CHAPTER 6

OPTIMIZATION USING GENETIC ALGORITHM

Optimization is necessary to improve the classification accuracy and also to increase the Quality Of Services (QOS). Better QoS has the systems produces much higher accurate results and this leads the researchers to choose optimization field for all domains of research. In injection moulding domain also optimization is required to enhance the performance of injection system. Genetic algorithm is used as optimization tool in this proposed research work and this factors are verified based on the predicted theoretical values and experimental results.

6.1 INTRODUCTION

Characteristic injection mouldings are common all over the place in everyday life; a few examples comprise of components of automotive, electronics wares, home items and toys. Figure 6.1 reveals the investigational system operated for conducting the experiments. Injection mouldings has the principal gain of being used to build complicated geometries in a stage operate for an automated procedure. It is widespread universal mechanized procedure to create easy to complex metal, plastic and ceramic components.

Injection moulding has the capability to change thermo sets, Wax, magnesium, thermoplastics and powdered metals into multiple products. It is the generally and universally exploited mechanized procedure employing injection moulding that is differently completely in sizes, intricacy and use.
The procedure for injection moulding needs the utilization moulding machine, a mould and raw plastic material.

![Image](image.png)

**Figure 6.1 Experimental setup of Injection Moulding**

**6.2 MATERIALS USED**

HDPE is the natural colour polymer with good process ability, very good mechanical properties. HDPE is designed to make injection moulded product like industry handling, pallets and luggage shells. In this density and melt flow index are used to define the material property and the flexural strength is used to define the durability of the material.
### Table 6.1 General property of HDPE material

<table>
<thead>
<tr>
<th>S.NO</th>
<th>Property</th>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Density</td>
<td>g/cc</td>
<td>0.96</td>
</tr>
<tr>
<td>2</td>
<td>Melt Flow Index</td>
<td>g/10 min</td>
<td>8.0</td>
</tr>
<tr>
<td>3</td>
<td>Tensile strength</td>
<td>Mpa</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>Elongation at yield</td>
<td>%</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>Elongation at break</td>
<td>%</td>
<td>800</td>
</tr>
<tr>
<td>6</td>
<td>Flexural strength</td>
<td>Mpa</td>
<td>30</td>
</tr>
</tbody>
</table>

### 6.3 EXPERIMENTAL PROCEDURE

The experiments were performed on injection moulding machine De–Tech85LNC5 to produce plastic disc of 3mm thickness and 100mm diameter. The discs were produce by HDPE of grade 080M60 material. The experiment were carried out with four controllable, parameters melt temperature, injection pressure, packing pressure, packing time, with three level processing. Therefore the L27 orthogonal array was selected for this study. It is the difference between the size of a mould cavity and the size of the finished part divided by the size of a mould. Usually it is expressed in percentage. Four points were marked on the specimen, and measurements were made with micrometer screw gauge (with an accuracy of 0.01 mm). For each specimen, the average thickness was calculated as the arithmetic mean of the three points. The relative shrinkage was determined as

\[ s = \frac{D_m - D_p}{D_m} \times 100\% \]  

(6.1)
Based on the literature of injection moulding process discussed in the second chapter and the objectives of the investigation the experiments were planned. This experimental work throws light on the influence of moulding process parameters on shrinkage which is not adequately studied till date.

6.4 PRODUCTION CYCLE (PC) IN INJECTION MOULDING

The production cycle in the injection moulding process has the following stages with different cycles of action plan.

1. Connection of the injection and mould units

2. Screw back I

3. Mould closed

4. Injection

5. Packing

6. Screw back II.

7. Ejection

Cycle (3) to (7) is repeated.

6.4.1 Stages in the Injection Moulding Cycle

There are three main stages in the injection moulding cycle; stage 1, injection, followed by stage 2, holding pressure and plasticizing, and finally, stage 3, ejection of the moulded part. When stage 3 is completed, the mould closes again and the cycle is repeated.
Stage 1: Injection of the plastic melts into the Mould.

