable ranges of each of the factors involved. But for engineering projects involving many factors, the number of possible combinations is prohibitively large. In addition higher order interactions among the influencing factors may be needed for specific projects. A customary method of reducing the number of test combinations is to use what are known as partial factorial experiments like for five parameters and four levels the $L_{16}$ - Orthogonal Array is given as shown in Table 3.1.

The OAs provides a recipe for fractional factorial experiments which satisfy a number of situations. When a fixed number of levels for all factors is involved and the interactions are unimportant, standard OAs will satisfy most experimental design needs. A modification of the OAs becomes necessary when mixed levels and interactions are present. Simple designs with smaller number of factors, at fixed levels, will be discussed first.
Table 3.1 \( L_{16} \) - Orthogonal Array

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

3.2.4 Signal to Noise (S/N) Ratio

The terms factors, variables and parameters are synonymously used to refer to factors which influence the outcome of the product or process under investigation. Taguchi further categorized the factors as controllable factors and noise factors. The goal is robust optimum which is influenced minimally by these less controllable variables; the study of the impact of noise factors on the optimum parameters is desirable.

**Controllable Factors:** Factors whose levels can be specified and controllable during the experiment and in the final design of the product or process.

**Noise Factors:** These are factors which have influence on the product or process results, but generally are not maintained at specific levels during the production process or application period.

**Inner Array:** The OA of the controllable factors.
Outer Array: - The OA of recognized noise factors.

Experiment: - The experiment refers to the whole experimental process.

Trial condition: - The combination of factors/levels at which a trial run is conducted.

Condition of Experiment: - Unique combinations of factor levels described by the OA.

Repetitions or Runs: - These define the number of observations under the same conditions of an experiment. The experiment requires a minimum of one run per condition of the experiment. But one run does not represent the range of possible variability in the results. Repetition or runs enhances the available information in the data. Taguchi suggests guidelines for repetitions.

\[ S/N = -10 \log_{10}(MSD) \]

Where, MSD = Mean squared deviation from the target value of the quality characteristic.

Use of the S/N ratio of the results, instead of the average values, introduces some minor changes in the analysis.

- Degree of freedom of the entire experiment is reduced. The S/N ratio calculation is based on data from all observations of a trial condition. The set of S/N ratios can then be considered as trial results without repetitions. Hence the DOF, in case of S/N is the number of trials – 1. The rest of the analysis follows the standard procedure.

- S/N must be converted back to meaningful terms. When the S/N ratio is used, the results of the analysis, such as estimated performance from the main effects or confidence interval are expressed in terms of S/N. To express the analysis in terms of the experimental result, the ratio must be converted back to the original units of measurement.

3.2.5 Analysis of Variance (ANOVA)

Analysis provides the variance of controllable and noise factors. By understanding the source and magnitude of variance, robust operating conditions can be predicted. This is a second benefit of the methodology.

- **Total Number of Trials:** - The total number of trials is the sum of the number of trials at each level.

- **Degrees of Freedom (DOF):** - It is a measure of the amount of information that can be uniquely determined from a given set of data.

- **Sum of Squares:** - The sum of squares is a measure of the deviation of the experimental data from the mean value of the data. Summing each squared deviation emphasizes the total deviation.
3.3 ANALYTICAL HIERARCHY PROCESS

The crux of AHP is to enable a decision maker to structure multi attribute decision analysis (MADA) problem visually in the form of an attribute hierarchy [15].

3.3.1 Stages of Analytical Hierarchy Process

The general approach of AHP consists of four stages.

1) Determination of the overall objective, attributes and competing alternatives with simple hierarchy structure.

2) Determination of the self weight or the self-importance of the attributes.

3) Determination of the relative importance of each of the alternative with respect to each attribute.

4) Overall priority weight determination of each of this alternative.

STAGE 1 Determination of the overall objective, attributes and competing alternatives with simple hierarchy structure.

Develop a hierarchical structure with objective at top level, the attributes at the second level and the alternative at the end level.

![Simple Hierarchy Structure](image)

**Fig 3.1 Simple Hierarchy Structure**

STAGE 2

Determination of the self weight (the self-importance) of the attributes
a) Construct the pair-wise matrix for the attributes pairs by using Table provided by J.J. Satty[15].

