OPTIMIZATION OF MACHINING PARAMETERS USING GENETIC ALGORITHM

A Thesis submitted
in partial fulfillment of the requirements for
the award of degree of

MASTER OF ENGINEERING
IN
MECHANICAL ENGINEERING
(CAD/CAM & ROBOTICS)

By:
DEEPAK BHANDARI
(R.No. 8018104)

Under the guidance of:

Dr. H. B. Sharda
Dean of Student Affairs and Prof., MED
T. I. E. T. Patiala.

Shri Ashish Sharma
Lecturer, MED
T. I. E. T. Patiala.

DEPARTMENT OF MECHANICAL ENGINEERING
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
(DEEMED UNIVERSITY)
PATIALA-147004 (INDIA)
2003

CERTIFICATE

This is to certify that the thesis entitled **"Optimization of Machining Parameters Using Genetic Algorithm"** being submitted by **Mr. Deepak Bhandari** in partial fulfillment of the requirements for the award of the degree of **MASTER OF ENGINEERING (CAD/CAM & ROBOTICS)** at **Thapar Institute of Engineering & Technology (Deemed University), Patiala**, is a bonafide work carried out by him under our supervision and guidance.

Dr. H. B. SHARDA

Dean of Student Affairs and Prof., **MED**
Thapar Institute of Engg. & Tech.,
Patiala.

Sh. ASHISH SHARMA

Lecturer, MED
Thapar Institute of Engg. & Tech.,
Patiala.

Dr. D. S. BAWA

Dean of Academic Affairs
Thapar Institute of Engg. & Tech.,
Patiala.

Dr. R. G. TATHGIR

Professor & Head, MED
Thapar Institute of Engg. & Tech.,
Patiala.

ACKNOWLEDGEMENT

I express my sincere gratitude to my guides, **Dr. H. B. SHARDA,** Dean of Student Affairs and Prof**.,** MED and **Sh. ASHISH SHARMA,** Lecturer, MED, Thapar Institute of Engg. & Tech., Patiala for their valuable guidance, proper advice, painstaking and constant encouragement during the course of my work on this thesis.

I am deeply indebted to my parents for their inspiration and ever encouraging moral support, which enabled me to pursue my studies. I do not find enough words with which I can express my feeling of thanks to my brother **Sanjeev Bhandari** and forever friends **Sachin Mangla and Ishan Bansal,** for their help, inspiration and moral support which went a long way in successful completion of the present study.

I am also thankful to the entire faculty and staff members of Mechanical Engg. Department for their direct-indirect help and nice cooperation.

DEEPAK BHANDARI

ABSTRACT

The present competitive environment forces the organizations to stand on alert every minute and use its resources (man, material, and machine) to their best possible extent. The present work aims to explore genetic algorithm (GA) as a method for optimizing machining parameters selection. The algorithm searches for the best solution in terms of cutting speed and feed with the aim of optimizing an objective function. The objective functions considered here take into account optimizing anyone or a weighted sum of the two significant criteria- total production cost and total production time, in single pass turning operation. The GA technique used for optimization purpose provides an efficient and promising tool in view of high degree or non-linearity in the objective function. The technique, although applied here only on single pass turning, is applicable to much wider variety of machining situations.

CONTENTS

PAGE NO.

CHAPTER - 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPTER 1

INTRODUCTION

1.1 OPTIMIZATION

Optimization algorithms are becoming increasingly popular in engineering activities, primarily because of the availability and affordability of high-speed computers. They are extensively used in those engineering problems where the emphasis is on maximizing or minimizing of a certain goal.

For example, optimization algorithms are routinely used in aerospace design activities to minimize the overall weight, simply because every element or component adds to the overall weight of the aircraft. Chemical engineers on the other hand are interested in designing or operating a process plant for an optimum rate of production. Mechanical engineers design mechanical components for the purpose of achieving either a minimum manufacturing cost or maximum rate of production.

Production engineers are interested in designing optimum schedules of various machining operations to minimize the idle time of machines and the overall completion time. Civil engineers are involved in designing buildings, dams and other structures in order to achieve a minimum overall cost or maximum safety or both. Electronics engineers are interested in designing communication networks so as to achieve minimum time for communication from one node to another.

Thus, the ultimate aim of the optimization is to improve an existing process that meets the given requirements and satisfies all the restrictions/constraints placed on it. This is called the optimum process.

1.2 OPTIMIZATION IN MACHINING

Machining parameters such as speed, feed and depth of cut play vital role in machining the given work piece to the required shape. These have a major affect on the quantity of production, cost of production and production rate; hence their judicious selection assumes significance.

The selected machining parameters should yield desired quality on the machined surface while utilizing the machining resources such as machine tool and cutting tool to the fullest extent possible, consistent with the constraints on these resources. Traditionally the selection of machining parameters is carried out based on the experience of the machinist or the planner and referring the available catalogues and handbooks.

Manual selection of machining parameters reflects the problem of variability in experience and judgement among the planners. In addition to this, the induction of cost intensive NC-machines onto the shop floor, stresses more emphasis on the effective utilization of these resources using the optimal machining parameters. Present industries use both conventional and NC machines on their shop floor, hence it becomes necessary to go for automated methods to select the optimal machining parameters that suit the demands of the present industries.

Computer aided procedures have been found reliable for their fastness, accuracy and consistency in the automated selection of machining parameters compared to their manual counterparts. Various optimization techniques can be used to find the optimal machining parameters for a particular machining operation.

1.3 APPROACHES FOR MACHINING PARAMETER SELECTION

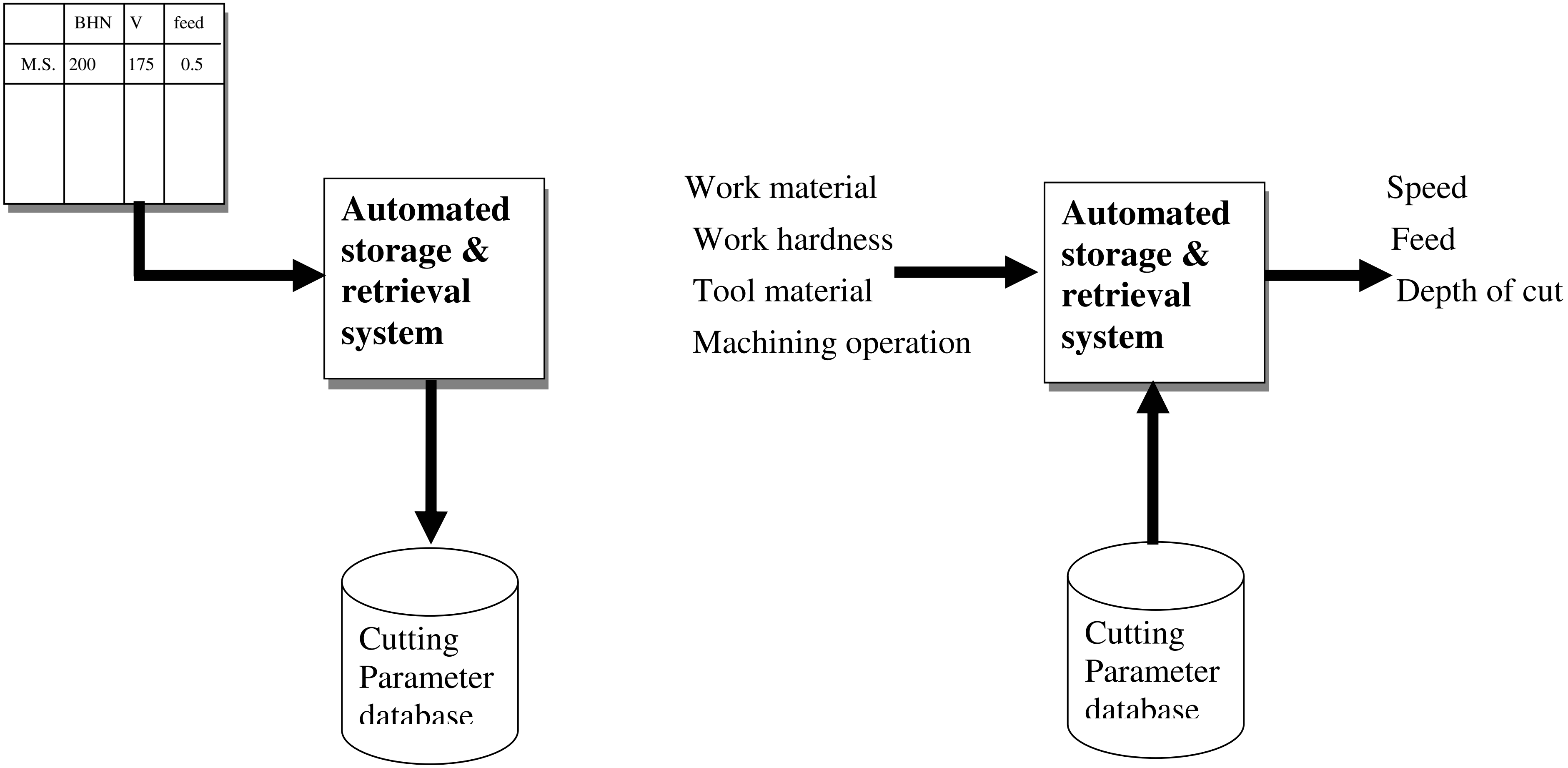
The advances in the manufacturing engineering and developments in the areas of computer aided design and computer aided manufacturing, give a high level of automation in the present competitive environment. In accordance with this, attempts have also been made to automate the selection of machining parameters. The traditional methods have been replaced by reliable computer aided procedures in the selection of machining parameters. Much of the research work has been done and ample literature is available in this direction.

The published literature reveals that there are two most popular approaches for automated selection of machining parameters as give below:

- **DATA STORAGE AND RETRIEVAL APPROACH**
- **MATHEMATICAL MODEL APPROACH**

1.3.1 DATA STORAGE AND RETRIEVAL APPROACH

This approach consists of two steps as shown in the figure (1.1). In the first step, recommended parameters for various combinations of work material, tool material and machining operation are collected from the shop floor, cutting tool and machine tool manufacturer's catalogues and handbooks. Then, these parameters are stored in a database in a structured format as shown in the figure 1.1 (a). Based on the user inputs, the proper set of cutting conditions is retrieved from the database as shown in figure 1.1 (b) and presented to the user.



(a) (b)

Figure 1.1 Data storage and retrieval approach

(a) Storage and (b) Retrieval

This is in-effect a computerized look-up table approach. Though this approach aids in the fast and consistent selection of parameters, the parameters are general in nature and a lot of computer storage space is required. Also they are often characterized by lack of adequate maintenance and difficulties in updating the current machining recommendations.

1.3.2 MATHEMATICAL MODEL APPROACH

In this approach, the problem is converted into mathematical models based on certain criteria. The criteria differ for different problems. In case of automated machining parameters selection, mathematical model are formulated to determine the machining parameters for a particular objective such as minimization of unit cost or maximization of production rate. In this approach, empirical equations formulated based on experimental data for tool life (usually extended tool life equations) and other practical restrictions on machining variables are used in the model development. The resultant function from this mathematical modeling is called objective function, which is subjected to practical constraints. The developed model is then solved using an optimization technique to yield optimal parameters that reflects the shop floor capabilities. This procedure is illustrated in Figure 1.2. The success of this approach depends on the reliability of the empirical equations used in the model formulation.

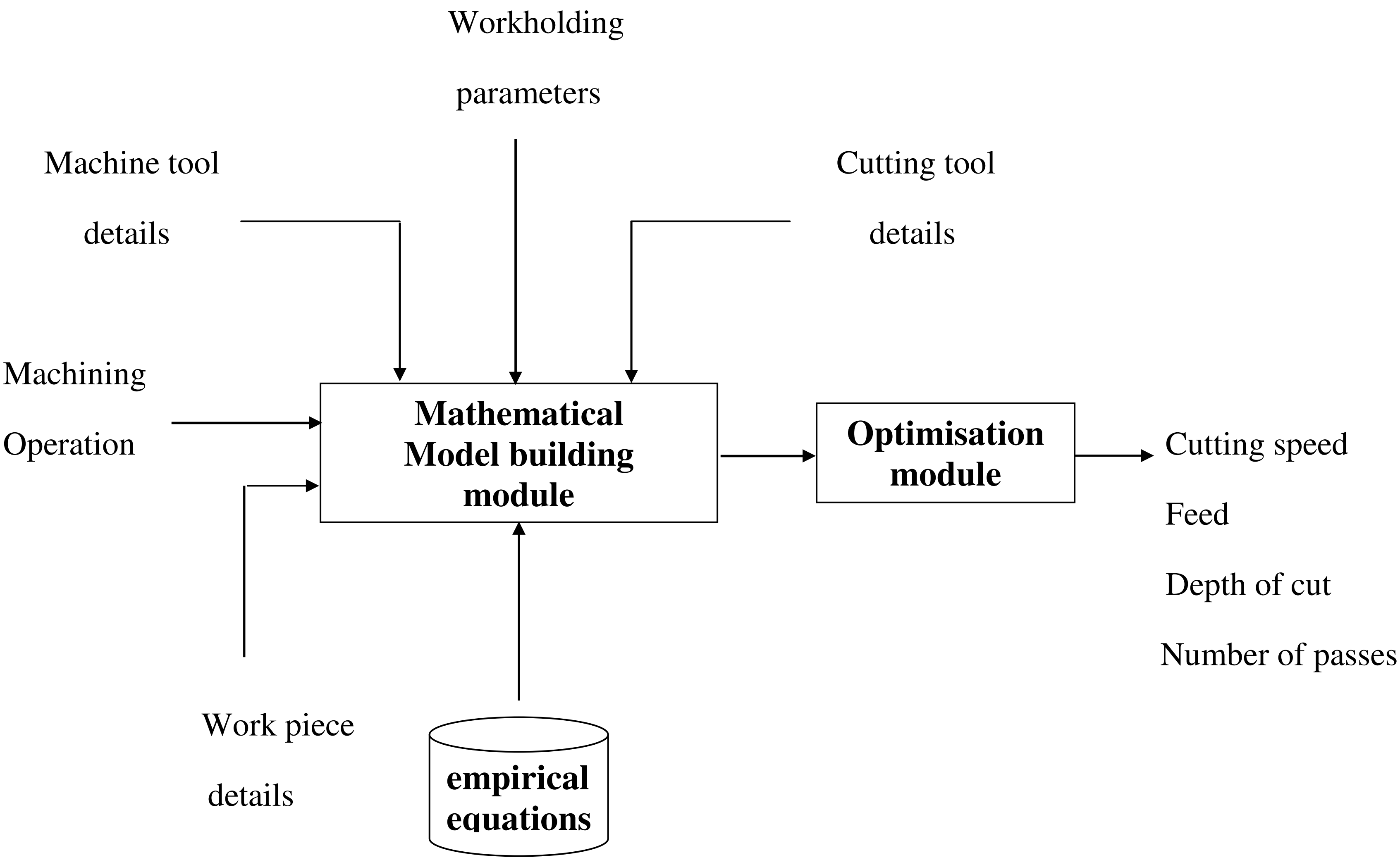


Figure 1.2 Mathematical model approach

CHAPTER 2

LITERATURE-SURVEY

Armarego and Brown [1] expressed the production cost and production rate in terms of speed and feed rate and partially differentiated these terms with respect to speed and feed which are equated to zero to get optimum cutting conditions.

Yen and Wright [2] developed a unified method of adaptive control of constraints in which a suitable cutting region is determined satisfying all the physical constraints. The objective of the optimization is to maximize the production rate under constraints of plastic deformation, crater wear and fracture. Taylor’s tool life equation is not used and the tool life equations for different model of failures are established on the basis of properties of tool and work material.

R. Meenakshi Sunderam and Tajen Cheng [5] proposed a process planning system with the aid of a microcomputer. They have developed a computer program in BASIC. In their software development, the geometric programming technique is used for the selection of optimal machining parameters.

J. S. Agapiou [6] developed a Nelder-Mead Simplex Method to determine the optimum machining conditions. An objective function incorporating a combination of the minimum production cost and minimum production time requirements is considered for the optimization. The two criteria of production time and production cost are prioritized through their weight coefficients. A constant multiplier is used to normalize the objective function. Physical constraints regarding the cutting parameters are also considered. The superiority of the combined objective function over the single criterion objective functions, using the production cost, or production time, or maximum profit rate is also illustrated.

J. S. Agapiou [7] outlined the use of two techniques (Dynamic Programming and Nelder-Mead Simplex Method) to determine the optimum cutting conditions for the multi-pass operations. The optimum number of machining passes is obtained through the dynamic programming technique and the optimum machining conditions for each pass are then determined based on the combined objective function of total production cost and the total production time. This approach can be very effectively applied to the multi-stage machining since the optimum arrangement of the different operations can be determined by the dynamic programming method while the optimum cutting conditions for the operation in each machining stage are obtained using the method for single pass (Nelder-Mead Simplex Method) while incorporating the objective function.

A. Ihsan Sonmez and Adil Baykasoglu [14] have outlined the development of an optimization strategy to determine the optimum cutting parameters for multi-pass milling operations like plain milling. The developed strategy is based on the maximum production rate and incorporates eight technological constraints. The optimum number of passes is determined via dynamic programming and the optimum values of cutting conditions are found based on the objective function by using the geometric programming technique. This paper also underlies the importance of using optimization strategies rather than handbook recommendations.

Dereli, T. and Filiz, H. I. [16] explained the application of GA for determination of optimal sequence of machining operations based on either minimum tool change or minimum tool traveling distance or safety. Combination of these criteria also might be used. They have also explained how the optimum position of tools on the automatic tool changer or turret magazine of a CNC machine tool is obtained by using GA.

T. S. Sidhu [17] outlined the use of goal programming technique in setting optimum machining parameters in single pass turning operation. The optimum values for speed and feed are determined by setting different goals under a given set of conditions. The calculated values are set on the lathe and surface finish of the turned work pieces are measured for different values of depth of cut.

Q. Meng [21] have used machining theory for calculating optimum cutting conditions in turning. The method uses a variable flow stress machining theory to predict cutting forces, stress etc. which are then used to check process constraints such as machine power, tool plastic deformation and built-up edge formation. Their results indicate that the method is capable of selecting the appropriate cutting conditions.

M. S. Shunmugam and S. V. Bhaskara Reddy [22] outlined the use of genetic algorithm for the selection of optimal conditions in multi-pass face-milling operation. In this work, the total production cost is used as an objective function. The number of passes, speed and feed are obtained using the machining theory and then the depth of cut for the each pass is determined by using the genetic algorithm. The technological constraints considered here is allowable speed and feed, dimensional accuracy, surface finish, tool wear and machine tool capabilities.

Y. V. Hui [23] developed a time dynamic economic model for single pass turning. The model incorporates considerations on the stochastic nature of the tool life and tool maintenance activities such as tool replacement and tool regrinding. Furthermore, they have developed a model of quality cost of tool cutting in terms of deviation from target roughness and deviation from target dimension. Finally, they have provided a quality machining economic model for turning to investigate the trade-off between quality cost and other cost factors.

Nafis Ahmad and Dr. A. F. M. Anwarul Haque [24] outlined the use of genetic algorithm to find out the optimum machining parameters. In this work, machining parameters for the turning rotational components are optimized by a genetic algorithm optimization toolbox developed in Matlab environment. Here machining time is considered as the objective function and constraints are machining capacity, limits of feed rate, depth of cut and cutting speed. Machining time is minimized through a series of generations while some genetic operators are applied at each generation. The result of the work shows how a complex optimization problem is handled by a genetic algorithm and converges very quickly.

In the present work, optimization of machining parameters (feed and speed) has been done with GA (genetic algorithm) by using the data from J. S. Agapiou [6]. Hence, in the 3rd chapter, basics of GA are explained and in the 4th chapter, mathematical formulation of the single pass machining optimization problem has been done.