In stage 1, the mould is closed and the nozzle of the extruder is pushed against the sprue bushing of the mould. The screw, not rotating at this point, is pushed forward so that the plastic melt in front of the screw is forced into the mould.

Stage 2: Holding Pressure and Plasticising

When the mould is completely filled, the screw remains stationary for some time to keep the plastic in the mould under pressure which is called the “hold” time. During the hold time additional melt is injected into the mould to compensate for contraction due to cooling. Later, the gate, which has the narrow entrance into the mould, freezes. At this point, the mould is isolated from the injection unit. However, the melt within the mould is still at high pressure. As the melt cools and solidifies, the pressure should be high enough to avoid sink-marks, but low enough to allow easy removal of the parts.

During the plastication stage, the material is pushed forward from the feed hopper through the barrel and towards the nozzle by a rotating screw. When the gate freezes, the screw rotation is started. The period of screw rotation is called screw “recovery”. The rotation of the screw causes the plastic to be conveyed forward. As the plastic moves forward, heat from the electric heater bands along the barrel and shear starts to melt the plastic. At the discharge end of the screw, the plastic will be completely melted. The melt that accumulates at the end of the screw pushes the screw backward. Thus the screw rotates and moves backward at the same time. The rate at which plastic melt accumulates in front of the screw can be controlled by the screw backpressure, that is, the hydraulic pressure exerted on the screw. This also controls the melt pressure in front of the screw.
When a sufficient molten material gets accumulated in front portion of the screw, the rotation of the screw stops. During screw recovery the plastic in the mould is cooled, but typically the cooling is not finished by the end of screw recovery. As a result, the screw will remain stationary for some period until cooling is completed. This period is often referred to as “soak” time. During this time, additional plastic will melt in the extruder from conductive heating. Also, the melted material will reach more thermal uniformity, although the soak time is usually too short to improve thermal homogeneity significantly.

Stage 3: Ejection

When the material in the mould has cooled sufficiently to retain its shape, the mould opens and the parts are ejected from the mould and when the moulded part has been ejected, the mould closes and the cycle is repeated. The different stages can be graphically illustrated as shown in Figure 6.2. The top bar shows the movement of the extruder screw, the second bar shows the action going on inside the mould and the third bar indicates at what times the mould is open and closed. The major part of the injection moulding cycle is the cooling time required for the plastic in the mould to reduce to a temperature where the part can be removed without significant distortion. The main variable that determines the cooling time is the thickness of the moulded part.

6.5 MOULDING CONDITION

6.5.1 Resin Temperature

When moulding NOVADURAN, resin temperature should be generally about 240°C ~ 265°C. Liquidity will be better as the temperature
rises, but extremely high temperature will accelerate heat degradation which will end up with physicality deterioration of the moulded article.

6.5.2 Injection and Pressure

(a) Pressure

Injection pressure can be considered as the fill pressure (primary pressure) and the hold pressure (secondary pressure). Generally the fill pressure will be set stronger than the hold pressure. When low-temperature solidification, crystalline resin like NOVADURAN will cause a big shrink, therefore the hold pressure is necessary for filling up and is closely related to the moulding shrinkage. Increasing the hold pressure is effective to resolve sink and void problem, but if it increases too much, it might cause burr, hence attention is required.

(b) Injection speed

In the case of thin moulded product or multi-cavity moulded product where more size precision is required, faster injection speed is prefered. In contrast, slower injection speed is better for thick moulded product. Also, the program control of injection speed is effective to resolve the jetting and the flow mark.

(c) Injection time

Setting will differ by the moulding machine, but basically should be considered as below. Injection time (filling time + pressure keeping time) i.e., gate sealing time. Gate sealing time is the time when resin stops flowing by solidification at the gate part. If pressure keeping is put away before the gate is sealed, molten resin will backflow from the gate by the tool internal pressure, which will cause measurement and physicality variability, and war
page, sink, and void problems, because of decrease in moulded product's filling density (packing property). To estimate the gate sealing time, the weight of moulded product is measured by gradually increasing the injection time, and look for the injection time when the weight of moulded became a certain amount and stop changing.