**Table 3.2 the fundamental scale for pair-wise comparison**

<table>
<thead>
<tr>
<th>Intensities</th>
<th>Definition</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Equal importance</td>
<td>Two activities contribute equally to the objective</td>
</tr>
<tr>
<td>3</td>
<td>Weak importance (of one over the other)</td>
<td>The judgment is to favor one activity over another but not conclusive</td>
</tr>
<tr>
<td>5</td>
<td>Strong importance</td>
<td>The judgment is strongly in favors one activity over another</td>
</tr>
<tr>
<td>7</td>
<td>Demonstrated importance over the other</td>
<td>Conclusive judgment as to the importance of one activity over another.</td>
</tr>
<tr>
<td>9</td>
<td>Absolute importance</td>
<td>The judgment in the favor one activity over another is of highest possible order of affirmation</td>
</tr>
<tr>
<td>2,4,6,8</td>
<td>Intermediate value between two adjacent judgment</td>
<td>When compromise is needed</td>
</tr>
</tbody>
</table>

Assuming a factors, the pair wise comparison of factors i with j yields a square matrix $A$ of $n \times n$ where $r_{ij}$ denotes the comparative importance of factor i with respect to factor j. In the matrix

$$A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \ldots & r_{1n} \\ r_{21} & r_{22} & r_{23} & \ldots & r_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & r_{n3} & \ldots & r_{nn} \end{bmatrix}$$

When $i=j$, $r_{ii}=1$ and $r_{ij}=1/r_{ij}$.

b) Find the relative normalized weight ($W_i$) of each attribute such that

$$\sum_{i=1}^{n} W_i = 1$$
\[ W_i = \frac{GM_i}{\sum_{j=1}^{n} GM_i} = [w_1 \ w_2 \ w_3 \ldots \ w_N]^1 \]

Where \( i = 1 \) and \( j = 1 \)

\[ GM_i = \prod_{j} r_{ij} \]

c) Consistency check:

The AHP allows inconsistency, but provides a measure of the inconsistency in each set of judgments. This measure is an important by-product of the process of deriving priorities based on pair-wise comparison. The measure used for consistency check is consistency ratio (CR)

Causes of inconsistency:

1) lack of information
2) Inadequate model structure.

Procedure for consistency check is as given below.

i) Matrix multiplication of original matrix \( A \) and weighing factor matrix \( W_i \) to obtain new matrix \( N = (A \times W_i) \).

ii) Divide matrix \( N \) by matrix \( W_i \) to obtain eigenvalue matrix \( \lambda \).

iii) Now calculate consistency index (CI)

\[ CI = \lambda_{max} - \frac{n}{(n-1)} \]

Where \( n = \) order of matrix

iv) Obtain the random index (RI) from random indices Table as given by Satty.

<table>
<thead>
<tr>
<th>N</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>RI</td>
<td>0</td>
<td>0</td>
<td>0.58</td>
<td>0.90</td>
<td>1.12</td>
<td>1.24</td>
<td>1.32</td>
<td>1.41</td>
</tr>
</tbody>
</table>

v) Consistency ratio (CR) is given by

\[ CR = \frac{CI}{RI} \]
It is important to have a low CR does not become the goal of the decision making process. A low CR is necessary but not sufficient for a good decision. **CR of 0.1 or less than 0.1 is generally to be accepted. It means that pair-wise comparison made is proper.** If the value of CR is more than 0.10, then the pair-wise comparison is not proper and we have to switch over for another alternative of pair-wise comparison.

**STAGE 3**

**Determination of the self weight of each of the alternative with respect to each attribute**

Using the same procedure as in stage 2, determine self weight of each alternative w.r.t. each attribute.

**STAGE 4**

**Overall priority weight determination of each of this alternative**

The final stage of AHP is to compute the contribution of each alternative to the overall goal.

The overall priority for each alternative is obtained by summing the product of the attribute weight and the alternative weight w.r.t. that attribute.

In this way overall priority for each alternative is computed [16].

**3.4 GENETIC ALGORITHM**

**3.4.1 Introduction to Genetic Algorithm**

The basic principle of GA are first laid down by John Holland in 1970 and got popular in late 1980. They are based on genetic processes of biological organism and natural selection such as survival of the fittest.

A genetic algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms (also known as evolutionary computation) that use techniques inspired by evolutionary biology.