CHAPTER 3

GENETIC ALGORITHM

Machining optimization problem requires a robust search method (Robustness means numerical stability and ability to find a solution for a wide range of algorithmic parameters), which runs well in complex situations. The genetic algorithm **(GA)** approach has been selected firstly because their behavior is robust and so far efficient and secondly more and more attention is being is drawn to GA in a variety of disciplines. Evolution program started in the sixties when a group of biologists used digital computers to simulate genetic systems.

But it was John Holland who, in 1975, published **'Adaptation in natural and artificial systems'**, laying down the two main principles for current GAs: the ability of simple representation (bit strings) to encode complicated structures. In his schemata theory he explains, how with proper control, there is rapid improvement in the bit strings as also occurs in animal population. Thereafter a number of his students and other researchers have contributed to developing this field. An extensive list of GA-related papers is referenced in **Goldberg [13]**.

3.1 BASIC GA OPERATIONS

Let us consider the following minimization problem:

$$\textbf{Minimize } f(x)$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, \qquad i=1,2,\ldots\ldots\ldots\ldots\ldots\ldots,N.$$

3.1.1 CODING

In order to use GAs to solve the above problem, variables $\mathbf{x_i}$ are first coded in some string structure. It is important to mention here that the coding of variables is not absolutely necessary. There exist some studies where GAs is directly used on the variables **(Real-Coded GAs).** The type of coding used is a critical decision, which is made o the basis of the nature of the problem.

As an example, if four bits are used to code each variables in a two-variable function optimization problem, the strings **(0000 0000)** and **(1111 1111)** would represent the points

$$\mathbf{(x_1^{(L)}, x_2^{(L)})} \qquad \mathbf{(x_1^{(U)}, x_2^{(U)})}$$

respectively, because the sub-strings (0000) and (1111) have the minimum and the maximum decoded values. Any other eight-bit string can be found to represent a point in the search space according to a fixed mapping rule. Usually, the following linear mapping rule is used:

$$\mathbf{x_i = x_i^{(L)} + [x_i^{(U)} - x_i^{(L)} /(2^{li}-1)] * \text{Decoded values of } (S_i)} \qquad \mathbf{(1)}$$

In the above equation, the variable $\mathbf{x_i}$ is coded in a sub-string S_i of length I_i. The decoded value of binary sub-string S_i is calculated as $\sum 2^j b_j$ where b_j can be 0 or 1 as obtained from the string. Here, j varies from 1 to I_i with four bits to code each variable; there is only 2^4 or 16 distinct sub-strings possible, because each bit-position can take a value either 0 or 1. The accuracy that can be obtained with a four bit coding only approximately 1/16th of the search space. But as the string length increased by one, the obtainable accuracy increased 1/32nd of the search space.

The length of a sub-string representation depends upon the desired accuracy in that variable.
Generalizing this concept, we may say that I_i–bit coding for a variable, the obtainable accuracy in that variable is approximately

$$\mathbf{x_i = (x_i^{(U)} - x_i^{(L)}) / (2^{li} - 1)}.$$

Once the coding of the variable has been done, the corresponding point $x = (x_{1,}\ x_{2,}\ x_{3,}\ldots\ldots\ldots\ldots.,\ x_N)$ can be found using equation (1). Thereafter, the function value at the point x can also be calculated by substituting x in the given objective function f(x).

3.1.2 FITNESS FUNCTION

GAs mimics the 'survival-of-the-fittest' principle of nature to make a search process. Therefore, GAs are naturally suitable for solving maximization problems. Minimization problems are usually transformed into maximization problems by some suitable transformation. In general, a fitness function **F(x)** is first derived from the objective function **f(x)** and used in successive genetic operations. Certain genetic operators require that the fitness function to be non-negative, although certain operators do not have this requirement.

For maximization problems the fitness function can be considered to be the same as the objective function or **F(x) = f(x)**. For minimization problem, the fitness function is an equivalent maximization problem chosen such that optimum point remains unchanged. A number of such transformations are possible .The following function is often used for that purpose:

$$F(x) = 1/(1+f(x))$$

This transformation does not alter the location of the maximum, but converts a minimization problem to an equivalent maximization problem. The fitness function value of a string is known as the string's **fitness.**

3.1.3 GENRATION

The operation of GAs begins with a population of random strings representing machining or decision variables. Thereafter, each string is evaluated to find the fitness value. The population is then operated by three main operators- **reproduction, crossover,** and **mutation** – to create a new population of points. The new population is further evaluated and tested for termination. If the termination criterion like the maximum generation or predefined fitness value is not met, the population is iteratively operated by the above operators and evaluated. This procedure is continued until the termination criterion is met. One cycle of these operations and the subsequent evaluation procedure is known as a **generation in GA's terminology.**

3.1.4 GA OPERATORS

Reproduction is usually the first operator applied on a population. Reproduction selects good strings in a population and forms a mating pool. That is why the reproduction operator is sometimes known as the **selection operator**. There exists a number of reproduction operators in GA literature, but the essential idea in all of them is that the above-average strings are picked from the current population and their multiple copies are inserted in the mating pool in a probabilistic manner. The commonly used reproduction operator is the **proportionate reproduction operator** where a string is selected for the mating pool with a probability proportional to the fitness **(F_i).** Thus the i-th string in the population is selected with a probability proportional to **F_i.** Since the population size is usually kept fixed in a simple GA, the sum of the probability of the strings being selected for the mating pool must be one therefore, the probability for selecting the i-th string is

$$\mathbf{p_i = F_i / \sum F_j}$$

Where i varies from 1 to n, n being the population size and j also varies from 1 to n in the summation. One way to implement this selection scheme is to imagine a roulette-wheel with its circumference marked for each string proportionate to the string's fitness. The roulette-wheel is spun n times, each time selecting an instance of the string chosen by the roulette-wheel pointer.

Fig.3.1 shows a roulette-wheel for five individual having different fitness value. Since the third individual has the higher fitness value than any other, it is expected that the roulette-wheel selection will choose the third individual more than any other individual. This roulette-wheel selection scheme can be simulated easily. Using the fitness value **F_i** ,

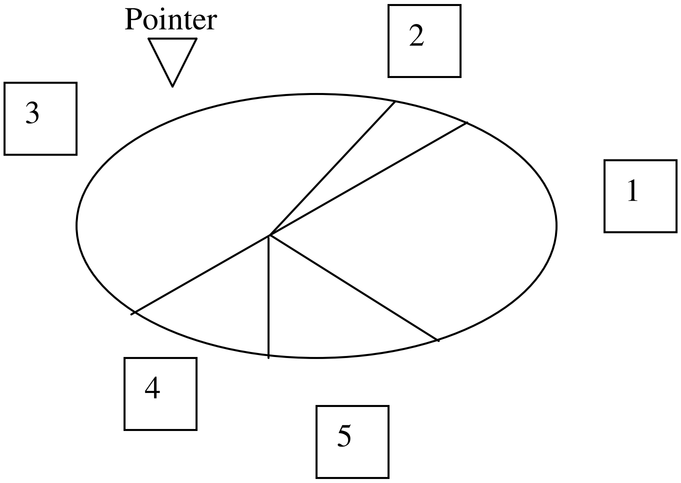


Figure.3.1 A Roulette-wheel

TABLE 3.1

Point	Fitness
1	25.0
2	5.0
3	42.0
4	10.0
5	18.0

of all strings, the probability of selecting a string $\mathbf{p_i}$ can be calculated. Thereafter the cumulative probability $\mathbf{(P_i)}$ of each string being copied can be calculated by adding the individual probability from the top of the list. Thus, the bottom most string should have cumulative probability $\mathbf{(P_n)}$ equal to one.

The roulette-wheel concept can be simulated by realizing that the i-th string in the population represents the cumulative probability values from P_{i-1} to P_i. The first string represents the cumulative values from 0 to P_i. Thus, the cumulative probability of any string lies between 0 to1. In order to choose n strings, n random numbers between 0 and 1 are created at random. Thus a string that represents the chosen random number in the cumulative range (calculated from the fitness values) for the string is copied to the mating pool. This way, the string with a higher fitness value will represent a larger range in the cumulative probability values and therefore has a higher probability values and therefore has a higher probability of being copied into the mating pool. On the other hand, a string with a smaller fitness value represents a smaller range in cumulative probability values and has a smaller probability of being copied into the mating pool.

In **reproduction**, good strings in a population are probabilistically assigned a larger number of copies and a mating pool is formed. It is important to note that no new strings are formed in the reproduction phase.

In the **crossover operator**, new strings are created by exchanging information among strings of the mating pool. Many crossover operators exist in the GA literature. In most of the crossover operators, two strings are picked from the mating pool at random and some portions of the strings are exchanged between the strings. A single point crossover operator is performed by randomly choosing site along the string and by exchanging all bits on the right side of the crossing site as shown:

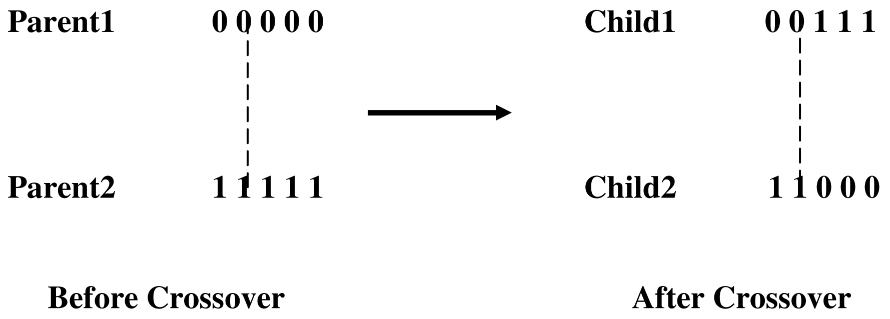


Figure 3.2

The two stings participating on the crossover operation are known as ‘parent strings’ and the resulting strings are known as ‘children strings’. Here, the dotted line represents the crossover site. It is intuitive that good sub strings from parent strings can be combined to from a better child string, if an appropriate site is chosen. Since the knowledge of appropriate site is usually not known before hand, a random site is often chosen. With a random site the children string produce may or may not have a combination of good sub string from parent string, depending on whether or not the crossing falls in the appropriate place. But we do not worry about this too much, because if good strings are created by crossover there will be more copy of them in the next mating pool generated by reproduction operator. But if good strings are not created by crossover, they will not survive to long, because reproduction is selected against those strings in subsequent generations.

It is clear from the above discussion that the effect of crossover may be beneficial. Thus, in order to preserve some of the good strings that are already present in the mating pool, not all strings in the mating pool are used in the crossover. When a crossover probability of p_c is used, only 100p_c percent strings in the population are used in the crossover population and 100(1-p_c) percent of the population remains as they are in the current population.

A crossover operator is mainly responsible for the search of the new string taking into account the character string in the mating pool.

The **mutation operator** changes 1 to 0 and vise-versa with a small mutation probability, p_m. The bit-wise mutation is performed bit by bit by flipping a coin with a probability pm is simulated as follows:

A number between 0 and 1 is chosen at random. If the random number is smaller than p_m , the outcome of coin flipping is true; otherwise the outcome is false.

If at any bit the outcome is true then the bit is altered; otherwise the bit is kept unchanged. The need for mutation is to create a point in the neighborhood of the current point, thereby achieving a local search around the current solution. The mutation is also used to maintain diversity in the population. For example, consider the following population having four eight-bit strings:

0110 1011
0011 1101
0001 0110
0111 1100

It is to be noticed in the above population that all four strings have a 0 in the left most bit position.

If the true optimum solution requires 1 in that position, them neither reproduction nor crossover operator described above will be able to create 1 in that position. Then mutation operator is the only rescue.

3.2 DIFFERENCE BETWEEN GAs AND TRADITIONAL METHODS

As seen from the above description of the working principles of GAs, they are radically different from most of the traditional optimization methods. However, the fundamental differences are described as:

(i) DISCRETIZATION OF SEARCH SPACE

GAs work with a string coding of variables instead of the variables. The advantage of working with a coding of variables is that the coding discretizes the search space; even though the function may be continuous function can be handled with no extra cost. This allows GAs to be applied to a wide variety of problems.

(ii) NO GRADIENT INFORMATION REQUIRED

In GA operators or in their working principles no information about the gradient of a function is required. Infact, GAs do not require any auxiliary information except the function values.

(iii) PROBABILISTIC IN NATURE

One other difference is the operation of GAs is the use of probabilities in their operators. None of the genetic operators work deterministically. In the reproduction operator, simulation of the roulette- wheel selection scheme is used to assign the true number of copies. In the crossover operator, even though good strings (obtained from the mating pool) are crossed, strings to be crossed are created at random and so is the case with the crossing sites. In the mutation operator, a random bit is suddenly altered. The action of these operators may appear to be native, but careful studies may provide some interesting insights about this type of search.

(iv) ROBUSTNESS

The basic problem with most of the traditional methods is that they use fixed transition rules to move one point to another. For instance, in the steepest descent method, the search direction is always calculated as the negative of the gradient at any point, because in that direction the reduction in the function value is maximum. In trying to solve a multi-modal problem with many local optimum points (interestingly, many real world engineering optimization problems including those of machining optimization are likely to be multimodal), search procedures may easily get trapped in one of the local optimum points.
Consider the bimodal function shown in figure 3.4. The objective function has one local minimum and one global minimum. If the initial point is chosen to be a point in the local basin (point $x^{(t)}$ in the figure) the steepest descent algorithm will eventually find the local optimum point.

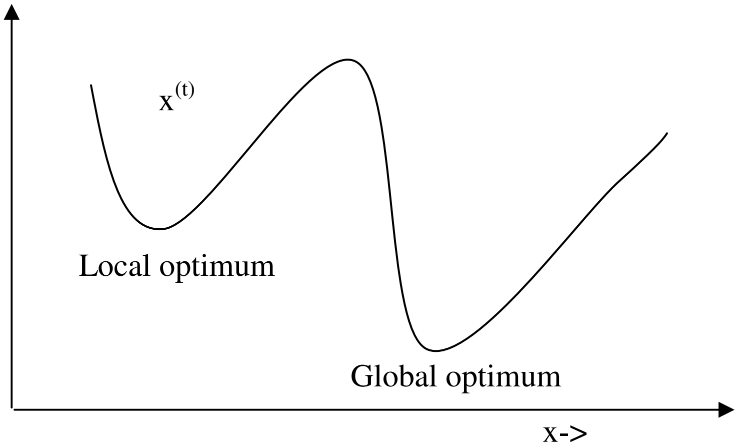


Figure 3.4

Since the transition rules are rigid, there is no escape from these local optima. The only way to solve the above problem to global optimality is to have a starting point in the global basin. Since this information is usually not known in any problem, the steepest –descent method (and for that matter most traditional methods) fails to locate the global optimum. However, these traditional methods can be best applied to a special class of problems suitable for those methods. For example, the gradient search methods will outperform, almost any algorithm in solving continuous, unimodal problems, but they are not suitable for multi-modal problem.

Thus in general, traditional methods are not robust. A robust algorithm can be designed in such a way that it uses the steepest descent direction most of the time, but also uses the steepest descent direction (or any other direction) with some probability. Such a mixed strategy may require more number of function evaluations to solve continuous, unimodal problems, because of the extra computations involved in trying with non-descent direction. But this strategy may be able to solve complex multimode problem to global optimality. In the multi-model problem, shown in the above figure, the mixed strategy may take the point $x^{(t)}$ into the global basin (when tried with non-descent directions) and finally find the global optimum point.

GA use similar search strategies by using probability in all their operators. Since an initial random population is used, to start with, the search can proceed in any direction and no major decisions are made in beginning. Later on, when the population begins to converge in some bit positions, the search directions narrow and a near optimal solution is achieved. This nature of narrowing the search space as the search progress is adaptive and is a unique characteristic of genetic algorithms.

3.3 BASIC ALGORITHM

Step 1 Choose a coding to represent problem parameters, a selection operator, a crossover operator, and a mutation operator. Choose population size, n, crossover probability, p_m. Initialize a random population of strings of size 1. Choose a maximum allowable generation number $\mathbf{t_{max}}$. Set $t = 0$.

Step 2 Evaluate each string in the population.

Step 3 If $\mathbf{t > t_{max}}$ or other termination criteria is satisfied, Terminate.

Step 4 Perform **reproduction** on the population.

Step 5 Perform **crossover** on the random pairs of strings.

Step 6 Perform **mutation** on every string.

Step 7 Evaluate strings in the new population. Set $\mathbf{t = t+1}$ and go to **step 3**.

The algorithm is straightforward with repeated application of three operators **(Step 4 to 7)** to a population of points (strings).

3.4 GAs FOR CONSTRAINED OPTIMIZATION

Genetic algorithms have also been used to solve constrained optimization problems. Although different methods to handle constraints have been suggested, penalty function methods have been mostly used. In the penalty function method, a penalty term corresponding to the constraint violation is added to the objective function. In most cases, a **bracket operator penalty term** is used. In a constrained minimization problem, the objective function **f(x)** is replaced by the penalized function:

$$\mathbf{P(x) = f(x) + \sum u_j < g_j(x) >^2}$$

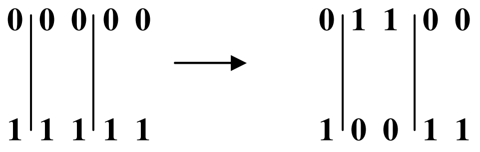
where u_j is penalty coefficient, which is usually kept constant throughout GA simulation. $<g_j(x)> = g_j(x)$, when $g_j(x)$ is negative; zero otherwise.

3.5 OTHER GA OPERATORS

There exist a number of variations to GA operators. In most cases, the variants are developed to suit particular applications.

In trying to solve problems with many variables, the single point crossover operator described earlier may not provide adequate search. Moreover, the single-point crossover operator has some bias of exchange for the right-most bits. They have a higher probability of getting exchanged than the left-most bits in the string. Thus, if ten variables are coded left to right with the first variable being at the left-most position and the tenth variable at the right-most position, the effective search on the tenth variable is more compared to the first variable.

In order to overcome this difficulty, a **multi-point crossover** is often used. The operation of a two-point crossover operator is shown below:



Two random sites are chosen along the string length and bits inside the cross-sties are swapped between the parents.

3.6 REAL CODDED GAs

GAs has also been developed to work with continuous variables (instead of discrete variables). In that GAs, binary strings are not used. Instead, the variables are directly used. Once a population of random sets of points is created, a reproduction operator (roulette-wheel or other selection operators) can be used to select good strings in the population. In order to create new strings, the crossover the mutation operators described earlier cannot be used efficiently. Even though simple single-point crossover can be used on those points by forcing the cross-sites to fall only on the variable boundaries, the search is nor adequate. This type of GA has been used in earlier studies by Wright. Recently, new and efficient crossover operators have been designed so that search along variables is also possible.

Let us consider $x_i^{(j)}$ and $x_i^{(k)}$ values of design variables x in two parent strings j & k . The crossover between these two values may produce the following new value:

$$\mathbf{x_i^{(new)} = (1-\lambda)\, x_i^{(j)} + \lambda\, x_i^{(k)} \qquad 0<\lambda<1 \qquad (II)}$$

The parameter γ is a random between zero & one. The above equation calculates a new value bracketing $x_i^{(j)}$ and $x_i^{(k)}$. This calculation is performed for each variable in the string. This crossover has a uniform probability of creating a point inside the region bounded by two parents. An extension to this crossover is also suggested to create points outside the range bounded by the parents. Eshelman and Schaffer have suggested to blend crossover operator (BLX-α), in which a new point is created uniformly at random from a larger range extending an amount $\alpha \mid x_i^{(j)} - x_i^{(k)} \mid$ on either side of the region bounded by two parents. The crossover operation depicted in Equation (II) can also be used to achieve BLX-α by varying λ in the range $(-\alpha, 1+\alpha)$. In a number of test problems, Eshleman and Schaffer have observed that α=0.5 provides good results.