(d) Back flow prevention

The measurement might become instable by the gas and the air generated from molten resin when plasticization occurs. To stabilize the measurement and improve the kneading effect, the screw back pressure is kept between 5 ~ 10kg/cm². However, if the back pressure is too strong, it might degrade the plasticization ability.

As for compound reinforced PC such as glass fiber reinforced material etc. the backflow prevention ring sometimes cracks when the load becomes large compared with the non-reinforced material. When moulding without being aware of this, the uneven dimension and the deviation from tolerance in the moulding of a precise part occur due to the unstable measurement. It is necessary to note that such a trouble easily occurs in case of overload and insufficient purge.

6.6 PROCESS VARIABLES

Each process variable can be categorized into one of the five main types such as speed, pressure, time, temperature and stroke. The relationship between the five process variables is of an interactive nature as each variable cannot be readily isolated. This relationship can be simply demonstrated, for example, upon increasing the hydraulic back pressure, the linear retraction speed of the screw (during recovery) changes causing an increase in the screw
recovery time, the melt temperature and/or homogeneity. As a result of the increase in the melt temperature further changes occur to the mould fill time, the injection pressure, the mould temperature, the product ejection temperature and the product dimensions. Hence, by increasing a pressure variable like, the hydraulic back pressure, three other main variable types are collectively influenced. More importantly, the process and subsequently the moulded components are affected.

When changing to a particular process variable or machine setting to occur (which significantly affect the stability of the moulding process so that defective components are produced) it is important that the correct process variable should be controlled so as to rectify the disturbance. For instance, the selection of the wrong hopper throat temperature can cause short mouldings to be produced which then misleads the moulder into altering other variables like the holding pressure and/or, the shot volume and/or, the mould filling speed to overcome the short moulding problem. As the initial selection was incorrect, the process remains unstable but, in changing another variable type, the moulder is led to believe that the problem is resolved. However, in reality defective and/or inconsistent parts will continue to be produced throughout the production run. The following headings highlight typical process variables which need to be monitored and/or controlled during each cycle. Each of the listed variables will not be discussed in detail to highlight the importance of each variable with respect to the stability of the process. Process variables can be categorized as follows:

6.7 **OPTIMIZATION METHOD**

A key element of what is assumed to be intelligence, the capability to learn from past experience. Especially when things are done repeatedly, intelligent behaviour would avoid doing a mistake more than once and would
prefer making advantages decision again. For optimization a class of today’s most popular heuristic approaches is known as GAs. Due to its widespread use and vast amount of literature dealing with GAs, a comprehensive review of research activities is doomed to failure; thus we stick to outline of the fundamental ideas. The adjective genetic reveals the roots of these algorithms. Adapting the evaluation strategy from natural life form, the basic idea is to start with a set of (feasible) solutions and to compute a set of new solutions by applying some well defined operations on the old ones. Then, some solutions (new and/or solutions) are selected to form a new set with which another is started, and on until some stopping criterion is met. Solutions are represented by sets of attributed and different solutions are met by different collections of attribute values. The decision that solutions are dismissed and which is taken over to form a new starting point for the next iteration is made on the basis of priority rule. GAs is adaptive heuristic search algorithm promised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem. First pioneered by John Holland in the 60s, GAs has been widely studied, experimented and applied in many fields in engineering worlds. Not only does GAs provide alternative methods to solving problem, it consistently out performs other traditional methods in most of the problems link. Many of the real world problems involved in finding optimal parameters, which might prove difficult by traditional methods use idea of GAs. However, because of its outstanding performance in optimization, GA has been wrongly regarded as a function optimizer. In fact, there are many ways to view this. Perhaps most users come to GAs looking for a problem solver, but this is a restrictive view.
GA can be used as problem solvers, challenging technical puzzle, basis for competed machine learning, computational model of innovation and creativity, computational model of other innovating system and guiding philosophy. Many scientists have tried to create living programs. These programs do not merely simulate life but try to exhibit the behaviours and characteristics of real organisms in an attempt to exist as a form of life. Suggestions have been made that a life would eventually evolve into real life. Such suggestion may sound absurd at the moment but certainly not implausible if technology continues to progress at present rates. Nearly everyone can gain benefits from GAs, once the encode solutions of a given problem to chromosomes in GA, and compare the relative performance (fitness) of solutions. An effective GA representation and meaningful fitness evaluation are the keys of the success in GA applications. The appeal of GAs comes from their simplicity and elegance as robust search algorithms as well as from their power to discover good solutions rapidly for difficult high-dimensional problems. GAs is useful and efficient when,