Genetic algorithms are one of the best ways to solve a problem for which little is known. They are a very general algorithm and so will work well in any search space. All you need to know is what you need the solution to be able to do well, and a genetic algorithm will be able to create a high quality solution. Genetic algorithms use the principles of selection and evolution to produce several solutions to a given problem.
Genetic algorithms tend to thrive in an environment in which there is a very large set of candidate solutions and in which the search space is uneven and has many hills and valleys. True, genetic algorithms will do well in any environment, but they will be greatly outclassed by more situation specific algorithms in the simpler search spaces. Therefore you must keep in mind that genetic algorithms are not always the best choice. Sometimes they can take quite a while to run and are therefore not always feasible for real time use. They are, however, one of the most powerful methods with which to (relatively) quickly create high quality solutions to a problem[17].

Genetic algorithms are a very effective way of quickly finding a reasonable solution to a complex problem. Granted they aren't instantaneous, or even close, but they do an excellent job of searching through a large and complex search space. Genetic algorithms are most effective in a search space for which little is known. This is where genetic algorithms thrive. They produce solutions that solve the problem in ways that never have even considered. Then again, they can also produce solutions that only work within the test environment and flounder once tried to use them in the real world [18].

3.4.2 Comparison of Natural GA Terminology

STRINGS in GA are analogous to CHROMOSOME in biological system. In natural terminology, we say those chromosomes are composed of GENES, which may take some value called ALLELES. In GA strings are composed of FEATURE, which may take value called FEATURE VALUE. In biological system gene position is given by LOCUS while in GA feature is given by STRING POSITION. In biological system total genetic package is called GENOTYPE. In GA total package is given by POPULATION/STRUCTURE. In natural system organism formed by interaction of genotype with external environment is called as PHENOTYPE. In GA structures decode to form PARAMETER SET.

The correspondence between natural and GA terminology is summarized in following way:

<table>
<thead>
<tr>
<th>NATURAL</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromosome</td>
<td>String</td>
</tr>
<tr>
<td>Gene</td>
<td>Feature</td>
</tr>
<tr>
<td>Allele</td>
<td>Feature value</td>
</tr>
<tr>
<td>Locus</td>
<td>String position</td>
</tr>
<tr>
<td>Genotype</td>
<td>Population /Structure</td>
</tr>
<tr>
<td>Phenotype</td>
<td>Parameter set / Alternate solution</td>
</tr>
</tbody>
</table>
3.4.3 Need of GA

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

The genetic algorithm differs from a classical, derivative-based, optimization algorithm in two main ways, as summarized in the following Table. Classical Algorithm Genetic Algorithm

1. Classical Algorithm generates a single point at each iteration. The sequence of points approaches an optimal solution. Genetic Algorithm generates a population of points at each iteration. The best point in the population approaches an optimal solution.
2. Classical Algorithm selects the next point in the sequence by a deterministic computation. Genetic Algorithm selects the next population by computation which uses random number generators

3.4.4 Methodology of Genetic Algorithm (GA)

Genetic algorithms are implemented as a computer simulation in which a population of candidate solutions to an optimization problem evolves toward better solutions. The evolution usually starts from a population of randomly generated individuals and are represented in binary as strings of 0s and 1s, but other encodings are also possible. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population or reaching a certain population convergence or Stop when no improvements have been found in the last n generations. Procedure for GA is given as follows
• **Initialization (Generation of Initial Population)**

Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found as shown in Fig. 3.2

![Fig 3.2 Flow Chart of Genetic Algorithm](image)

• **Representation (Coding)**

A standard representation of the solution is as an array of bits. A representation of a solution in an array of bits, where each bit represents a different object, and the value of the bit (0 or 1)
Evaluation of Fitness Function for Initial Population

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent.

A fitness function is a particular type of objective function that quantifies the optimality of a solution (a chromosome) in a genetic algorithm so that that particular chromosome may be ranked against all the other chromosomes. Optimal chromosomes, or at least chromosomes which are more optimal, are allowed to breed and mix their datasets by any of several techniques, producing a new generation that will (hopefully) be even better.

Another way of looking at fitness functions is in terms of a fitness landscape, which shows the fitness for each possible chromosome.

An ideal fitness function correlates closely with the algorithm's goal, and yet may be computed quickly. Speed of execution is very important, as a typical genetic algorithm must be iterated many, many times in order to produce a usable result for a non-trivial problem.