One interesting feature of this type of crossover operator is that the created point depends on the location of both parents. If both parents are close to each other, the new point will also be close to the parents. On the other hand, if parents are far from each other, the search is more like a random search.

The random search feature of these crossover operators can be relaxed by using a distribution other than random distribution between parents. A recent study using a polynomial probability distribution with a bias towards near-parent points has been found to perform better than BLX-0.5 in a number of test problems. More studies in this direction are necessary to investigate the efficacy of real-coded GAs.

CHAPTER 4

PROPOSED METHODLOGY

4 .1 FORMULATION OF THE OPTIMIZATION PROBLEM

The formulation of the optimization problem requires the knowledge of some mathematical equations, which represent the economical and physical parameters for the machining process and the whole machine-tool system. The mathematical formulation used here is:

4.1.1 FORMULATION OF THE OBJECTIVE FUNCTION

Both the production cost and total production time are considered in the formulation of the objective function. The production cost per component for a machining operation is comprised of the sum of the costs for tooling, machining, tool changing time, handling time and quick return time:

$$\text{Time of machining}, \ T_m = \frac{L}{fN}$$

$$V = \frac{\pi DN}{1000}$$

$$\therefore \quad T_m = \frac{\pi DL}{1000fV}$$

$$\text{Cost of cutting action} = C_o T_m$$

$$\text{Cost of tool changing} = C_o \frac{T_m}{T} T_{cs}$$

Cost of tooling = $C_t \frac{T_m}{T}$

Handling cost = $C_o T_h$

Quick return time = T_r

Total cost of Production = $C_o T_m + C_o \frac{T_m T_{cs}}{T} + C_t \frac{T_m}{T} + C_o T_h + C_o T_r$

Where V and *f* are, respectively, the cutting speed and feed, while D and L are the outside diameter and length of the part, respectively. C_o is the operating cost in \$/min, C_t is the tool cost per cutting edge in \$/edge, T_{cs} is the tool change time in min/edge, T_h is the loading and unloading time in min/piece, **T** is tool life in min/edge.

Taylor's expanded tool life equation is

$$V * f^{a} * d^{b} * T^{c} = K$$

Where a,b,c and K are empirical constants and d is the depth of cut.

Thus, the total cost of production as an objective function can be written as:

$$C_u = C_o * V^{-1} * f^{-1} + A * V^{(1/c-1)} * f^{(a/c-1)} * d^{(b/c)} * K^{(-1/c)} * (C_o * T_{cs} + C_t) + C_o * (T_h + T_r) \quad \text{………… } \mathbf{(1)}$$

Where $A = \pi * D * L / 1000$.

The total time required to produce a part is the sum of the times necessary for machining, work piece handling, tool changing and tool quick return and thus can be written as :

Total time $= T_u = T_m + T_h + \frac{T_m T_s}{T} + T_r$ ………**(2)**

For the multi-objective problem, the objective function consists of the sum of the production cost and the production time along with different weight coefficients for each criterion:

$$\mu(V, f, d) = w_1 * C_u + w_2 * \lambda * T_u$$

Where w_1 and w_2 are the weight coefficients representing the relative importance of the production cost and the total production time criteria, respectively.

These weight coefficients satisfy the following condition:

$$w_1 + w_2 = 1, \qquad 0 \le w_1 \le 1 \text{ and } 0 \le w_2 \le 1$$

The objective function would provide the optimum of the individual production cost or time criteria by setting the corresponding weight coefficient equal to zero. The optimum multi-objective function is normalized through the use of a constant multiplier, λ:

$$\lambda = C_{umin} / T_{umin}$$

where, C_{umin} and T_{umin} are the minimum production cost and the minimum production time, respectively, under the defined process constraints. The values are determined by putting the optimum values of the cutting speed V_{opc} and V_{opt} respectively, in equations (1) and (2), for the highest possible feed f_h. For V_{opc} and V_{opt}, we have:

$$V_{opc} = K / f_h{}^{a} * d^{b} [(1/c-1) * (t_{cs} + c_t / c_o)]^{c}$$
$$V_{opt} = K / f_h{}^{a} * d^{b} [(1/c-1) * (t_{cs})]^{c}$$

4.1.2 CONSTRAINTS

Practical limitations of the actual cutting conditions always exist for the optimization of the objective function.

1. FEED

The maximum allowable feed has a pronounced effect on both maximum spindle speed and production rate. Feed changes have a more significant impact on tool life than depth of cut changes. The system energy requirement reduces with feed since the optimum speed becomes lower. Therefore, the largest possible feed consistent with the available machine power and surface finish is desirable in order for the machine to be fully utilized. It is often possible to obtain much higher metal removal rates without reducing tool life by increasing the feed and decreasing the cutting speed. This technique is particularly useful for roughing cuts where the maximum feed is dependent on the maximum tool force the machine tool is able to withstand.

In general, the maximum feed in a roughing operation is limited by the maximum force that the cutting tool, machine tool, workpiece, and the fixture are able to withstand. The maximum feed in a finish operation is limited by the surface finish requirement and can be often predicted to a certain degree based on the surface finish and tool nose radius.

2. SPEED

Cutting speed has a greater effect on the tool life than either depth of cut or feed. When compared, the cutting speed has only a secondary effect on the chip-breaking. There are certain combinations of speed and feed, which are preferred for the easy chip-removal and are mainly dependent on the type of the tool and the workpiece material.

The following physical constraints are considered in the formulation of the objective model :

- **Feed and Speed Limitations :** Maximum and minimum permissible feed rate and cutting speed::

  $f_{min} \leq f \leq f_{max}$,

  $V_{min} \leq V \leq V_{max}$,

- **Power Limitation :** The power consumption allowed on the steel material is given as a function of feed, speed and depth of cut::

  $0.0373 * V^{0.9} * f^{0.78} * d^{0.75} \leq HP_{max}$,

- **Surface Roughness Limitation :** The surface roughness allowed for the steel material is given as a function of feed, speed and depth of cut::

  $14785 * V^{-0.52} * f^{1.004} * d^{0.25} \leq SR_{max}$,

- **Temperature Constraint :** The temperature constraint given by inequality is ::

  $74.96 * V^{0.4} * f^{0.2} * d^{0.1025} - 17.8 \leq T_{max}$,

- **Cutting Force Constraint :** Constraint for the maximum cutting force allowed is::

  $844 * V^{-0.103} * f^{0.725} * d^{0.75} \leq F_{max}$

4.2 GA IMPLEMENTATION

In the section, GA has been developed to obtain the optimal solution from the objective functions and various constraints that we have discussed in the problem formulation:

4.2.1 CHROMOSOME REPRESENTATION

The genes or process variables of the solution are coded first. The values for the string size, population size and the maximum number of generations to be used are found through the experimentation The values giving best results are found to be 30 for the length of each string, 60 for the population size and 240 for the maximum no. of generation.

4.2.2 INITIALIZATION

Initialization involves the formation of an initial population of chromosomes wherein each gene of a chromosome is obtained randomly by using the random number generator.

4.2.3 EVALUATION OF CHROMOSOMES

Each chromosome of the population is then evaluated to get its fitness function and the various constraints values. A penalty operator for each constraint is included in the fitness function so as to account the situation when constraint crosses its extreme limit. Thus for the objective function F(x), we have the fitness function [13]::

$$P(x) = F(x) + \mathbf{u_1 \langle t(x) \rangle^2 + u_2 \langle f(x) \rangle^2 + u_3 \langle p(x) \rangle^2 + u_4 \langle r(x) \rangle^2}$$

where u_j is penalty coefficient, whose value is calculated through experimentation. Here, t(x), f(x), p(x) and r(x) are the temperature, force, power and the surface roughness constraints. Whenever any constraint crosses its extreme limit, the square of its value multiplied with penalty coefficient gets added to objective function so as to reduce the value of the fitness function to zero. The values of penalty coefficients calculated through experiments are $u_1 = 100$, $u_2 = 100$, $u_3 = 1000$ and $u_4 = 1000$.

Then the statistics for the each chromosome, which involves the sum of fitness values, average, probability of selection is done.

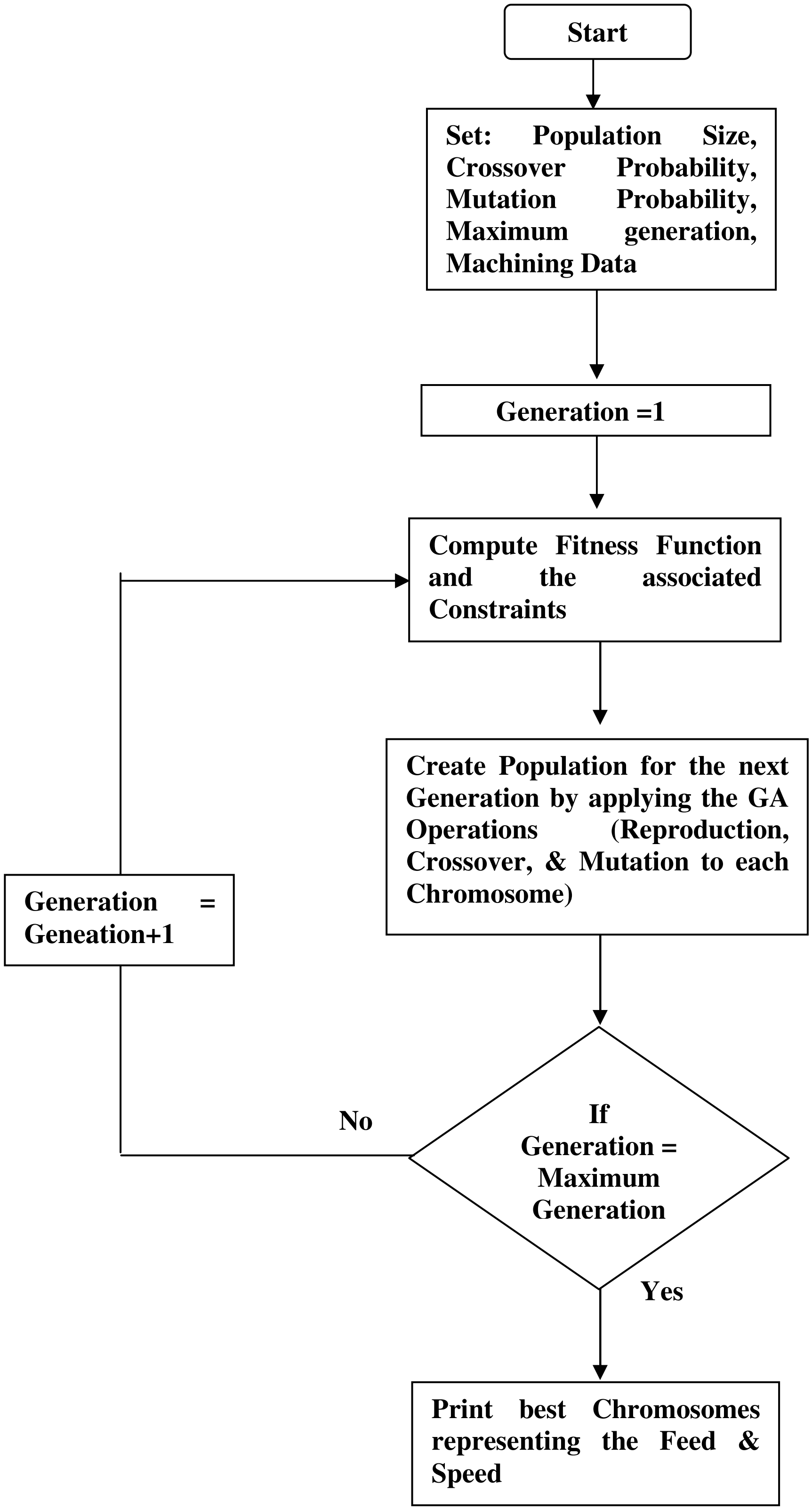
4.2.4 CROSSOVER AND MUTATION

A new population of offsprings is generated using crossover and mutation operators. The parents for crossover are selected from the previous population of chromosomes on the basis of the statistics and the selection operator in which the probability of selection of each parent is directly proportional to its fitness values. The crossover operator is applied on the mating sites of the parents. Then the mutation operator is applied so as to avoid the entrapment in the local optima. Then a complete new population is obtained.

4.2.5 TERMINATION

Unless the generation reaches its maximum value, the above procedure is repeated over and again to produce subsequent generations. The procedure gets terminated when (generation = maximum generation). Then we can pick the best solution from the latest solution.

4.3 FLOW CHART OF GA FOR MACHINING OPTIMIZATION PROBLEM



CHAPTER 5

RESULTS AND CONCLUSION

In this chapter, the various results related to the optimization of machining parameters have been discussed. The results are taken for two depth of cuts i.e. doc = 2.54 mm and doc = 5.08 mm, applied to the various objective functions. Here three objective functions i.e. the total production time, total production cost and combined of these two, has been taken.

Machining parameters used for the example are::

Parameter	**Value**	**Parameter**	**Value**
L	203 mm	a	0.29
D	152 mm	b	0.35
V_{min}	30 m/min.	c	0.25
V_{max}	200 m/min.	K	193.3
f_{min}	0.2540 mm/rev.	T_{cs}	0.50 min/edge.
f_{max}	0.7620 mm/rev.	T_r	0.13 min/pass.
SR_{max}	A	T_h	1.50 min/piece.
HP_{max}	B	C_o	0.10 $/min.
F_{max}	C	C_t	0.50 $/edge.
T_{max}	D		

Here the constraints surface roughness, power, force and temperature have given values A, B, C and D respectively. In the next sections, the various results will be discussed by taking the different values of these constraints.

In the section 5.1, the total production time is taken as an objective function with depth of cut, doc = 2.54 mm and the constraints force, surface roughness, temperature and power. Machining parameters used for the example are::

Parameter	**Value**
SR_{max}	8 μm
P_{max}	4 KW
F_{max}	600 N
T_{max}	400 deg C

The table 5.1(a), shows the values of randomly selected machining parameters, the associated fitness values and the various constraints at the starting generation. After applying the various GA operations, we get the table 5.1(b), that shows the values of above stated variables at the end generation.

It is clear from these tables that the value of fitness function becomes zero, when the constraints cross their extreme limits. These both tables are also graphically plotted in fig. 5.1(a) to fig. 5.1(d).

Fig. 5.1(e) and fig. 5.1(f), shows the variation of best fitness function and the objective function values with generation, respectively. These figures clearly show how the values of both these functions get improved with generation, after applying the GA operations. From fig. 5.1(f), the best value of the objective function i.e. the total production time is approximated to be 4.41001081 min. It is clear from table 5.1(b) that the constraints surface roughness = 7.18585 μm, force = 554.72638 N and temperature = 399.17020 deg C have their extreme limits for the calculated best objective function value.

From fig. 5.1(g), that we have from J. S. Agapiou [6], the value of objective function lies at point A (less than 5 min) that justifies our objective function value.

5.1 TOTAL PRODUCTION TIME AS AN OBJECTIVE FUNCTION with doc = 2.54mm

TABLE 5.1(a)

```
                   Accumulated Statistics for Generation ---> 0

*********************************************Calculated Constraints************
         FEED     SPEED    FITNESS    POWER   ROUGNESS   TEMP      FORCE
*******************************************************************************

  1)   0.265398 175.937546 0.000000  2.79767  1.90405  483.68732  384.44864
  2)   0.368651 164.810791 0.000000  3.40864  2.92479  503.94052  491.11554
  3)   0.707231 154.328079 0.000000  5.34063  6.21660  561.13177  792.88513
  4)   0.277501 113.571846 0.000000  1.95355  3.87308  406.91574  415.08380
  5)   0.353852  96.545815 0.000000  2.04021  6.32779  400.02536  503.28018
  6)   0.472962  33.475842 0.000000  0.98618 42.36076  272.06448  691.45142
  7)   0.711192 138.368027 0.000000  4.86201  7.37989  537.01575  804.95435
  8)   0.651704 190.695023 0.000000  6.06176  4.15165  602.04474  731.40137
  9)   0.387555 119.104492 0.000000  2.64588  5.03855  444.98053  526.28137
 10)   0.492019 110.423180 0.000000  2.97738  7.18337  453.13144  630.51129
 11)   0.665699 154.542709 0.000000  5.10076  5.83777  554.48468  758.74115
 12)   0.484704 160.232910 0.000000  4.11417  4.01815  527.12061  600.61646
 13)   0.494961  91.315140 0.000000  2.52109  9.64615  419.18689  645.54816
 14)   0.493255 179.400391 0.000000  4.61712  3.44399  554.31158  601.35675
 15)   0.491475 156.755234 0.000000  4.07760  4.21269  523.85925  608.03699
 16)   0.516609 155.135895 0.000000  4.19992  4.49947  527.02167  631.08917
 17)   0.601101  84.331436 0.000000  2.73089 13.23088  422.27063  749.20697
 18)   0.409431 152.891190 0.000000  3.45764  3.64249  499.24252  533.97498
 19)   0.451691 184.893066 0.000000  4.42934  3.01142  551.14954  562.45667
 20)   0.405126  78.410904 0.000010  1.88015  9.94484  377.20782  566.98322
 21)   0.587060  76.430313 0.000000  2.45384 15.00455  403.29477  743.85382
 22)   0.563392  67.814285 0.000000  2.13381 17.26774  380.33258  730.79004
 23)   0.597078 131.112030 0.000000  4.04126  6.71968  506.52472  712.97296
 24)   0.480172  69.253204 0.000000  1.91964 14.24570  371.05927  649.44745
 25)   0.720388  85.014328 0.000000  3.16795 15.67467  439.96994  853.58325
 26)   0.522766 174.428543 0.000000  4.71053  3.81023  554.52881  629.02026
 27)   0.418351 135.793015 0.000000  3.16028  4.45741  477.41769  548.93884
 28)   0.634586  69.288139 0.000000  2.38709 18.83339  393.44321  794.90350
 29)   0.344363 167.934296 0.000000  3.28725  2.65450  500.75650  466.54953
 30)   0.712110 138.465622 0.000000  4.87000  7.38155  537.31549  805.65033
 31)   0.729286  32.286057 0.000000  1.33816 69.13161  293.74640  949.96069
 32)   0.569894  82.386620 0.000000  2.56522 12.99407  413.55881  722.50745
 33)   0.641435  87.315018 0.000000  2.96410 13.39540  434.26746  782.56622
 34)   0.595076 150.749207 0.000000  4.57015  5.41693  536.25598  701.25433
 35)   0.274393  75.849586 0.392977  1.34655  7.07339  342.77292  428.89325
 36)   0.587354  76.946098 0.000000  2.46970 14.85941  404.47147  743.61719
 37)   0.707795  56.659851 0.000000  2.16875 28.53373  370.01880  878.10016
 38)   0.633645 166.084961 0.000000  5.23687  4.97945  565.42816  726.75287
 39)   0.417891 185.984482 0.000000  4.19076  2.76041  543.68805  531.30115
 40)   0.451644 113.705063 0.000000  2.85942  6.30464  450.59079  590.80573
 41)   0.546790 144.663055 0.000000  4.12245  5.29734  518.05121  662.28235
 42)   0.522768  98.583450 0.000000  2.81863  9.07047  437.73364  666.45245
 43)   0.281335 119.956047 0.000000  2.07419  3.61358  417.50220  416.91788
 44)   0.509273 136.066681 0.000000  3.69090  5.41384  497.69907  632.93280
 45)   0.580263 120.321571 0.000000  3.65823  7.44024  485.93192  704.46088
 46)   0.452371  83.005196 0.000000  2.15682 10.18841  395.32184  610.65503
 47)   0.709183  73.238640 0.000000  2.73648 19.35467  412.11722  856.77966
 48)   0.445151 166.812454 0.000000  3.99186  3.47012  526.61487  562.37225
 49)   0.396755 114.283096 0.000000  2.59636  5.49305  439.53491  537.55481
 50)   0.395401 190.531403 0.000000  4.10198  2.51714  542.90100  509.16626
 51)   0.516358 125.885933 0.000000  3.47869  6.17828  483.29285  644.36145
 52)   0.287945  31.923386 0.000001  0.64164 27.66502  239.73877  484.84283
 53)   0.529086  59.035137 0.000000  1.79343 20.01530  354.15176  708.12640
 54)   0.720278  95.229164 0.000000  3.50814 13.18954  461.21063  843.73486
 55)   0.260188  33.414845 0.000002  0.61774 23.31256  239.22202  448.41052
 56)   0.685605 197.917953 0.000000  6.52091  4.12848  617.74451  755.94000
 57)   0.459890 189.774918 0.000000  4.59853  2.94721  559.18365  568.33783
 58)   0.524591  37.749146 0.000000  1.19127 39.15896  292.70181  736.37189
 59)   0.461275 154.051361 0.000000  3.82051  4.05876  513.32312  581.73853
 60)   0.402880 158.939590 0.000000  3.53574  3.37873  505.63794  525.69720

max=0.392977 min=0.000000 avg=0.006550 sum=0.392992 no_mutation=0 no_cross=0

  FEED      0.274393
  SPEED    75.849586
  FITNESS   0.392977
```