i. The search space is large, complex or poorly understood.

ii. Domain Knowledge is scarce or expert knowledge is difficult to encoded to narrow the search space.

iii. No mathematical analysis is available.

iv. Traditional search methods fail.

The advantage of the GA approach is the ease with which it can handle arbitrary kinds of constraints and objectives; all such things can be handled as weighted components of the fitness function, making it easy to adapt the GA scheduler to the particular requirements of a very wide range of possible overall objectives. GAs has been used for problem solving and for
modelling. GAs are applied to many scientific, engineering problems, in business and entertainment, including: optimization, automatic programming, machine and robot learning, economic models, immune system models, ecological models, population genetics models, interactions between evolution and learning and models of social systems.

6.8 FLOW PROCESS OF GENETIC ALGORITHM

The algorithm operates through a simple cycle:

1. Creation of a population of strings.
2. Evaluation of each string.
3. Selection of the best strings.
4. Genetic manipulation to create a new population of strings.

Figure 6.2 shows how these four stages interconnect. Each cycle produces a new generation of possible solutions (individuals) for a given problem.

![Diagram](image.png)

**Figure 6.2 The reproduction cycle**
At the first stage, a population of possible solutions is created as a start point. Each individual in this population is encoded into a string (the chromosome) to be manipulated by the genetic operators. The manipulation process applies the genetic operators to produce a new population of individuals, the offspring, by manipulating the genetic information possessed by the pairs chosen to reproduce. This information is stored in the strings (chromosomes) that describe the individuals. Two operators are used: Crossover and mutation. The offspring generated by this process take the place of the older population and the cycle is repeated until a desired level of fitness is attained or a determined number of cycles are reached.

6.8.1 Crossover

Crossover is one of the genetic operators used to recombine the population genetic material. It takes two chromosomes and swaps part of their genetic information to produce new chromosomes. This operation is similar to sexual reproduction in nature.

![Crossover process diagram](Image)

**Figure 6.3 Crossover process**

Figure 6.3 shows after the crossover point has been randomly chosen, portions of the parent’s chromosome (strings) Parent 1 and Parent 2 are combined to produce the new offspring Son. The selection process
associated with the recombination made by crossover assures that special genetic structures, called building blocks, are for future generations. These building blocks represent the fit genetic structures in the population.

### 6.8.2 Mutation

The recombination process alone cannot explore search space, sections not represented in the population’s genetic structures. This could make the search get stuck around local minima. Here mutation goes into action. The mutation operator introduces new genetic structures in the algorithm escape local minima traps. Since the modification is totally random and thus not related to any previous genetic structures present in the population, it creates different structures related to others sections of the search space. Figure 6.4 mutation is implemented by occasionally altering a random bit from a chromosome (string), it shows the operator being applied to the fifth element of the chromosome.

![Mutation Diagram](image)

**Figure 6.4 Mutation**

A number of other operators, other than crossover and mutation, have been introduced since the basic model was proposed. They are usually versions of the recombination and genetic alterations processes adapted to
constraints of a particular problem. Examples of other operators are: inversion, dominance and genetic edge recombination.