Definition of the fitness function is not straightforward in many cases and often is performed iteratively if the fittest solutions produced by GA are not what are desired. In some cases, it is very hard or impossible to come up even with a guess of what fitness function definition might be. Interactive genetic algorithms address this difficulty by outsourcing evaluation to external agents (normally humans).

Selection (Reproduction)

Selection is the stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding (recombination or crossover). There are several generic selection algorithms, such as tournament selection, fitness proportionate selection (roulette-wheel selection), stochastic universal sampling, truncation selection, stochastic remainder sampling and elitism. The stochastic remainder sampling and elitism are used for superfit candidate selection i.e. intermediate population selection.

Crossover (Recombination)

In genetic algorithms, crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to biological crossover, upon which genetic algorithms are based. Crossover is mechanism for diversification. The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Cross over proceeds in two steps:
(1) Select parent pair (pair off) from mating pool randomly.

(2) Using the pre-set crossover probability, $p_c$, throw a random $r \in (0, 1)$

Repeat the above process until next generation is full.

Crossover techniques used are of the following type.

One-point crossover, Two-point crossover, Cut and splice, Uniform Crossover (UX), Half Uniform Crossover/ partial uniform crossover (PUX), Arithmetic, Heuristic

- **Mutation**

In genetic algorithms, mutation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next. It is analogous to biological mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should usually be set fairly low (0.01 is a good first choice). If it is set to high, the search will turn into a primitive random search.

**Accepting:**

Accepting place mutated offspring in new population.

**Termination criterion**

Use new population for a further run of algorithm up to desired position i.e. The new population is then used for next iteration of the algorithm. Commonly, the algorithm terminates when either

- A maximum number of generations has been produced. A satisfactory fitness level has been reached for the population or
- Reaching a certain population convergence or
- Stop when no improvements have been found in the last n generations.

**Decoding for Optimal solution**

After satisfying termination criterion, convert the highest fit binary strings of each generation into decimal string and get optimum solution.

**Test**

If the optimum solution is obtained stop otherwise go to initialization step.

**3.5 COMPUTATIONAL FLUID DYNAMICS ANALYSIS [19]**

Computational analysis involves three steps mainly,
Preprocessing

It involves the discretisation of the domain into small cells or solution points. The group of cells is called as grid. The grid is generated by the preprocessing software. Grid generation is most important step in getting the solution of the problem more accurate the grid is more the solution will be.

Grid Generation

Geometry was created in GAMBIT 2.3.16.

Solver

In the solver pressure, momentum, energy equations are solved at various solution points on the grid. Suitable boundary conditions are applied in the solver. The governing equations are solved using FVM method. In the solver different parameters like material operating conditions are also specified. Suitable under relaxation factors, time step and convergence criteria are chosen.

For the solution FLUENT 6.3 solver is used. The solver works on finite difference method.

Problem setup in the solver:

The grid is imported in the solver. The grid is checked for integrity and functionality with the solver. For the solution following. The detail of the scheme used to obtained the solution selected as below;

- Solver: 3D (Double Precision).
- Solution Scheme: Implicit.
- Turbulence Model: Two equation $\kappa-\epsilon$ RNG.
- Near wall Treatment: Non equilibrium wall function.

Boundary condition setup:

Boundary conditions are setup in the solver by specifying the appropriate values at the boundary. The turbulence boundary conditions are specified by specifying the values for turbulence intensity and hydraulic diameter.

The turbulence intensity is calculated as follows;

Turbulent Intensity: $I = 0.16(Re)^{-0.125}$
For the solution initially first order schemes are used to iterate for pressure and velocity. When the flow becomes stabilized energy equation is turned on. As the solution with the first order residuals converges second order schemes are used for Velocity, Turbulence intensity and turbulent dissipation rate. Flow involves the swirl and therefore pressure iterations are carried out using PRESTO scheme.

**Post processing**

Analysis and interpretation of the obtained results is carried out in the post processing. Values of various process parameters like pressure, temperature, velocity, surface fluxes are obtained in the post processing.

**3.6 CLOSURE**

Thus in this chapter the research methodology based on design of experiment, analytical hierarchy process and genetic algorithm is described which is used for experimentation – I with the brief details and computational fluid dynamics used for experimentation – II and its possible application to offset printing process.