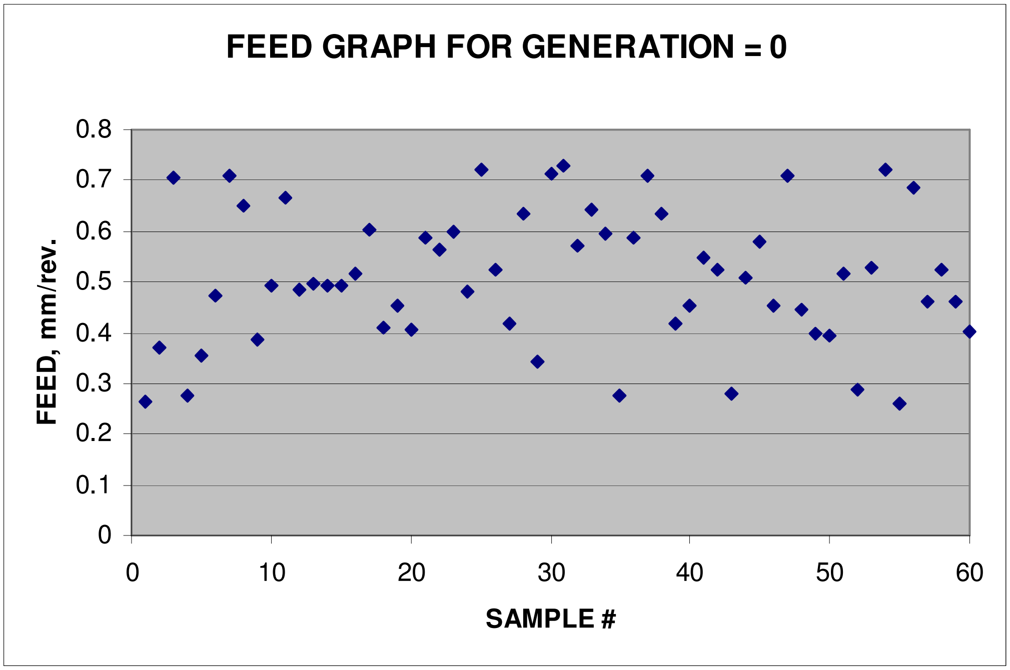


FIG. 5.1(a)

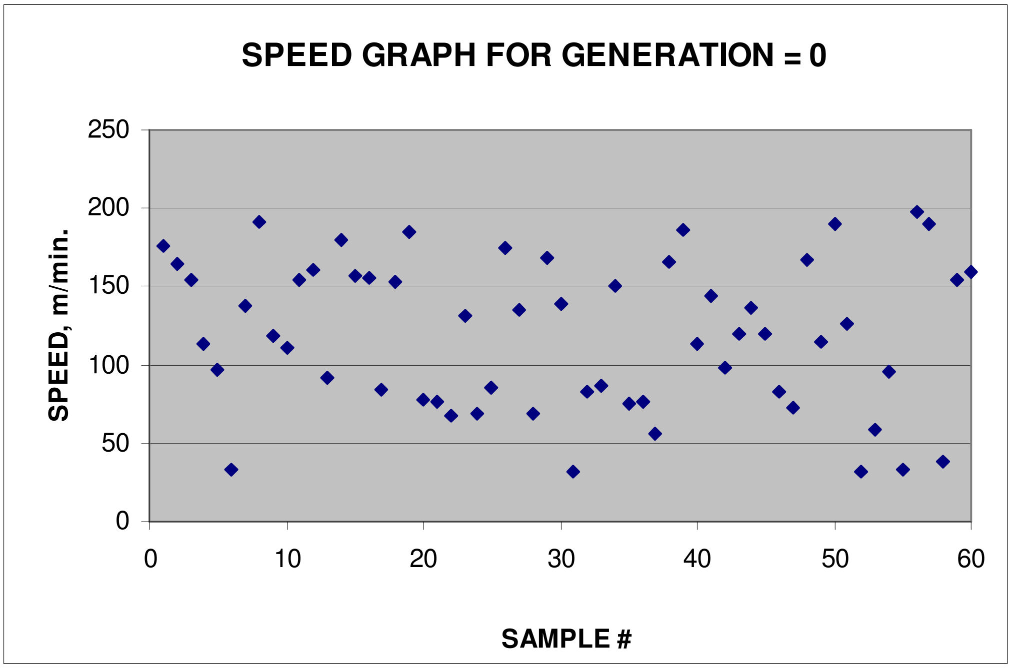


FIG. 5.1(b)

Fig. 5.1 Speed and feed graphs for starting generation with doc = 2.54mm

TABLE 5.1(b)

```
Accumulated Statistics for Generation ---> 240

*********************************************Calculated Constraints************
        FEED       SPEED    FITNESS   POWER   ROUGNESS   TEMP       FORCE
*******************************************************************************

  1)   0.375904  91.115112 0.474351  1.87830  6.90060  386.29996  498.64789
  2)   0.367274  89.691856 0.468479  1.72542  6.42043  376.67111  466.84549
  3)   0.345866  89.830795 0.459197  1.87392  6.99879  385.62106  500.56079
  4)   0.347882  91.115112 0.462216  1.91110  6.79283  389.07382  500.03418
  5)   0.375160  89.465988 0.471436  1.99395  7.53395  392.25635  529.14117
  6)   0.334824  89.039345 0.452653  2.08532  7.46118  398.76367  538.78033
  7)   0.375663  90.048653 0.472576  2.00774  7.47003  393.43286  529.30798
  8)   0.344063  90.631027 0.459676  1.88565  6.77258  387.31244  496.31622
  9)   0.347387  89.258080 0.458948  1.87392  6.99878  385.62128  500.56073
 10)   0.367275  84.601303 0.000016  1.86495  8.02911  381.48785  524.01385
 11)   0.335940  90.616554 0.455826  1.85057  6.61364  385.35538  487.80048
 12)   0.375160  89.465988 0.471436  1.99395  7.53395  392.25635  529.14117
 13)   0.375774  90.710060 0.473665  2.02147  7.38958  394.66266  529.02850
 14)   0.347387  46.763103 0.000003  1.04733 18.69620  293.70218  534.43652
 15)   0.333830  89.472473 0.452904  1.82056  6.70010  382.80646  486.20270
 16)   0.376656  90.658424 0.473947  2.02413  7.41340  394.76205  529.95874
 17)   0.336436  89.604828 0.454383  1.83407  6.73746  383.66730  488.87833
 18)   0.335603  81.067444 0.438477  1.67279  7.82551  367.71472  492.97583
 19)   0.268090  79.288788 0.396434  1.37621  6.45984  347.52383  419.83749
 20)   0.392069  91.692383 0.481686  2.49519  7.18585  399.17020  554.72638
 21)   0.375904  89.656906 0.472051  2.00087  7.52453  392.76880  529.78735
 22)   0.333830  68.222473 0.000010  1.42632 10.11764  341.62900  499.74280
 23)   0.279904  89.288353 0.423411  1.58386  5.63143  368.61752  427.98999
 24)   0.347386  83.945450 0.449641  1.77323  7.68298  375.83936  503.68146
 25)   0.344163  92.311913 0.462420  1.91753  6.58793  390.32486  495.49716
 26)   0.448093  91.747498 0.000000  2.34281  8.66683  411.38889  600.34229
 27)   0.327621  90.045982 0.450828  1.80444  6.51145  382.32651  479.31985
 28)   0.439274  91.691132 0.000000  2.30549  8.50352  409.58093  591.78937
 29)   0.379136  83.945450 0.000014  1.89842  8.38812  382.78534  536.65283
 30)   0.347820  83.945450 0.449845  1.77496  7.69262  375.93768  504.13766
 31)   0.343417  91.747498 0.461179  1.90375  6.63517  389.14835  495.02618
 32)   0.321093  89.025848 0.445811  1.75821  6.49266  378.90695  472.92178
 33)   0.343419  89.025200 0.456717  1.85285  6.94605  384.27518  496.54092
 34)   0.312226  90.051247 0.442975  1.73804  6.20371  378.50241  462.87909
 35)   0.347389  91.649178 0.462846  1.91905  6.72316  389.91031  499.22403
 36)   0.376620  89.465828 0.472041  2.00000  7.56341  392.57480  530.63373
 37)   0.347387  91.115112 0.461990  1.90897  6.78311  388.95786  499.51767
 38)   0.343402 172.426010 0.000000  3.35894  2.54296  505.96777  464.36221
 39)   0.279904  78.715363 0.402719  1.41402  6.82050  349.61990  433.48917
 40)   0.375636  89.049835 0.470963  1.98757  7.59721  391.59631  529.87872
 41)   0.347847  42.509823 0.000002  0.96218 21.64111  282.12344  540.14203
 42)   0.371199  83.903946 0.000015  1.86651  8.21799  381.01489  528.51007
 43)   0.367275  80.033424 0.000013  1.77407  8.73592  372.72061  526.96906
 44)   0.312660  91.628418 0.445860  1.76733  6.05057  381.37521  462.53131
 45)   0.375665  90.712715 0.473625  2.02107  7.38711  394.64368  528.91620
 46)   0.375664  90.048653 0.472576  2.00774  7.47003  393.43286  529.30804
 47)   0.375160  89.548996 0.471570  1.99562  7.52334  392.40848  529.09143
 48)   0.343404  87.426003 0.453981  1.82280  7.13977  381.36682  497.43732
 49)   0.343417  90.624458 0.459365  1.88277  6.76055  387.14847  495.64413
 50)   0.375665  89.913788 0.472363  2.00504  7.48711  393.18680  529.39038
 51)   0.329031  91.747498 0.454334  1.84125  6.35612  385.68008  479.90286
 52)   0.375266  89.465988 0.471481  1.99439  7.53609  392.27954  529.24957
 53)   0.347843 112.932098 0.000000  2.31819  4.90112  425.54379  489.23764
 54)   0.343417 110.952232 0.000000  2.25891  4.97037  421.29248  485.58658
 55)   0.347389  91.649178 0.462846  1.91905  6.72316  389.91031  499.22403
 56)   0.375160  89.465988 0.471436  1.99395  7.53395  392.25635  529.14117
 57)   0.376620  89.465828 0.472041  2.00000  7.56341  392.57480  530.63373
 58)   0.375160  89.465988 0.471436  1.99395  7.53395  392.25635  529.14117
 59)   0.347387  89.257919 0.458948  1.87392  6.99879  385.62097  500.56082
 60)   0.335603  81.067444 0.438477  1.67279  7.82550  367.71466  492.97552

max=0.481686 min=0.000 avg=0.366902 sum=22.014145 no_mutation=17185 no_cross=4308


 FEED     0.392069
 SPEED   91.692383
 FITNESS  0.481686
```

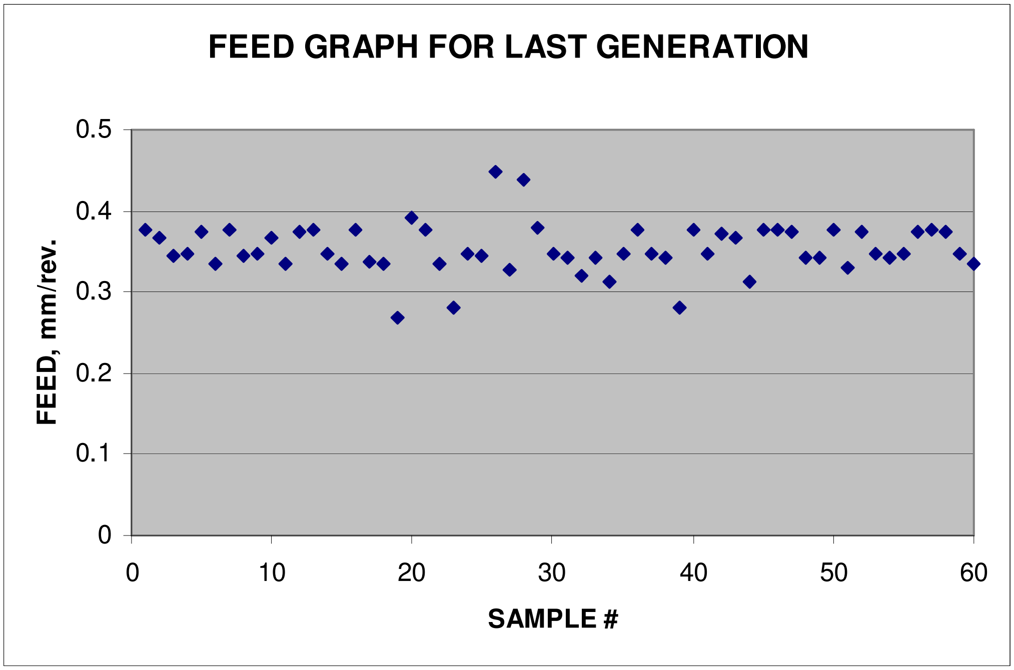


FIG. 5.1(c)

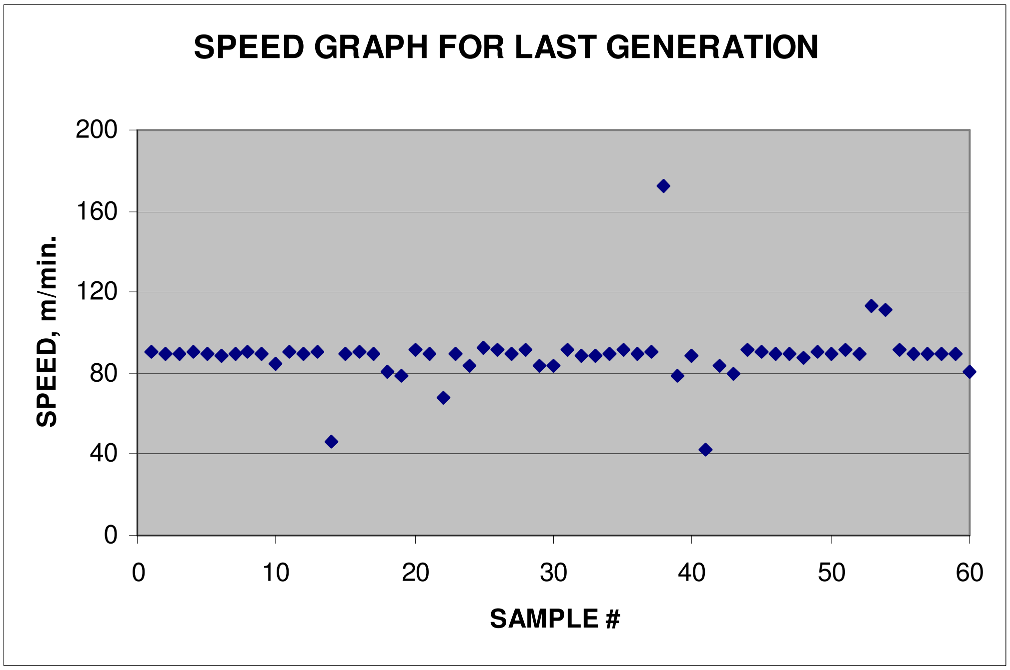


FIG. 5.1(d)

Fig. 5.1 Speed and feed graphs for last generation with doc = 2.54mm

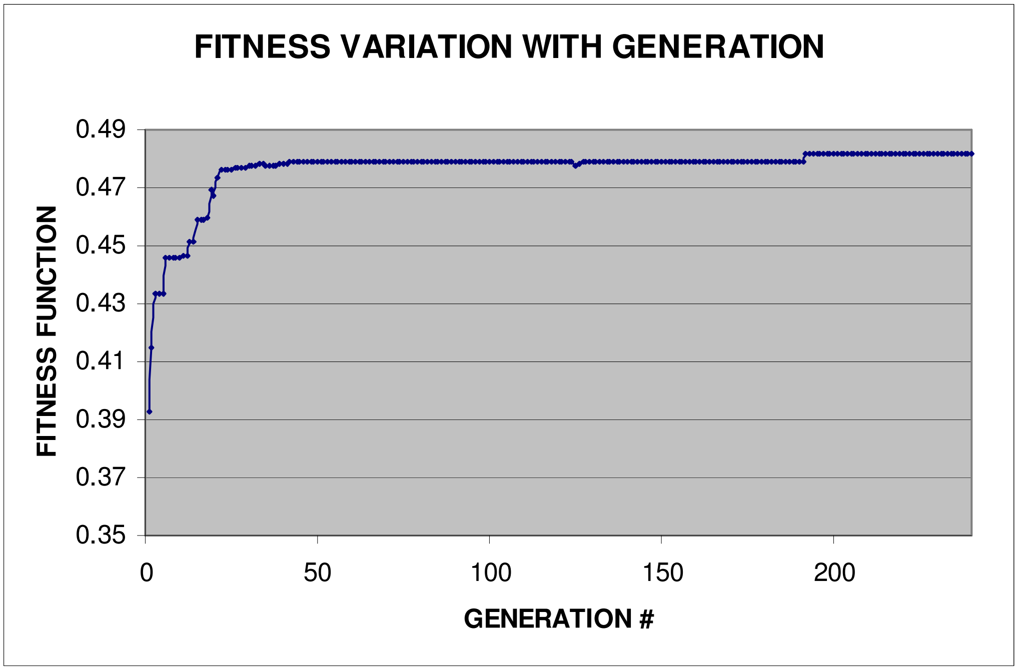


FIG. 5.1(e)

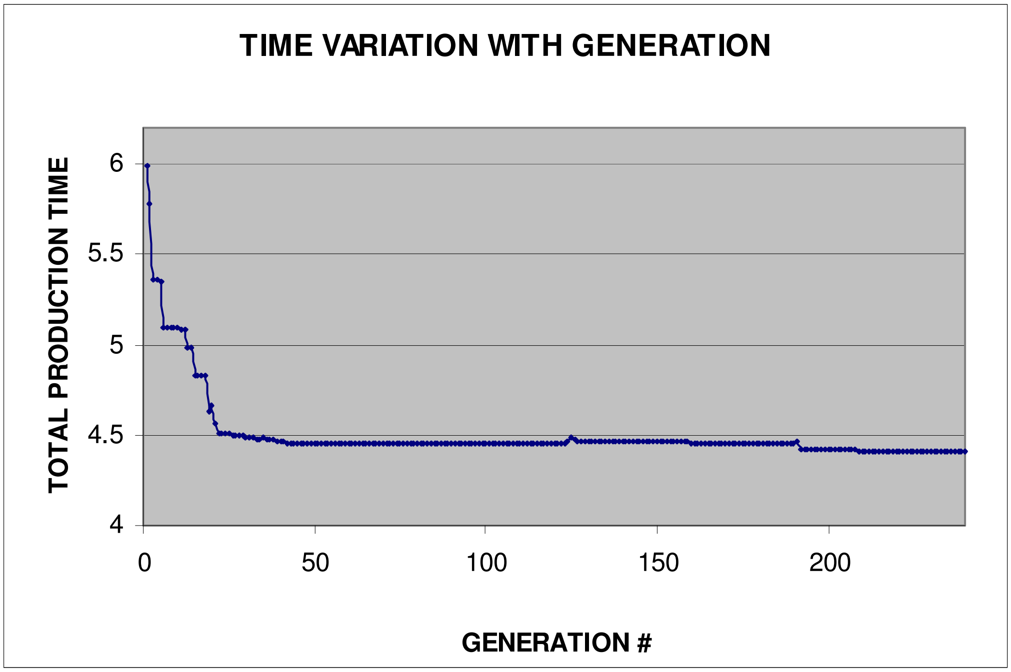


FIG. 5.1(f)