6.8.3 Problem dependent parameters

This description of the GAs’ computational model reviews the steps needed to create the algorithm. However, a real implementation takes account of number of problem-dependent parameters. Problem constraints will dictate the best option. Other parameters to be adjusted are the population size, crossover and mutation rates.

6.8.4 Encoding

Critical to the algorithm performance is the choice of underlying encoding for the solution of the optimization problem (the individuals on the population). Traditionally, binary encoding has being used because they are easy to implement. The crossover and mutation operators described earlier are specific to binary encodings. When symbols other than 1’s or 0’s are used, the crossover and mutation operators must be tailored accordingly. A large number of optimization problems have continuous variables. A common technique for encoding them in the binary form uses a fixed-point integer encoding, each variable being coded using a fixed number of bits. The binary code of the entire variable can then be concatenated in the strings of the population. A drawback of encoding variables as binary strings is the presence of Hamming cliffs: 01111 and 10000 are integer representations of 15 and 16, respectively, and have a Hamming distance of 5. For the GA to change the representation from 15 to 16, it must alter all bits simultaneously. Such Hamming cliffs present a problem for the algorithm, as both mutation and crossover cannot overcome them easily.
It is desirable that the encoding makes the representation as robust as possible. This means that even if a piece of the representation is randomly changed, it will still represent a viable individual. For instance, suppose that a particular encoding scheme describes a circuit by the position of each of its components and a pointer to their individual descriptions. If this pointer is the description’s memory address, it is very unlikely that, after a random change in its value, the pointer will still point to a valid description. But, if the pointer is a binary string of 4 bits pointing into an array of 16 positions holding the descriptions, regardless of the changes in the 4 bit string, the pointer will always point to a valid description. This makes the arrangement more tolerant to changes, more robust.

**6.8.5 Evaluation**

The evaluation step in the cycle is the one more closely related to the actual system the algorithm is trying to optimize. It takes the strings representing the individuals of the population and, from them, creates the actual individuals to be tested. The way the individuals are coded in the strings will depend on what parameter one is trying to optimize and the actual structure of possible solutions (individuals).

However, the resulting strings should not be too big otherwise the process will get very slow, but should be of the right size to represent well the characteristics to be optimized. After the actual individuals have been created they have to be tested and scored. These two tasks again are much related to the actual system being optimized. The testing depends on what characteristics should be of the right size to represent well the characteristics to be optimized.
6.9 APPLICATIONS OF GENETIC ALGORITHM

1. GA in optimization and planning: TSP.

2. GA in Business and Their Supportive Role in Decision Making.

GAs has been used to solve many different types of business problems in functional areas such as finance, marketing, information systems, and production/such as tactical asset allocation, job scheduling, machine-part grouping, and computer network design. If the conception of a computer algorithms are analyzed based on the evolutionary of organism is surprising, the extensiveness with which this algorithms is applied in so many areas. These applications, be they commercial, educational and scientific, are increasingly dependent on this algorithms, the GAs. Its usefulness and gracefulness of solving problems has made it the more favourite choice among the traditional methods, namely gradient search, random search and others. GAs is very helpful when the developer does not have precise domain expertise, because GAs possesses the ability to explore and learn from their domain. Figure 6.5 provides the details of flowchart for general procedure of Genetic algorithm.
6.10 INTRODUCTION TO SIMULATED ANNEALING

SA is a generalization of a Monte Carlo method for examining the equations of state and frozen states of n-body systems. The concept is based on the manner in which liquids freeze or metals recrystallize in the process of annealing. In an annealing process a melt, initially at high temperature and
disordered, is slowly cooled so that the system at any time is approximately in thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered and approaches a frozen ground state at $T=0$. Hence the process can be thought of as an adiabatic approach to the lowest energy state. If the initial temperature of the system is too low or cooling is done insufficiently slowly the system may become quenched forming defects or freezing out in metastable states (i.e. trapped in a local minimum energy state).

The original Metropolis scheme was that an initial state of a thermodynamic system was chosen at energy $E$ and temperature $T$, holding $T$ constant the initial configuration is perturbed and the change in energy, $\Delta E$ is computed. If the change in energy is negative the new configuration is accepted. If the change in energy is positive it is accepted with a probability given by the Boltzmann factor $\exp(-\Delta E/T)$. This process is then repeated sufficient times to give good sampling statistics for the current temperature, and then the temperature is decremented and the entire process repeated until a frozen state is achieved at $T=0$.

By analogy the generalization of this Monte Carlo approach to combinatorial problems is straightforward. The current state of the thermodynamic system is analogous to the current solution to the combinatorial problem, the energy equation for the thermodynamic system is analogous to at the objective function, and ground state is analogous to the global minimum. The major difficulty (art) in implementation of the algorithm is that there is no obvious analogy for the temperature $T$ with respect to a free parameter in the combinatorial problem. Furthermore, avoidance of entrainment in local minima (quenching) is dependent on the “annealing schedule”, the choice of initial temperature, how much iteration
are performed at each temperature, and how much the temperature is decremented at each step as cooling proceeds.

The method itself has a direct analogy with thermodynamics, specifically with the way that liquids freeze and crystallize, or metals cool and anneal. At high temperatures, the molecules of a liquid move freely with respect to one another. If the liquid is cooled slowly, thermal mobility is restricted. The atoms are often able to line themselves up and form a pure crystal that is restricted. This crystal is the state of minimum energy for the system, which would correspond to the optimal solution in a mathematical optimization problem. However, if a liquid metal is cooled quickly i.e., quenched, it does not reach a minimum energy state but a somewhat higher energy state corresponding, in the mathematical sense, to a sub optimal solution found by iterative improvement or hill-climbing.

In order to make use of this analogy with thermodynamical systems for solving mathematical optimization problems, one must first provide the following elements:

i. A description of possible system configurations, i.e. some way of representing a solution to the minimization (maximization) problem, usually this involves some configuration of parameters that represent a solution.

ii. A generator of random changes in a configuration; these changes are typically solutions in the neighbourhood of the current configuration, for example, a change in one of the parameters.

iii. An objective or cost function $E(X)$ (analog of energy) whose minimization is the goal of the procedure.
iv. A control parameter $T$ (analogue of temperature) and an annealing schedule which indicates how is lowered from high values to low values e.g. after how many random changes in configuration is $T$ reduced.

There are two aspects of the SA process that are areas of active research. The first is the cooling schedule and the second is determining how many Monte Carlo steps are sufficient at each temperature. If the temperature is decreased too slowly, CPU-cycles will be wasted. On the other hand, if the cooling is too rapid, the search will be trapped in a sub optimal region of search space. Two different cooling schedules are generally used in practice. The first reduces the temperature by a constant amount in each phase, while the second reduces the temperature by a constant factor (e.g. 10%). The first method allows the simulation to proceed for the same number of Monte Carlo steps in the high, intermediate and low temperature regimes, while the latter causes the simulation to spend more time in the low temperature regime than the high one.

A simulation at a fixed temperature can be run before the full simulation and the expectation value determined as a function of the number of Monte Carlo steps. An understanding of this behaviour is used to estimate the number of attempts that should be made at each temperature. As stated for many of the other methods described earlier, SA is a computational methodology, not a fixed algorithm, or program. This means that the problem should dictate which variables should be changed at each Monte Carlo step, the magnitude of their change, the number of Monte Carlo attempts at each temperature and the cooling schedule. As such, a canned program that is not sufficiently flexible should not be used for more than educational purposes. SA has been in various combinatorial optimization problems and has been particularly successful in circuit design problems.
A neighbourhood structure is superimposed on the usually finite but large space of feasible solutions (configuration or in this context production schedules). Given current feasible configuration, say in this context, current sequence ($\sigma_{cur}$) a candidate solution ($\sigma_{can}$) is drawn randomly from the corresponding neighbourhood. This new configuration will be accepted subjected to either improvement of the objective function or another random experiment with acceptance probability given by $e^{\frac{-\Delta C}{\gamma}}$ where $\Delta C = C(\sigma_{cur}) - C(\sigma_{can})$ is the difference of the cost function values of the candidate and the current configuration. $\gamma$ is the control parameter corresponding to temperature in the original physical analogue in thermodynamics.