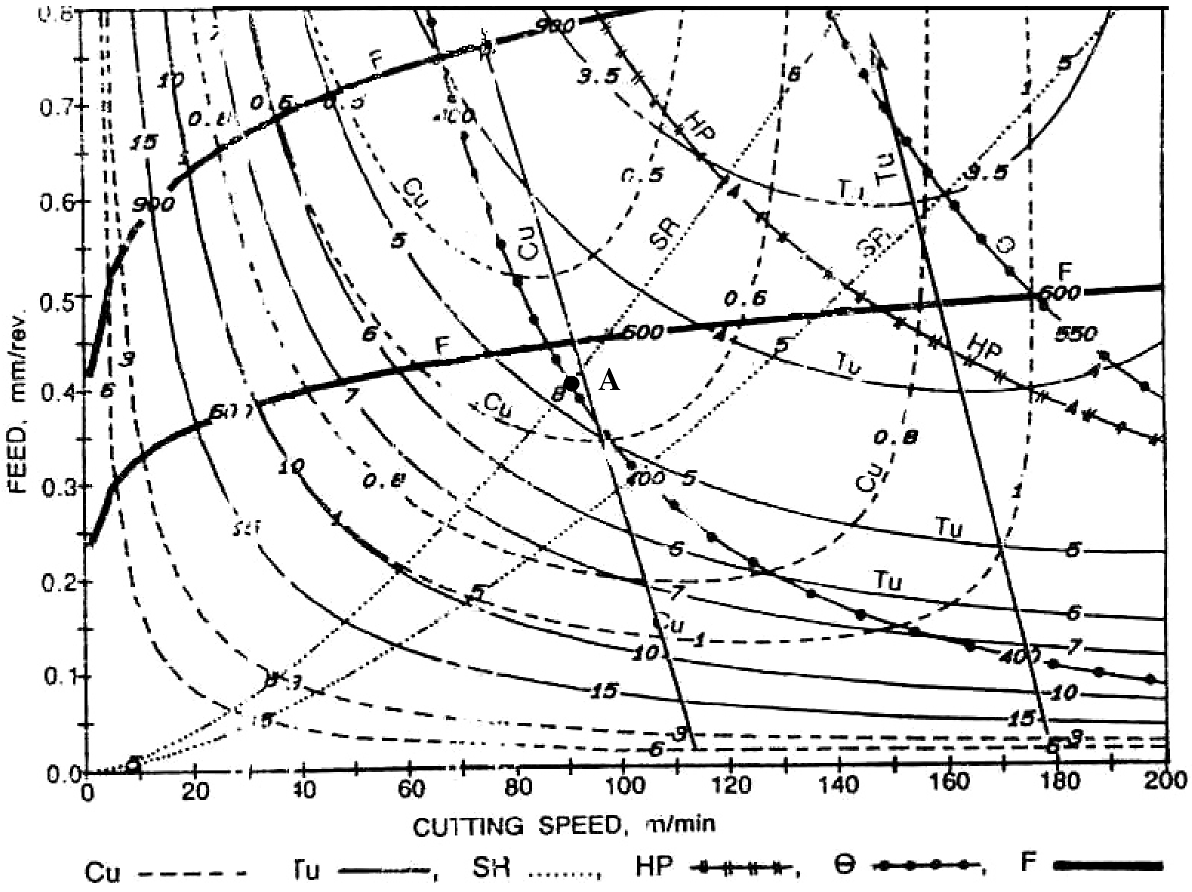


FIG. 5.1(g) Location of best result found by GA on the contours given by Agapiou [6], for doc = 2.54 mm and the constraints of power, force, surface roughness, and temperature.

In the section 5.2, the total production time is taken as an objective function with depth of cut, doc = 5.08 mm and the constraints force, surface roughness, temperature and power. Machining parameters used for the example are::

Parameter	Value
SR_{max}	8µm
HP_{max}	5KW
F_{max}	1200N
T_{max}	400 deg C

It is clear from tables 5.2(a) and 5.2(b) that the value of fitness function becomes zero, when the constraints cross their extreme limits. These both tables are also graphically plotted in fig. 5.2(a) to fig. 5.2(d).

Fig. 5.2(e) and fig. 5.2(f), shows the variation of best fitness function and the objective function values with generation, respectively. These figures clearly show how the values of both these functions get improved with generation, after applying the GA operations. From fig. 5.2(f), the best value of the objective function i.e. the total production time is approximated to be 5.44656 min. It is clear from table 5.2(b) that the constraints surface roughness = 7.97880 µm and temperature = 394.33133 deg C have their extreme limits for the calculated best objective function value.

From fig. 5.2(g), that we have from J. S. Agapiou [6], the value of objective function lies at point B (less than 6 min) that justifies our objective function value.

5.2 TOTAL PRODUCTION TIME AS AN OBJECTIVE FUNCTION with doc = 5.08mm

TABLE 5.2(a)

```
Accumulated Statistics for Generation ---> 0
*********************************************Calculated Constraints************
        FEED      SPEED    FITNESS    POWER   ROUGNESS    TEMP     FORCE
*******************************************************************************

  1)    0.601896 149.667526 0.000000  7.70457  6.58769  577.72589 1190.01672
  2)    0.527198 113.823242 0.000000  5.43068  8.74335  501.99533 1111.41150
  3)    0.270287 154.535706 0.000000  4.24675  2.80877  496.15115  663.84216
  4)    0.290274 141.421555 0.000000  4.14535  3.45274  485.36823  705.38983
  5)    0.314289 108.783211 0.000000  3.48280  5.57226  442.49451  767.36133
  6)    0.709576 108.464958 0.000000  6.55611 12.67798  523.27917 1385.28638
  7)    0.671191  46.700748 0.000000  2.94061 43.15869  364.18295 1449.10876
  8)    0.519633 195.971603 0.000000  8.75636  3.77323  626.31262 1040.92847
  9)    0.665176 155.502121 0.000000  8.62104  6.87189  599.11743 1274.51843
 10)    0.430016 159.317413 0.000000  6.26991  4.27431  553.08350  926.67468
 11)    0.704758  93.228828 0.000000  5.69080 15.84905  490.79654 1399.76123
 12)    0.395009  59.706654 0.000003  2.42594 17.44722  361.23190  962.43018
 13)    0.504574  74.656189 0.000000  3.59044 15.88415  417.46786 1123.61975
 14)    0.266873 147.868607 0.000000  4.04122  2.96542  485.88220  660.69763
 15)    0.284507 154.480118 0.000000  4.41859  2.95876  501.37384  689.00836
 16)    0.541538  39.574612 0.000000  2.14295 44.74839  324.68198 1261.25513
 17)    0.265586 159.109161 0.000000  4.30044  2.64006  500.36075  653.51825
 18)    0.415806  62.776356 0.000003  2.64154 17.02171  372.89856  993.84589
 19)    0.336536  59.330444 0.000004  2.12883 14.99841  348.35306  857.44293
 20)    0.696105 182.985504 0.000000 10.34113  5.61642  646.62653 1295.67725
 21)    0.476496  31.923330 0.000000  1.59842 54.55246  288.53766 1174.80847
 22)    0.581488 171.727509 0.000000  8.48798  5.16333  607.06958 1144.57190
 23)    0.717919 107.619682 0.000000  6.56974 12.98110  522.85101 1398.18384
 24)    0.260627 135.821854 0.000000  3.67514  3.29501  466.74823  655.06512
 25)    0.541009 193.130066 0.000000  8.91807  4.01728  627.74432 1073.38794
 26)    0.416957 144.553726 0.000000  5.60796  4.80413  527.92688  915.15741
 27)    0.685347 194.900528 0.000000 10.81306  5.02372  661.48584 1272.96716
 28)    0.757890  77.180283 0.000000  5.08107 22.71966  460.68573 1504.00781
 29)    0.695064 182.565384 0.000000 10.30772  5.62761  645.81714 1294.57288
 30)    0.663895 177.339615 0.000000  9.68878  5.61681  632.16113 1255.90881
 31)    0.728978  67.585785 0.000000  4.37415 26.73451  432.42569 1481.98474
 32)    0.711330 120.021751 0.000000  7.19542 10.89661  545.92004 1373.60791
 33)    0.273794 130.023682 0.000000  3.67212  3.69953  463.08255  681.90204
 34)    0.477893  45.728088 0.000001  2.21386 31.68526  336.10333 1135.21619
 35)    0.517391 171.940155 0.000000  7.75756  4.58340  592.94513 1051.51477
 36)    0.633017 166.295761 0.000000  8.81056  5.90431  609.65240 1221.21204
 37)    0.505782 177.379608 0.000000  7.83810  4.27300  597.80103 1031.09680
 38)    0.486744 199.588150 0.000000  8.45907  3.43663  622.61182  990.89966
 39)    0.506519 152.534561 0.000000  6.85050  5.38247  561.91119 1048.08557
 40)    0.583818 105.598839 0.000000  5.49662 10.85595  497.03064 1205.85364
 41)    0.726151 181.110077 0.000000 10.58902  5.95231  649.51013 1337.38110
 42)    0.369985 103.618721 0.000000  3.78603  7.06769  448.59866  867.97968
 43)    0.642848  47.710159 0.000000  2.89855 40.00734  364.15424 1401.44360
 44)    0.389450  57.867855 0.000003  2.33266 18.03830  355.45917  955.61401
 45)    0.277080 186.072372 0.000000  5.11744  2.17143  538.53845  663.30029
 46)    0.587628  78.165901 0.000000  4.21427 17.26155  439.25943 1249.04138
 47)    0.688636 185.771057 0.000000 10.39486  5.42978  649.21320 1283.61633
 48)    0.435565 143.546280 0.000000  5.76581  5.07304  531.17499  945.25739
 49)    0.398500 105.077202 0.000000  4.06255  7.45460  458.23004  914.68604
 50)    0.561466  86.991112 0.000000  4.47823 14.01541  454.91550 1195.44812
 51)    0.650304  74.220558 0.000000  4.35321 20.67566  439.05487 1351.34302
 52)    0.430250 125.804352 0.000000  5.07148  6.12358  501.67865  949.48627
 53)    0.628764 176.212204 0.000000  9.23335  5.37024  623.49402 1208.14832
 54)    0.316410 131.499115 0.000000  4.15272  4.20510  479.43826  756.44104
 55)    0.547799 195.628342 0.000000  9.11002  3.98920  632.69232 1081.72864
 56)    0.354577 107.605789 0.000000  3.78907  6.39450  451.68488  838.40784
 57)    0.692318 189.607254 0.000000 10.63199  5.29194  655.40668 1285.92371
 58)    0.758101 185.827393 0.000000 11.20708  5.97705  662.24023 1376.20044
 59)    0.291352 143.251266 0.000000  4.20573  3.39855  488.33719  706.36719
 60)    0.431961 192.545975 0.000000  7.46165  3.21945  598.57928  912.04132

max=0.000004 min=0.000000 avg=0.000000 sum=0.000017 no_mutation=0 no_cross=0

 FEED     0.336536
 SPEED   59.330444
 FITNESS  0.000004
```

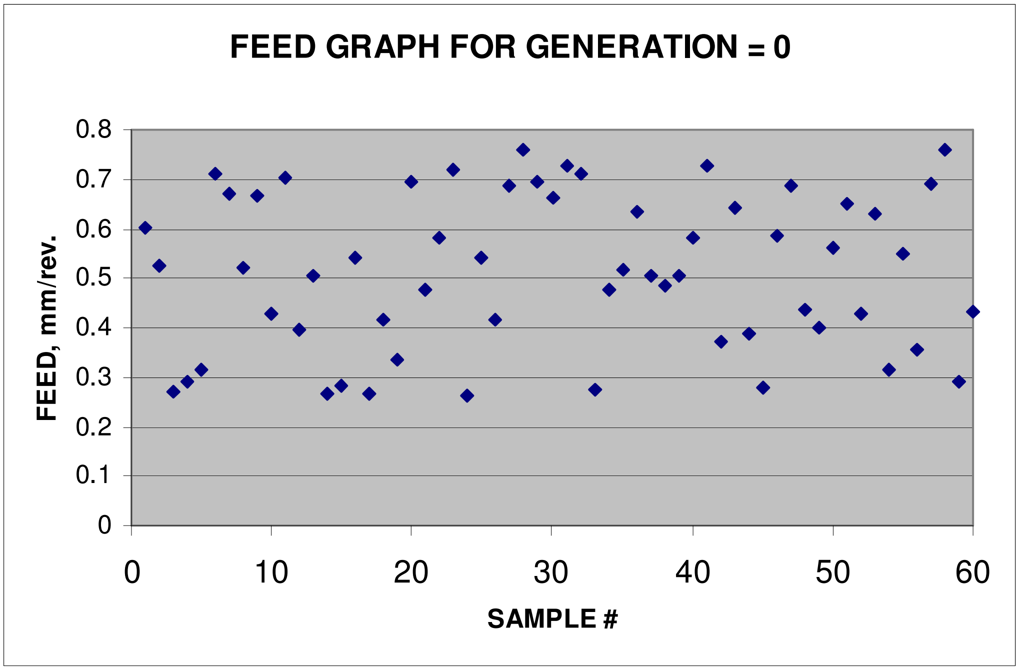


FIG. 5.2(a)

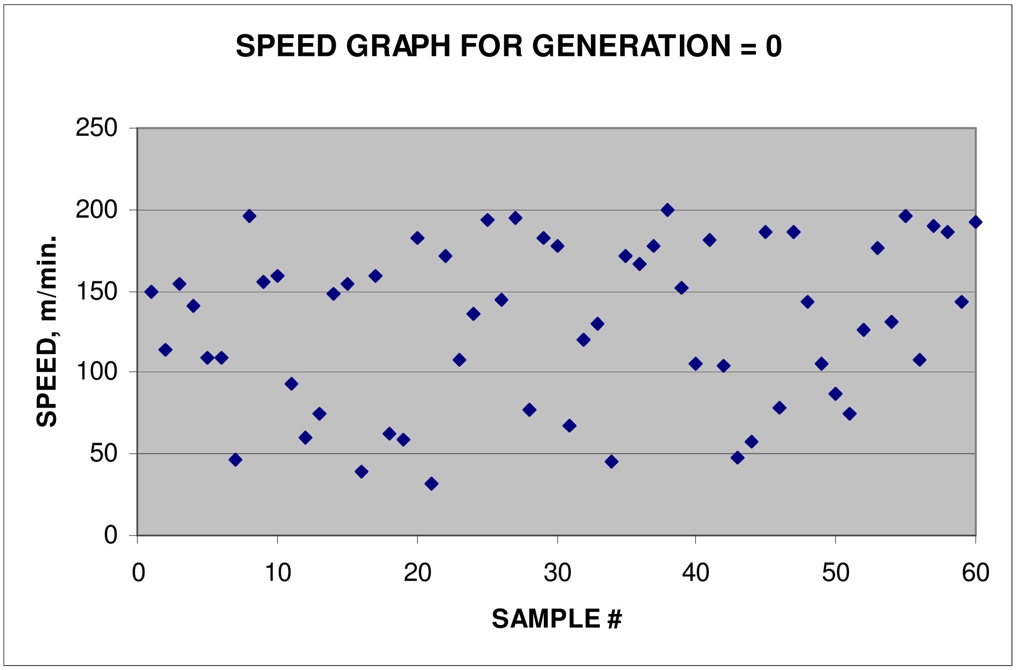


FIG. 5.2(b)

Fig. 5.2 Speed and feed graphs for starting generation with doc = 5.08mm

TABLE 5.2(b)

```
Accumulated Statistics for Generation ---> 240

*********************************************Calculated Constraints************
        FEED       SPEED   FITNESS    POWER   ROUGNESS    TEMP      FORCE
*******************************************************************************

  1)   0.292876  85.999023 0.420278   2.66787  7.42008  395.32318  746.65900
  2)   0.309557  86.024597 0.429376   2.69950  7.97880  394.33133  777.92627
  3)   0.293800  86.486893 0.000001   4.33965 13.84488  450.28546 1172.55408
  4)   0.295504  91.311440 0.000000   2.68808  7.37983  396.51981  747.93616
  5)   0.294052  86.178932 0.421234   2.68125  7.42634  395.99991  748.67114
  6)   0.293748  85.914803 0.420626   2.67170  7.45333  395.40674  748.34314
  7)   0.309549  87.352722 0.000000   2.82504  7.66018  402.54095  776.01056
  8)   0.293125  85.997719 0.420415   2.66960  7.42657  395.39084  747.11987
  9)   0.298703  86.082436 0.423650   2.71154  7.55715  397.11490  757.32471
 10)   0.294196  85.823067 0.420724   2.67231  7.47687  395.35602  749.25092
 11)   0.293748  85.846901 0.420513   2.66980  7.46229  395.27606  748.40302
 12)   0.292656  85.822792 0.419860   2.66138  7.43761  394.92209  746.40588
 13)   0.292877  85.998367 0.420277   2.66786  7.42018  395.32214  746.66089
 14)   0.277801  86.355331 0.412133   2.56968  6.99255  391.65628  718.29138
 15)   0.294048  85.914803 0.420794   2.67383  7.46097  395.49106  748.89679
 16)   0.294172  85.914803 0.420863   2.67471  7.46413  395.52591  749.12579
 17)   0.357548  85.916420 0.000000   3.11441  9.07901  411.97635  862.94940
 18)   0.292532  85.914803 0.419943   2.66307  7.42235  395.06400  746.09552
 19)   0.301611  86.452652 0.425839   2.74269  7.58141  398.63394  762.33203
 20)   0.308407  86.024597 0.428771   2.77833  7.81162  399.66403  775.13544
 21)   0.293748  86.085632 0.420910   2.67648  7.43086  395.73517  748.19250
 22)   0.294228  86.257217 0.421462   2.68470  7.42055  396.19977  748.92706
 23)   0.294048  85.914803 0.420794   2.67383  7.46097  395.49106  748.89679
 24)   0.298792  86.085640 0.423704   2.71226  7.55897  397.14572  757.48499
 25)   0.300807  86.354683 0.425244   2.73419  7.57415  398.22293  760.94525
 26)   0.285738  86.420502 0.416910   2.62855  7.18492  394.09412  733.05707
 27)   0.298705  43.331703 0.000002   1.46191 21.45293  297.49435  811.86371
 28)   0.294047  85.998779 0.420933   2.67617  7.44987  395.65234  748.82092
 29)   0.297841  86.409111 0.423712   2.71467  7.49198  397.50375  755.44891
 30)   0.356032  86.238075 0.000000   3.11456  8.98916  412.25363  859.96967
 31)   0.309487  85.914803 0.429161   2.78271  7.85432  399.74271  777.20337
 32)   0.277646  86.023300 0.411479   2.55967  7.02968  390.98019  718.28101
 33)   0.298791  85.998741 0.423560   2.70980  7.57057  396.97800  757.56158
 34)   0.295504  75.462135 0.000012   2.38837  9.13239  374.98016  761.52576
 35)   0.293125  87.325844 0.422589   2.70668  7.25557  397.93158  745.96088
 36)   0.309487  85.914803 0.429161   2.78271  7.85432  399.74271  777.20337
 37)   0.300807  86.414345 0.425341   2.73589  7.56621  398.33786  760.89203
 38)   0.292876  85.999016 0.420278   2.66787  7.42008  395.32318  746.65900
 39)   0.294172  86.910896 0.422501   2.70260  7.33449  397.43613  748.25153
 40)   0.298792  85.906082 0.423407   2.70717  7.58300  396.79932  757.64520
 41)   0.298672  86.486839 0.424295   2.72279  7.50272  397.88489  756.90851
 42)   0.348866  85.916946 0.000000   3.05528  8.85761  409.86975  847.70654
 43)   0.292861  86.021881 0.420307   2.66840  7.41668  395.36267  746.60980
 44)   0.309557  85.917107 0.429201   2.78327  7.85578  399.76599  777.32819
 45)   0.295094  88.079681 0.424885   2.74198  7.20969  399.92227  748.93756
 46)   0.298212  86.001862 0.423247   2.70578  7.55541  396.82297  756.49323
 47)   0.294048  85.914803 0.420794   2.67383  7.46097  395.49106  748.89679
 48)   0.298705  85.993828 0.423505   2.70905  7.56905  396.94467  757.40820
 49)   0.334367  86.019745 0.000000   2.95896  8.47264  406.45721  821.91620
 50)   0.294047  85.831795 0.420655   2.67150  7.47192  395.33102  748.96838
 51)   0.293125  87.325844 0.422589   2.70668  7.25557  397.93158  745.96088
 52)   0.298705  85.993828 0.423505   2.70905  7.56905  396.94467  757.40820
 53)   0.307504 172.352722 0.000000   5.18092  2.70853  533.11261  720.91333
 54)   0.485670  85.914803 0.000000   3.95470 12.34783  439.12057 1077.49805
 55)   0.292876  85.999016 0.420278   2.66787  7.42008  395.32318  746.65900
 56)   0.300807  86.414345 0.425341   2.73589  7.56621  398.33786  760.89203
 57)   0.298672  86.486839 0.424295   2.72279  7.50272  397.88489  756.90851
 58)   0.292470  86.024597 0.420091   2.66569  7.40638  395.25745  745.88434
 59)   0.298703  85.916412 0.423376   2.70684  7.57935  396.79462  757.47284
 60)   0.294228  86.736153 0.422247   2.69811  7.35836  397.11771  748.50708

max=0.429376 min=0.000000 avg=0.351944 sum=21.116669 no_mutation=17234 no_cross=4300

 FEED     0.309557
 SPEED   86.024597
 FITNESS  0.429376
```

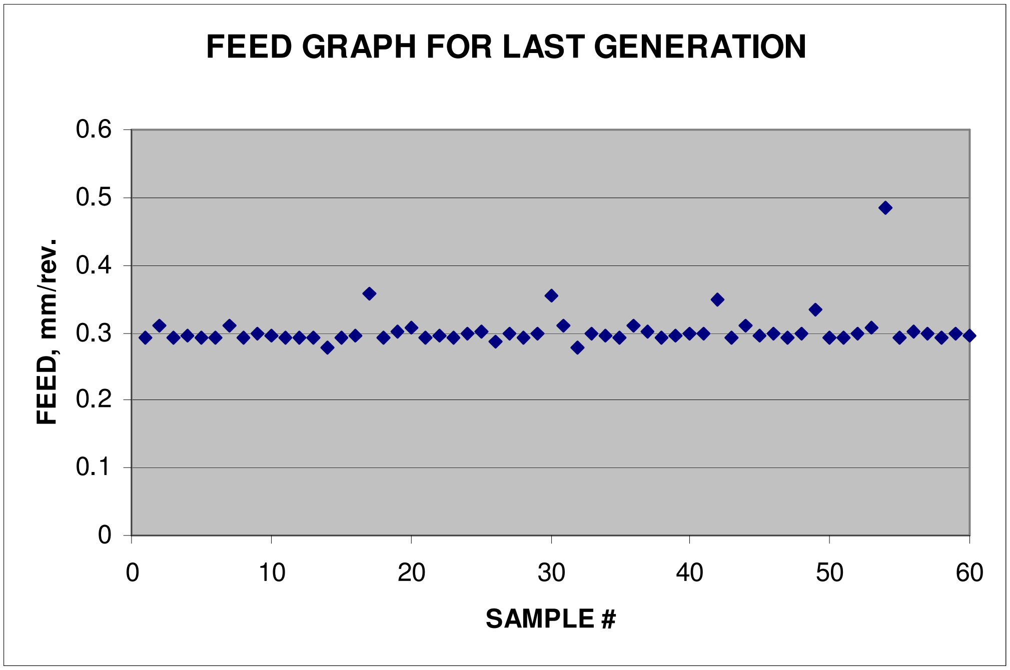


FIG. 5.2(c)

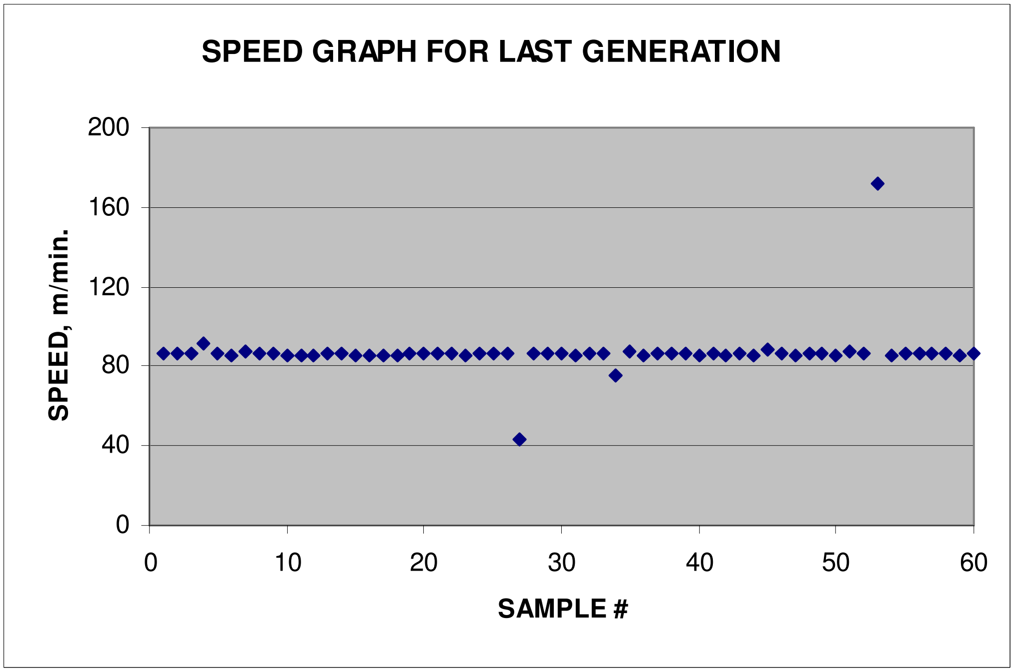


FIG. 5.2(d)

Fig. 5.2 Speed and feed graphs for last generation with doc = 5.08mm

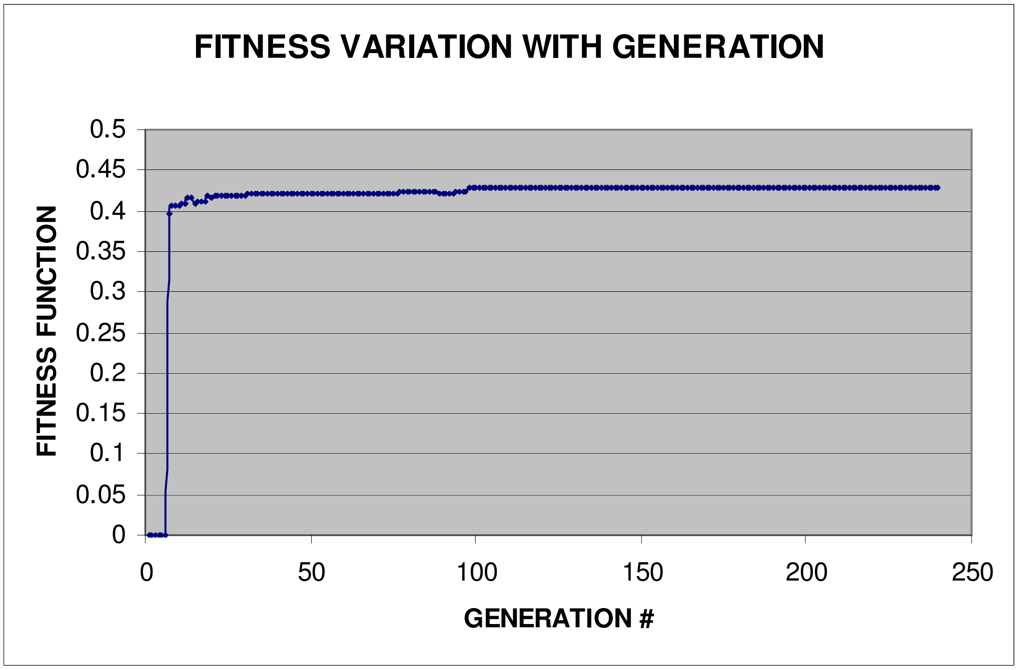


FIG. 5.2(e)

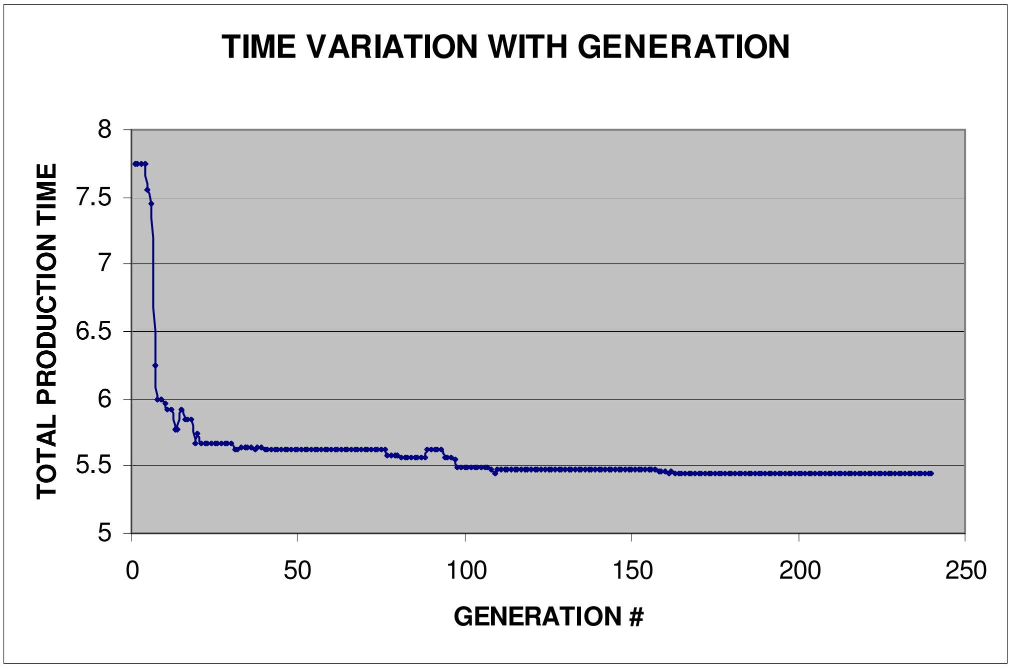


FIG. 5.2(f)

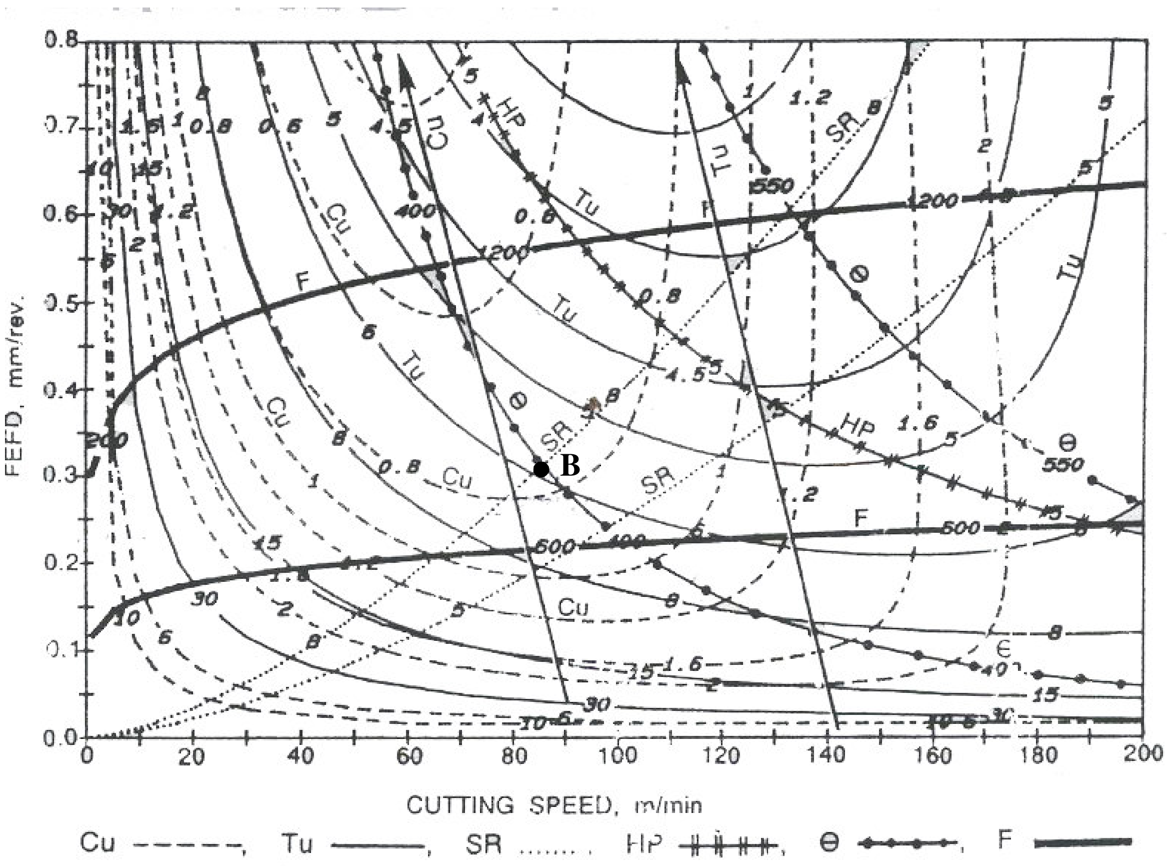


FIG. 5.2(g) Location of best result found by GA on the contours given by Agapiou [6], for doc = 5.08 mm and the constraints of power, force, surface roughness, and temperature.

In the section 5.3, the total production cost is taken as an objective function with depth of cut, doc = 2.54 mm and the constraints force, surface roughness, temperature and power. Machining parameters used for the example are::

Parameter	**Value**
SR_{max}	8 μm
HP_{max}	4 KW
F_{max}	600 N
T_{max}	400 deg C

The table 5.3(a), shows the values of randomly selected machining parameters, the associated fitness values and the various constraints at the starting generation. After applying the various GA operations, we get the table 5.3(b), that shows the values of above stated variables at the end generation. It is clear from these tables that the value of fitness function becomes zero, when the constraints cross their extreme limits. These both tables are also graphically plotted in fig. 5.3(a) to fig. 5.3(d).

Fig. 5.3(e) and fig. 5.3(f), shows the variation of best fitness function and the objective function values with generation, respectively. These figures clearly show how the values of both these functions get improved with generation, after applying the GA operations. From fig. 5.3(f), the best value of the objective function i.e. the total production cost is approximated to be 0.51912403 \$. It is clear from table 5.3(b) that the constraints surface roughness = 7.97486 μm, force = 556.27032 N and temperature = 399.54901 deg C have their extreme limits for the calculated best objective function value.

From fig. 5.3(g), that we have from J. S. Agapiou [6], the value of objective function lies at point C (less than 0.6 \$) that justifies our objective function value.

5.3 TOTAL PRODUCTION COST AS AN OBJECTIVE FUNCTION with doc =2.54mm

TABLE 5.3 (a)

```
Accumulated Statistics for Generation ---> 0
*********************************************Calculated Constraints************
        FEED      SPEED    FITNESS    POWER   ROUGNESS   TEMP      FORCE
******************************************************************************

  1)    0.292335 132.968033 0.000000  2.34474  3.21129  439.30115  424.22491
  2)    0.555556  91.106155 0.000000  2.75305 10.86986  428.98825  702.09052
  3)    0.449115 117.851395 0.000000  2.94018  5.93701  456.81610  586.27393
  4)    0.627980 100.989311 0.000000  3.32338 10.51159  459.32965  759.35553
  5)    0.400390 193.546219 0.000000  4.20124  2.48892  547.84985  512.99951
  6)    0.541260 167.803116 0.000000  4.67424  4.18479  549.66406  647.61218
  7)    0.305438  38.739738 0.000002  0.79968 21.87232  263.76880  496.20035
  8)    0.582464 198.214157 0.000000  5.74996  3.49716  597.72369  671.56482
  9)    0.297424  88.476028 0.620547  1.64706  6.06909  371.91083  447.66342
 10)    0.689890  52.994846 0.000000  2.00167 30.78431  357.85269  867.79700
 11)    0.551487 130.097961 0.000000  3.77203  6.27827  496.66095  673.60504
 12)    0.397367  45.946182 0.000002  1.14480 21.97858  299.94250  590.19592
 13)    0.256019 172.642548 0.000000  2.67435  1.89003  476.34061  375.26791
 14)    0.755221 119.513138 0.000000  4.46587  9.79364  511.76367  853.35394
 15)    0.597399 140.026352 0.000000  4.28952  6.08359  520.56134  708.51361
 16)    0.447940  99.459442 0.000000  2.51864  7.66359  425.44046  595.30597
 17)    0.522132 151.958099 0.000000  4.15675  4.69311  523.68036  637.30859
 18)    0.600518 125.005272 0.000000  3.88884  7.26672  497.20755  719.41583
 19)    0.361475 145.726425 0.000000  3.00483  3.45750  476.92996  490.23969
 20)    0.435209  72.425919 0.000000  1.85107 12.05711  370.38098  602.02740
 21)    0.500640  35.975334 0.000000  1.09994 40.20001  283.94785  715.32031
 22)    0.476404  53.474293 0.000000  1.51177 20.93806  332.29843  662.88568
 23)    0.397115  56.130802 0.000004  1.37016 16.20153  326.39215  578.08032
 24)    0.575647  77.674034 0.000000  2.45191 14.35514  404.36157  732.14325
 25)    0.671931  35.087349 0.000000  1.35295 56.10932  299.05627  887.67596
 26)    0.395937  30.829102 0.000000  0.79716 40.16294  252.87405  612.93542
 27)    0.508467 165.055237 0.000000  4.38617  4.03014  538.92578  619.95776
 28)    0.350291 142.078934 0.000000  2.86593  3.48170  468.87012  480.42664
 29)    0.641004  96.909134 0.000000  3.25397 11.42467  453.45416  773.96838
 30)    0.708093 138.297150 0.000000  4.84324  7.35333  536.41779  802.45142
 31)    0.522005  88.734421 0.000000  2.56096 10.62859  418.82538  672.88434
 32)    0.565257 163.732956 0.000000  4.72943  4.53731  549.01428  669.96820
 33)    0.536809 197.995209 0.000000  5.38993  3.22741  587.48926  633.04706
 34)    0.417081 147.055374 0.000000  3.38718  3.93695  493.14420  543.32709
 35)    0.411751 114.914871 0.000000  2.68589  5.65394  443.95822  551.90210
 36)    0.643016 174.114868 0.000000  5.52713  4.70342  578.29547  731.02441
 37)    0.590551  52.786884 0.000000  1.76680 26.49308  345.77914  775.59912
 38)    0.464380  61.641014 0.000000  1.68414 16.44254  350.88864  641.41229
 39)    0.529146 167.894470 0.000000  4.59468  4.08737  547.22375  637.03540
 40)    0.299137 104.653839 0.622860  1.92438  4.72906  399.46606  441.94870
 41)    0.475444  62.308090 0.000000  1.73206 16.56267  354.22733  651.74493
 42)    0.683941 161.017441 0.000000  5.40548  5.63562  567.11127  770.54846
 43)    0.677419  70.521309 0.000000  2.55206 19.57778  401.80118  831.96362
 44)    0.448983  96.658516 0.000000  2.45917  8.02238  420.60861  598.03876
 45)    0.391266 103.945671 0.000000  2.35824  6.25641  421.29047  537.28802
 46)    0.596014 151.364059 0.000000  4.59256  5.39203  537.33356  701.76575
 47)    0.291578  91.624153 0.619365  1.67360  5.64142  375.83350  439.70691
 48)    0.703280  88.674088 0.000000  3.22932 14.35145  445.52051  835.26465
 49)    0.695903 173.502808 0.000000  5.86001  5.11921  586.94110  774.41528
 50)    0.416767 185.150482 0.000000  4.16508  2.77182  542.37750  530.50543
 51)    0.275311  78.578583 0.600090  1.39370  6.72590  348.15155  428.39642
 52)    0.541328 185.495361 0.000000  5.11603  3.59383  572.89392  641.12781
 53)    0.504070  30.825405 0.000000  0.96227 51.19051  266.25266  730.20923
 54)    0.552917  84.893860 0.000000  2.57393 12.04393  416.12988  704.69434
 55)    0.374497  73.831947 0.000010  1.67511 10.07026  361.79858  538.84619
 56)    0.526812  96.564583 0.000000  2.78331  9.43297  434.67578  671.59253
 57)    0.672678  43.300648 0.000000  1.63632 40.80158  326.94373  869.66357
 58)    0.311996  83.715027 0.624028  1.62665  6.92619  367.04718  466.06430
 59)    0.556007 158.816315 0.000000  4.54258  4.67444  540.29895  664.04913
 60)    0.593069 148.181244 0.000000  4.48818  5.54143  532.08954  700.75732

max=0.624028 min=0.000000 avg=0.051448 sum=3.086909 no_mutation=0 no_cross=0

 FEED      0.311996
 SPEED   83.715027
 FITNESS  0.624028
```