6.11 HYBRID GA - SIMULATED ANNEALING

The motivation behind the GA-SA combination is the power of GA to work on the solution in a global sense while allowing SA to locally optimize each individual solution. The Hybrid GA-SA algorithm is explained in the following section. Many researchers have proposed the concept of joining the two powerful optimization techniques namely GA and SA. The steps followed are given below:

1. Initialize the parameters of the GA.
2. Generate the initial population.
3. Execute GA for one generation.
4. For each of the chromosomes do the following:
5. Initialize the parameters of the SA.
6. Improve the quality of solution using SA and the string with the best solution is returned to the new population of GA.

7. Repeat the steps 3 and 4 for required number of generations.

The hybrid algorithm executes in two phases, the GA and SA. In the first phase, the GA generates the initial solutions (only once) randomly. The GA then operates on the solutions using selection, crossover and mutation operators on the solutions using selection, crossover and mutation operators to produce new and hopefully better solutions. After each generation the GA sends each solution to the SA (second phase) to be improved. The neighbourhood generation scheme used in SA is a single insertion neighbourhood scheme. Once the SA is finished for a solution of GA, another solution GA is passed to SA.
This procedure progresses pending a situation where every solution of GA in a generation has been used up. As soon as the SA is completed for every solution in generation of the GA, the highest quality solutions for the
population size attained from SA are taken as the solutions of the GA for the subsequent generation. The interchange of the GA and SA progresses pending when the needed amount of generations are finished. Figure 6.6 provides the detailed flowchart of Hybrid Genetic algorithm integrated with simulated annealing.

6.12 EXPERIMENTAL FINDINGS

The Genetic algorithm used to optimize the proposed model and the optimization parameters includes the mould temperature, melt temperature, filling time, injection time, packing time and holding pressure. Based on the results obtained from simulation, minimum and maximum values are obtained and it makes the system into an optimized new generation model. Numerical experiments are carried out routed in these optimization parameters and the computed volumetric shrinkage values are utilized. Proposed model predicts the maximum and minimum value as 0.354 and this value is larger than the predicted value based on theoretical experimentations.

![Figure 6.7 Melt Temperature vs. Volume Variation](image_url)
Figure 6.7 depicts the melt temperature against volume variation for the predicted value and the experimental value. The predicted value is slightly lesser than the actual experimental value because of the physical properties of the material which is used for the moulding process.

![Graph showing melt temperature and volume variation](image)

**Figure 6.8 Mould Temperatures against Volume Variation**

Figure 6.8 depicts the volume variation against mould temperature. In this also experimental results are slightly higher than the predicted value.
Figure 6.9 Injection Time against Volume Variation

Figure 6.9 depicts the injection time against volume variation. In this process, at the range of 3.9 to 4.0 both the predicted values and experimental values are merely equal and from that observation, the proposed model has enhanced results which are much better than the theoretical value.
Figure 6.10 Packing Time against Volume Variation

Packing time and volume variation is related in Figure 6.10 for the theoretical values and experimental values. In general, once the material is ready to produce any object based on the molten temperature, this pressure is a factor to be considered.