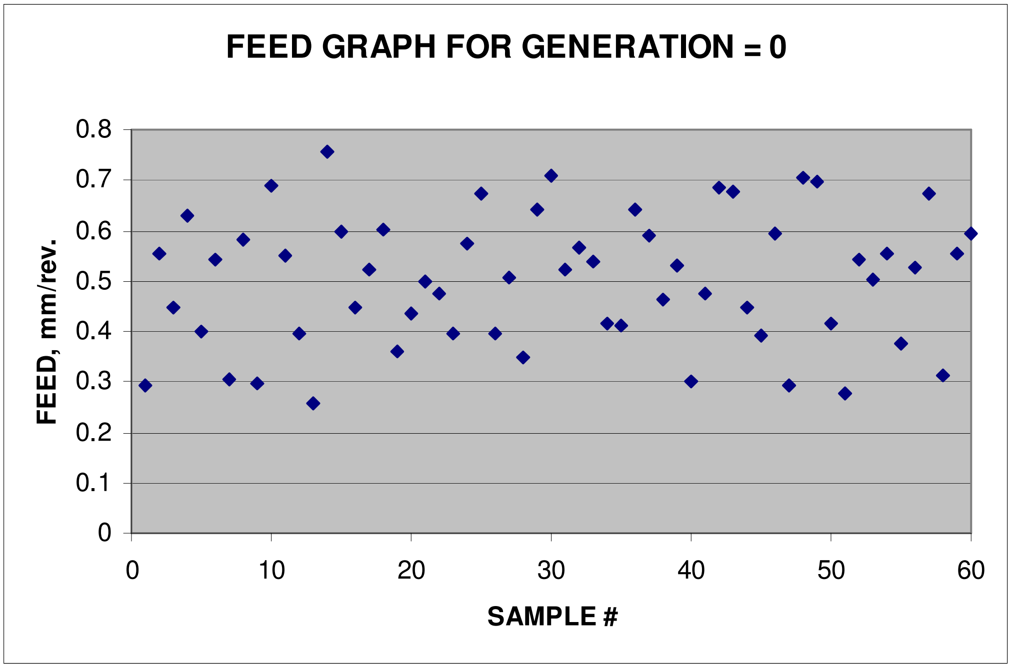


FIG. 5.3(a)

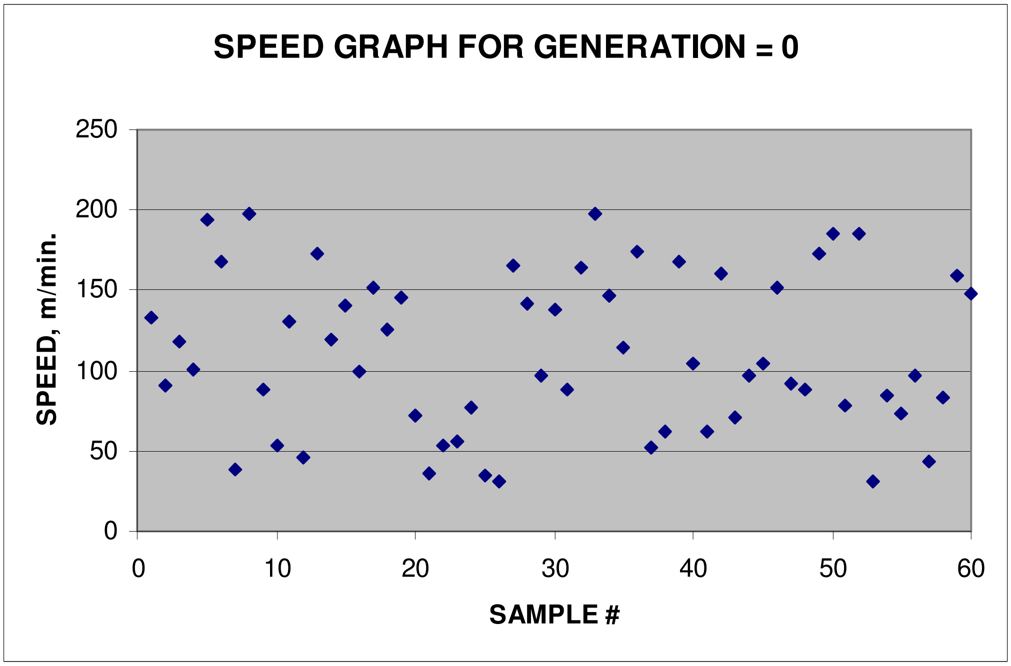


FIG. 5.3(b)

Fig. 5.3 Speed and feed graphs for starting generation with doc = 2.54mm

TABLE 5.3(b)

```
Accumulated Statistics for Generation ---> 240

*******************************************Calculated Constraints************
        FEED       SPEED    FITNESS   POWER   ROUGNESS    TEMP      FORCE
****************************************************************************

  1)    0.266643  91.917366 0.607243  1.56537  5.13215  369.35254  411.97861
  2)    0.278927 175.066818 0.000000  1.47929  6.95564  354.80020  446.01025
  3)    0.275449  90.388443 0.610978  1.58149  5.43925  369.27078  422.51578
  4)    0.321516  88.904922 0.631136  1.75787  6.51470  378.79578  473.43884
  5)    0.260336  92.937363 0.604394  1.55175  4.92693  369.20813  404.43839
  6)    0.260091  91.386017 0.603416  1.52730  5.04984  366.53860  404.85187
  7)    0.256181  90.270775 0.600517  1.49277  5.06733  363.49872  400.92871
  8)    0.260088  80.761177 0.593813  1.36650  6.09344  347.99875  409.94943
  9)    0.276007  85.215034 0.607598  1.50216  5.96114  360.40536  425.67062
 10)    0.402449  90.271416 0.658240  2.12326  7.97486  399.54901  556.27032
 11)    0.276007  85.215034 0.607598  1.50216  5.96114  360.40536  425.67062
 12)    0.260088  91.386093 0.603415  1.52729  5.04979  366.53799  404.84903
 13)    0.263212  80.766205 0.595715  1.37936  6.16634  348.88242  413.51080
 14)    0.260088  92.557159 0.604065  1.54489  4.95299  368.50043  404.32684
 15)    0.256119  90.270775 0.600482  1.49249  5.06610  363.48022  400.85831
 16)    0.263212  92.557167 0.605750  1.55934  5.01272  369.42398  407.84195
 17)    0.285460  85.092392 0.612432  1.54014  6.17964  362.74182  436.25449
 18)    0.402573  90.271416 0.000004  2.38031  9.23858  411.95892  618.60852
 19)    0.349105  84.929054 0.639941  1.79883  7.58564  378.06839  504.89075
 20)    0.280241  90.401688 0.613392  1.60313  5.53304  370.63129  427.82678
 21)    0.274230  90.276581 0.610292  1.57428  5.42530  368.73624  421.21283
 22)    0.260091  92.937363 0.604262  1.55061  4.92227  369.13519  404.16211
 23)    0.281606  90.266258 0.613993  1.60704  5.57277  370.77563  429.40091
 24)    0.402573  84.923195 0.000013  2.01019  8.75334  389.50204  559.84753
 25)    0.383120  90.405663 0.652769  2.04603  7.57322  395.70645  536.68903
 26)    0.392347  90.265556 0.655434  2.08145  7.77466  397.42166  546.11530
 27)    0.395319  90.399048 0.656275  2.09653  7.81621  398.29474  549.02930
 28)    0.281392  90.558594 0.614042  1.61077  5.54122  371.21948  429.02423
 29)    0.311292  90.516861 0.627564  1.74205  6.13679  379.08295  461.63382
 30)    0.289750  85.580887 0.614973  1.56622  6.21854  364.75378  440.74240
 31)    0.275978  85.215034 0.607582  1.50204  5.96051  360.39740  425.63812
 32)    0.280736  84.922844 0.609851  1.51750  6.09541  361.17139  431.09518
 33)    0.283392  47.898838 0.000005  0.91304 14.69375  284.15662  459.97214
 34)    0.403441  84.958916 0.000013  2.01433  8.76669  389.74612  560.69873
 35)    0.276007  90.361519 0.611247  1.58357  5.45279  369.38150  423.14951
 36)    0.395319  85.257751 0.000014  1.98890  8.54378  388.66226  552.29559
 37)    0.260134 173.728638 0.000000  2.72315  1.90232  479.16406  379.39029
 38)    0.268274  90.404884 0.607259  1.54952  5.29554  367.26099  414.50015
 39)    0.283854  84.538010 0.611142  1.52438  6.20608  361.32010  434.76120
 40)    0.274230  90.281113 0.610295  1.57435  5.42488  368.74399  421.21066
 41)    0.330019  85.209297 0.632722  1.72676  7.13350  374.15814  484.56277
 42)    0.279542  85.587204 0.609783  1.52311  5.99792  362.03064  429.42630
 43)    0.383118  88.942657 0.652624  2.01620  7.76334  393.01633  537.57501
 44)    0.260987  90.371941 0.603287  1.51610  5.15399  365.09021  406.32162
 45)    0.311292  85.580887 0.625045  1.65632  6.68278  370.28009  464.26349
 46)    0.402573  90.271416 0.658274  2.12377  7.97733  399.57474  556.39459
 47)    0.311292  85.583481 0.625047  1.65636  6.68247  370.28473  464.26181
 48)    0.438292  85.580887 0.000011  2.16296  9.42209  397.76648  594.97180
 49)    0.383660  89.242790 0.652825  2.02455  7.73464  393.68646  537.94208
 50)    0.401443  90.270767 0.657966  2.11911  7.95493  399.33893  555.26202
 51)    0.256119  90.401024 0.600569  1.49443  5.05500  363.70013  400.79956
 52)    0.275448  88.987694 0.610138  1.55942  5.56992  366.86011  423.18457
 53)    0.260091  93.038567 0.604312  1.55213  4.91415  369.30389  404.11835
 54)    0.349229  89.853111 0.641826  1.89295  6.96534  387.12296  502.14569
 55)    0.275443  90.398880 0.610981  1.58163  5.43817  369.28696  422.50403
 56)    0.347429  92.896431 0.641652  1.94271  6.58719  392.13013  498.58234
 57)    0.392343  90.400444 0.655437  2.08424  7.75695  397.66891  546.02875
 58)    0.277076  90.281113 0.611744  1.58708  5.48141  369.54306  424.37558
 59)    0.599135  90.517174 0.000000  2.90310 11.84224  434.61240  742.08984
 60)    0.260087  91.386093 0.603414  1.52728  5.04977  366.53766  404.84775

max=0.658274 min=0.0000 avg=0.537718 sum=32.263084 no_mutation=17407 no_cross=4285

 FEED     0.402573
 SPEED   90.271416
 FITNESS  0.658240
```

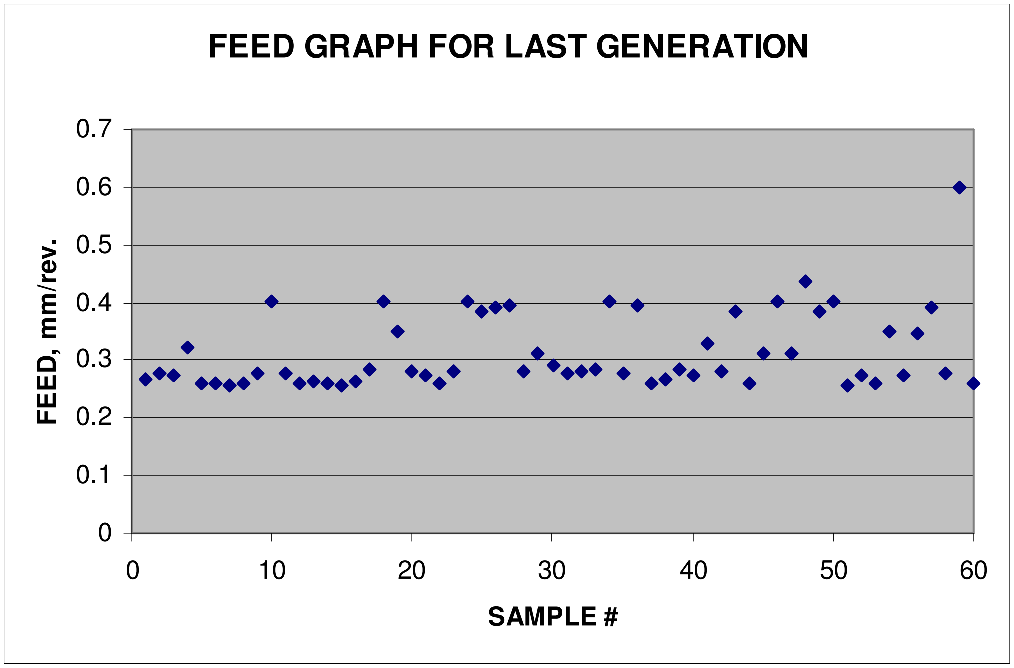


FIG. 5.3(c)

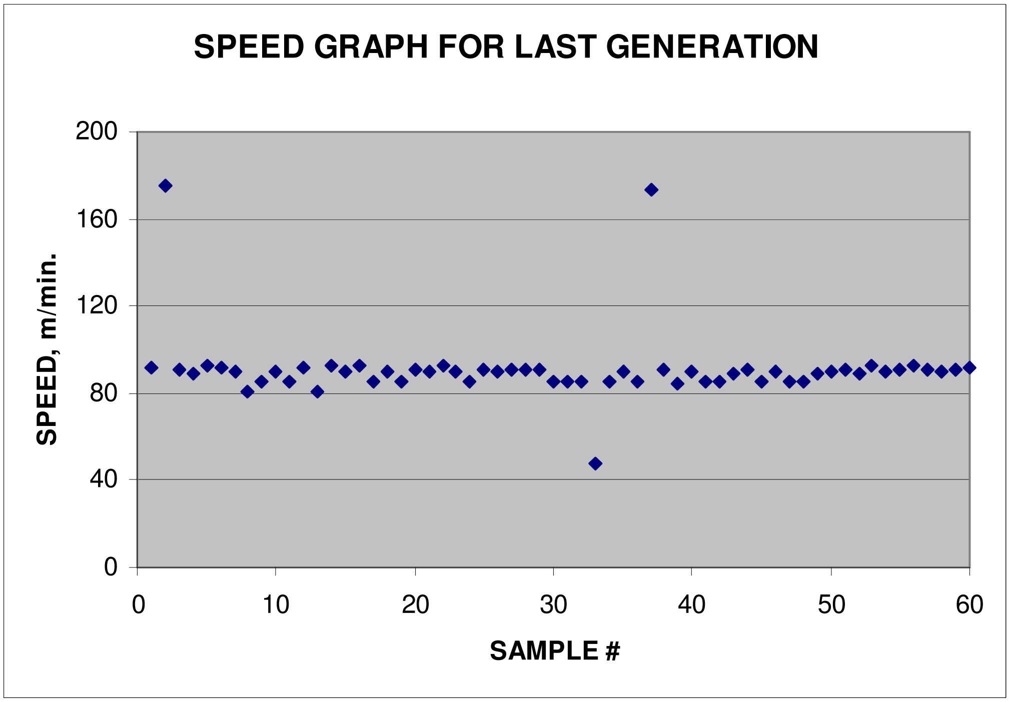


FIG. 5.3(d)

Fig. 5.3 Speed and feed graphs for last generation with doc = 2.54mm

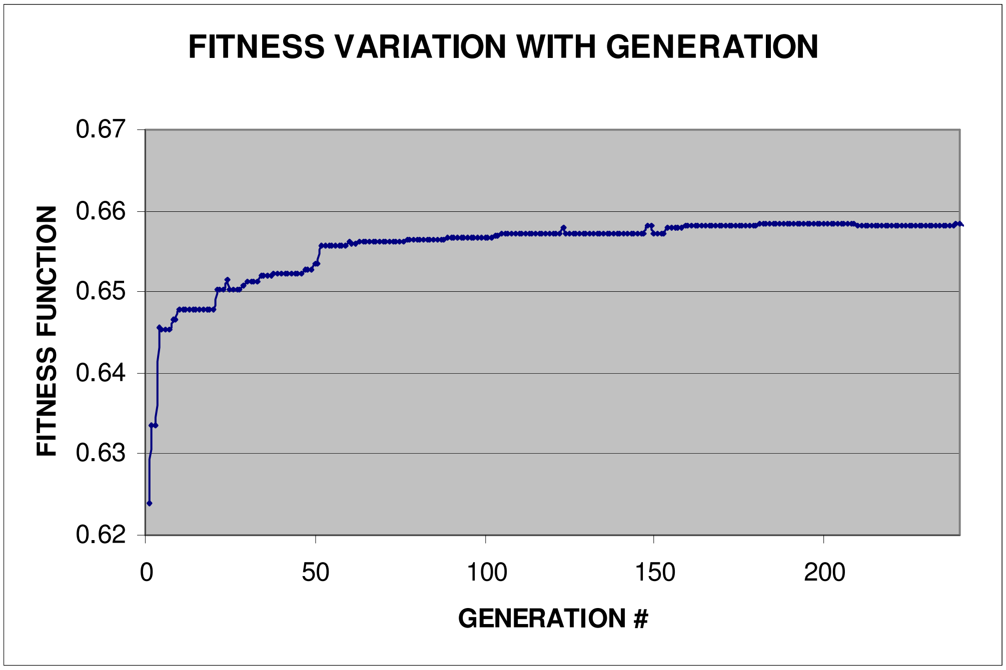


FIG. 5.3(e)