Figure 6.11 describes that relation between the pressure and volume variation. Table 6.2 distributes the details for 300 generations selected for genetic algorithm and parametric performance are also plotted in Figure 6.11.
Figure 6.11 Pressures against Volume Variation

Table 6.2 Population after 300 generations

<table>
<thead>
<tr>
<th>S.No</th>
<th>Melt temperature (°C)</th>
<th>Mold temperature (°C)</th>
<th>Injection time (s)</th>
<th>Packing time (s)</th>
<th>Holding pressure (MPa)</th>
<th>Volume of mould cavity mm$^3$</th>
<th>Volume of product mm$^3$</th>
<th>Volumetric variation mm$^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>250</td>
<td>59.0</td>
<td>3.758</td>
<td>11.400</td>
<td>31.000</td>
<td>2094.3951</td>
<td>2092.341</td>
<td>2.0541</td>
</tr>
<tr>
<td>2</td>
<td>255</td>
<td>59.0</td>
<td>3.791</td>
<td>11.450</td>
<td>31.500</td>
<td>2094.3951</td>
<td>2092.341</td>
<td>2.0541</td>
</tr>
<tr>
<td>3</td>
<td>260</td>
<td>59.5</td>
<td>3.812</td>
<td>11.500</td>
<td>31.550</td>
<td>2094.3951</td>
<td>2092.2791</td>
<td>2.0542</td>
</tr>
<tr>
<td>4</td>
<td>265</td>
<td>60.0</td>
<td>3.852</td>
<td>11.550</td>
<td>32.000</td>
<td>2094.3951</td>
<td>2092.3406</td>
<td>2.0545</td>
</tr>
<tr>
<td>5</td>
<td>269</td>
<td>60.0</td>
<td>3.899</td>
<td>12.500</td>
<td>32.500</td>
<td>2094.3951</td>
<td>2092.3406</td>
<td>2.0545</td>
</tr>
<tr>
<td>6</td>
<td>272</td>
<td>61.0</td>
<td>3.922</td>
<td>12.550</td>
<td>35.550</td>
<td>2094.3951</td>
<td>2092.3405</td>
<td>2.0546</td>
</tr>
<tr>
<td>7</td>
<td>275</td>
<td>61.5</td>
<td>3.992</td>
<td>13.000</td>
<td>33.000</td>
<td>2094.3951</td>
<td>2092.3405</td>
<td>2.0546</td>
</tr>
<tr>
<td>8</td>
<td>280</td>
<td>62.0</td>
<td>4.152</td>
<td>13.500</td>
<td>33.500</td>
<td>2094.3951</td>
<td>2092.3405</td>
<td>2.0546</td>
</tr>
<tr>
<td>9</td>
<td>282</td>
<td>62.0</td>
<td>4.152</td>
<td>14.000</td>
<td>33.550</td>
<td>2094.3951</td>
<td>2092.3407</td>
<td>2.0544</td>
</tr>
<tr>
<td>10</td>
<td>285</td>
<td>62.0</td>
<td>4.210</td>
<td>14.500</td>
<td>34.000</td>
<td>2094.3951</td>
<td>2092.3408</td>
<td>2.0543</td>
</tr>
</tbody>
</table>

Figure 6.12 depicts the comprehensive plot of parameters distributed over a range of values and it could be found that the dominant factor is the melting temperature.
This chapter has investigated the utilization of genetic based optimization for moulding process with several attributes taken into account such as melt temperature, mould temperature, the injection time, packing time, holding pressure and variations in the volume during the moulding process. An exhaustive experimentation has been carried out in this chapter and several parameters measured and superior performance in terms of quality and variations in volume have been recorded and reported using the proposed GA-SA optimization algorithm. It could be seen from experimental results that the proposed optimization algorithm closely follows the predicted output and hence the net resultant error is very minimal to nearly negligible. It could
also be further seen that in spite of the fact that the proposed algorithm operates in two phases, the time required for convergence to desired value is very minimal hence making it computationally efficient. Based on the experimental investigations, observations and analysis, essential inferences have been made which have been presented systematically in the succeeding chapter.