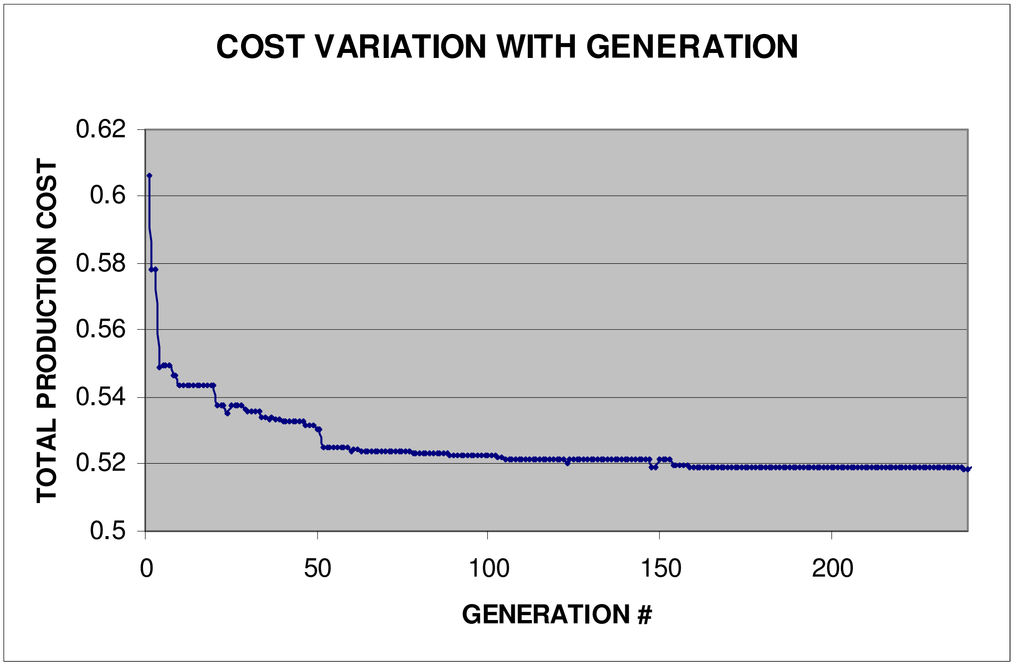


FIG. 5.3(f)

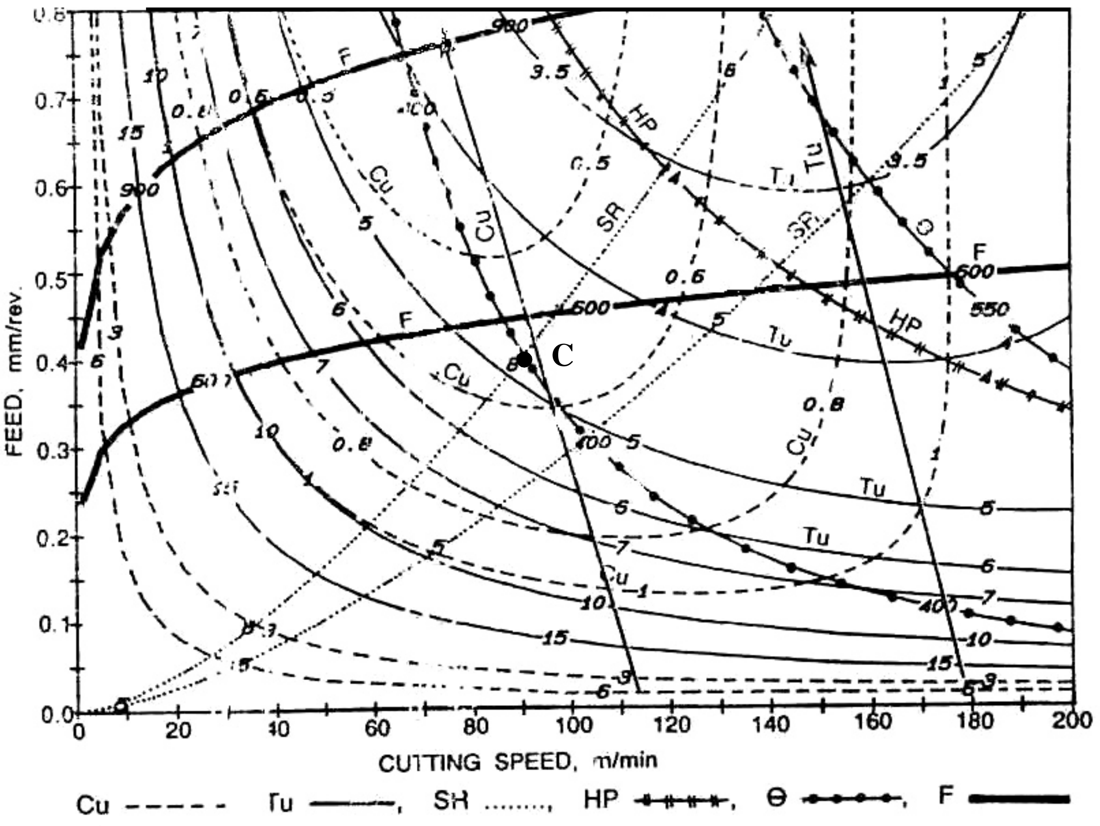


FIG. 5.3(g) Location of best result found by GA on the contours given by Agapiou [6], for doc = 2.54 mm and the constraints of power, force, surface roughness, and temperature.

In the section 5.4, the total production cost is taken as an objective function with depth of cut, doc = 5.08 mm and the constraints force, surface roughness, temperature, and power. Machining parameters used for the example are::

Parameter	**Value**
SR_{max}	8 μm
HP_{max}	5 KW
F_{max}	1200 N
T_{max}	400 deg C

The table 5.4(a), shows the values of randomly selected machining parameters, the associated fitness values and the various constraints at the starting generation. After applying the various GA operations, we get the table 5.4(b), that shows the values of above stated variables at the end generation.

Fig. 5.4(e) and fig. 5.4(f), shows the variation of best fitness function and the objective function values with generation, respectively. These figures clearly show how the values of both these functions get improved with generation, after applying the GA operations. From fig. 5.4(f), the best value of the objective functions i.e. the total production cost is approximated to be 0.71820772 $. It is clear from table 5.4(b) that the constraints surface roughness = 7.99032 μm and temperature = 399.88733 deg C have their extreme limits for the calculated best objective function value.

From fig. 5.4(g), that we have from J. S. Agapiou [6], the value of objective function lies at point D (less than 0.8 $) that justifies our objective function value.

5.4 TOTAL PRODUCTION COST AS AN OBJECTIVE FUNCTION with doc = 5.08mm

TABLE 5.4(a)

```
Accumulated Statistics for Generation ---> 0
*********************************************Calculated Constraints************
         FEED     SPEED    FITNESS    POWER   ROUGNESS    TEMP      FORCE
*******************************************************************************

  1)   0.408028 128.861130 0.000000  4.97227  5.59803  501.15790  911.45538
  2)   0.431649 135.988861 0.000000  5.45332  5.45799  518.45667  944.24762
  3)   0.417278 146.122513 0.000000  5.66609  4.72960  530.37244  914.66638
  4)   0.595206 187.120193 0.000000  9.33804  4.63906  631.92120 1154.00684
  5)   0.278887  71.214409 0.000011  2.16696  9.41010  361.56570  734.53174
  6)   0.649358  61.237061 0.000000  3.65727 27.65409  405.10986 1376.46960
  7)   0.347416  39.013523 0.000001  1.49645 29.28555  293.80276  915.51666
  8)   0.612997 138.745483 0.000000  7.29997  7.52875  562.06177 1215.18018
  9)   0.720976  35.888702 0.000000  2.45328 69.19968  330.94867 1567.52942
 10)   0.493170  58.797855 0.000002  2.84492 22.31654  376.01230 1132.20947
 11)   0.318152  70.350945 0.000008  2.37522 10.94163  369.79678  809.13660
 12)   0.285897 190.316025 0.000000  5.35153  2.16531  547.10760  676.98914
 13)   0.757483  54.809780 0.000000  3.73241 38.20441  399.41806 1556.46509
 14)   0.428904 183.063324 0.000000  7.09069  3.45162  585.39508  912.01007
 15)   0.646126 181.214676 0.000000  9.67227  5.28923  634.25781 1228.75525
 16)   0.472415 193.947983 0.000000  8.05371  3.48359  611.53827  972.48419
 17)   0.513896 194.835175 0.000000  8.63555  3.76454  623.39087 1033.19360
 18)   0.433894 166.907455 0.000000  6.58407  4.01844  564.85620  928.33972
 19)   0.670720 184.164490 0.000000 10.10402  5.35823  643.40460 1260.42566
 20)   0.616414 121.236298 0.000000  6.49346  9.29388  532.21277 1236.87427
 21)   0.470739 193.842026 0.000000  8.02746  3.47406  610.95361  970.03497
 22)   0.532331  87.484390 0.000000  4.31784 13.17163  450.96365 1149.48877
 23)   0.389342 117.283676 0.000000  4.40432  6.16224  477.31546  889.43939
 24)   0.268315  32.659256 0.000001  1.04245 29.60445  257.79843  772.92761
 25)   0.299420  42.516071 0.000002  1.43980 22.13460  295.25638  814.83728
 26)   0.728896 189.040680 0.000000 11.03787  5.59808  661.56079 1335.23486
 27)   0.717508  77.282845 0.000000  4.87446 21.46106  455.72604 1445.27966
 28)   0.758391 111.284836 0.000000  7.06669 13.03519  536.18542 1449.96973
 29)   0.653219  89.224884 0.000000  5.15573 15.69883  474.41107 1330.67969
 30)   0.675701 138.657486 0.000000  7.87162  8.31011  573.31702 1304.16724
 31)   0.301350 151.937500 0.000000  4.55282  3.21473  503.90561  719.55536
 32)   0.262905 159.025757 0.000000  4.26452  2.61538  499.20163  648.76184
 33)   0.537256  75.069305 0.000000  3.78934 16.77592  423.93973 1175.26904
 34)   0.478104  93.774849 0.000000  4.22680 10.64037  453.92004 1055.89746
 35)   0.742246  95.963181 0.000000  6.08174 15.97785  502.07230 1449.10608
 36)   0.263009  50.883278 0.000005  1.52968 14.78915  309.97009  728.35394
 37)   0.583679 153.893936 0.000000  7.71298  6.12279  580.70630 1160.51575
 38)   0.411408  33.794559 0.000001  1.50038 43.16876  286.52557 1050.06335
 39)   0.720651 176.316025 0.000000 10.27531  6.15290  641.38452 1333.64893
 40)   0.362764 174.197006 0.000000  5.95046  3.14607  554.05573  811.80292
 41)   0.425611 145.779343 0.000000  5.74200  4.84171  532.02728  928.09497
 42)   0.296374 125.682091 0.000000  3.78868  4.21812  464.17279  724.71539
 43)   0.383948  83.143631 0.000000  3.19658 10.25078  412.46042  911.71533
 44)   0.621441 109.097389 0.000000  5.94280 10.99975  510.34235 1257.54785
 45)   0.458077 113.079910 0.000000  4.83822  7.66873  486.26569 1004.41370
 46)   0.661922 168.979767 0.000000  9.25522  6.02654  619.34821 1259.34668
 47)   0.330835 111.854355 0.000000  3.71699  5.62374  452.44998  794.20056
 48)   0.483582 193.047501 0.000000  8.16754  3.59158  613.31000  989.56335
 49)   0.454688 145.674576 0.000000  6.04183  5.17950  539.18231  973.71405
 50)   0.426715 114.465324 0.000000  4.62830  7.01071  481.59308  952.89905
 51)   0.598122 165.502396 0.000000  8.39318  5.61826  601.39142 1172.59448
 52)   0.737637 120.437439 0.000000  7.42522 11.24201  550.81519 1409.76013
 53)   0.553303 146.055450 0.000000  7.05803  6.28283  562.08911 1122.33545
 54)   0.737833  83.728340 0.000000  5.35425 19.54118  473.88373 1462.92712
 55)   0.582040 179.800034 0.000000  8.85280  4.81970  618.77820 1140.04224
 56)   0.646133  59.594906 0.000000  3.55505 28.67692  400.12012 1375.29187
 57)   0.329858  48.588875 0.000003  1.75100 19.91369  318.88629  862.34546
 58)   0.260648 124.174423 0.000000  3.39045  3.77639  449.68634  661.07977
 59)   0.258701 127.025024 0.000000  3.44024  3.62097  453.24295  655.98602
 60)   0.716501 193.317734 0.000000 11.11272  5.31854  665.32050 1315.74939

max=0.000011 min=0.000000 avg=0.000001 sum=0.000035 no_mutation=0 no_cross=0

 FEED      0.278887
 SPEED   71.214409
 FITNESS  0.000011
```

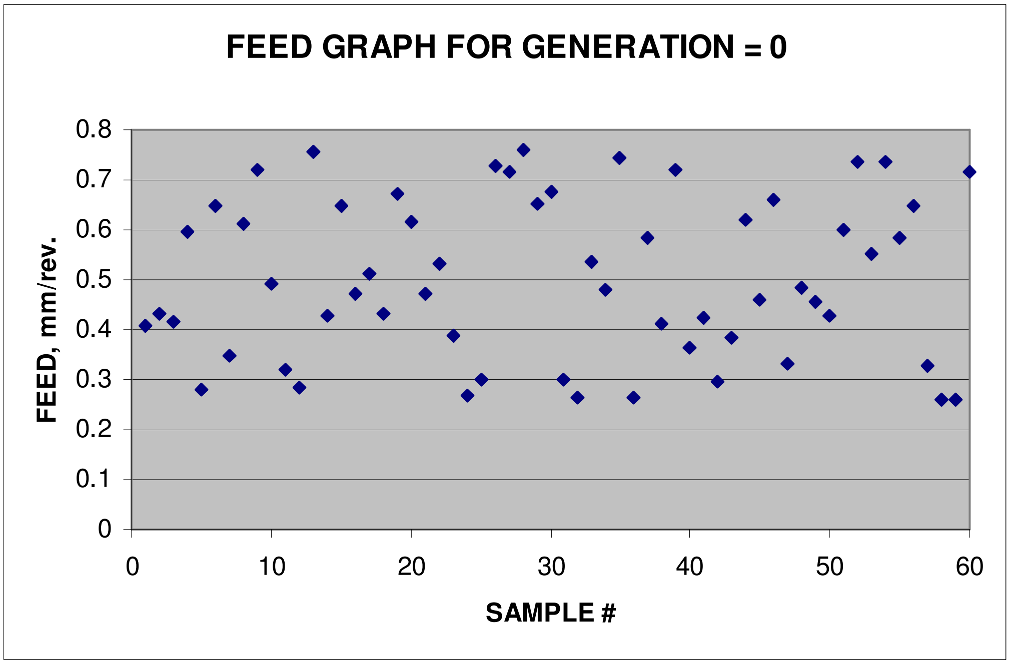


FIG. 5.4(a)

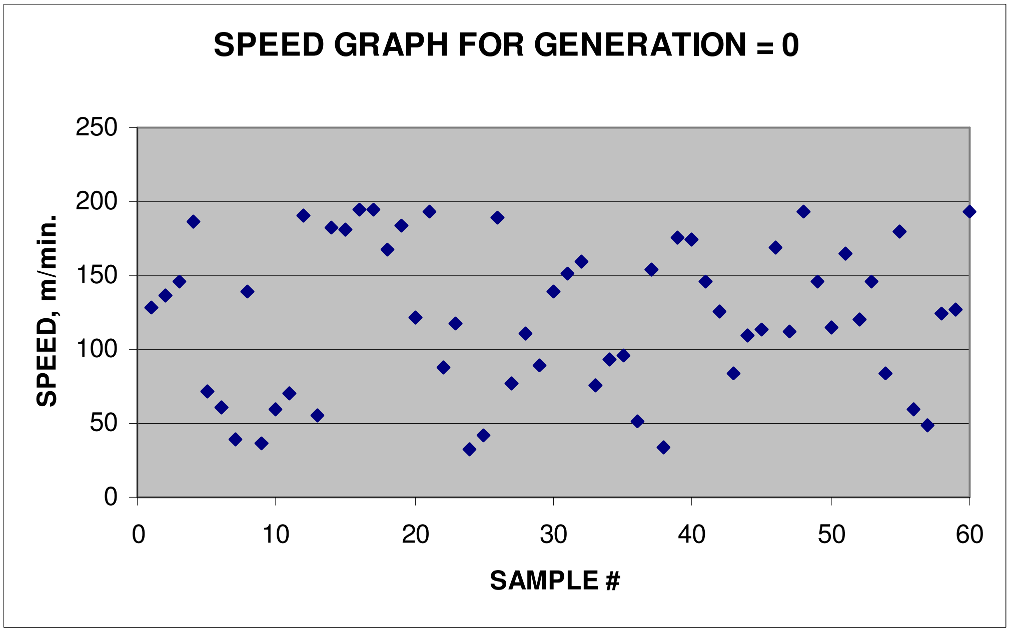


FIG. 5.4(b)

Fig. 5.4 Speed and feed graphs for starting generation with doc = 5.08mm

TABLE 5.4(b)

```
Accumulated Statistics for Generation ---> 240

*********************************************Calculated Constraints************
        FEED      SPEED    FITNESS    POWER   ROUGNESS    TEMP      FORCE
********************************************************************************

  1)   0.307866  85.610725 0.580168  2.72842  7.89848  396.82614  771.41779
  2)   0.295968  84.885216 0.576158  2.52139  6.99811  389.02112  711.99518
  3)   0.274949  84.880081 0.567242  2.50986  7.10415  388.00562  714.18414
  4)   0.307803  85.590622 0.580162  2.76149  7.85644  398.65726  774.43280
  5)   0.308684  85.591270 0.580485  2.36557  6.55369  382.00388  674.89355
  6)   0.256444  90.904022 0.554613  2.14442  7.08149  363.88120  692.93243
  7)   0.412747  85.539909 0.000002  5.09557  2.75585  529.43146  719.55255
  8)   0.303031 169.924545 0.000000  5.05704  2.72716  528.39166  714.32050
  9)   0.288209  85.451767 0.572622  2.61955  7.37255  392.94775  738.49042
 10)   0.376247  74.914909 0.000007  2.86475 11.76842  393.22394  907.95459
 11)   0.290952  85.611664 0.573634  2.64342  7.42188  394.03445  743.43842
 12)   0.274075  84.883972 0.566844  2.50373  7.08097  387.75467  712.53369
 13)   0.291454  85.249161 0.574090  2.63689  7.48283  393.47775  744.68781
 14)   0.312743  85.591789 0.581956  2.79603  7.98286  399.98761  783.42175
 15)   0.304966  84.879860 0.579662  2.72111  7.88302  396.50232  769.90021
 16)   0.313774 106.789581 0.000000  3.42093  5.72172  438.95169  767.88702
 17)   0.274069  87.707535 0.565034  2.57852  6.73725  393.09616  710.16498
 18)   0.336641  84.929802 0.000000  2.94070  8.69744  404.87128  827.03253
 19)   0.295972  84.875473 0.576166  2.65819  7.65023  394.02087  753.37555
 20)   0.274335  84.880043 0.566964  2.50548  7.08822  387.82413  713.02734
 21)   0.400160  85.539909 0.000000  3.38689 10.23373  420.99292  936.76642
 22)   0.312747  84.874825 0.000015  2.77497  8.08568  398.58527  784.09668
 23)   0.273141  84.929802 0.566396  2.49829  7.05096  387.56543  710.73389
 24)   0.303031  84.924545 0.578891  2.52143  6.99794  389.02359  711.99408
 25)   0.305836  84.888168 0.579985  2.72741  7.90443  396.75479  771.48517
 26)   0.274069  85.549332 0.566476  2.52135  6.99729  389.02164  711.96008
 27)   0.296033  84.929840 0.576153  2.66015  7.64436  394.14331  753.43915
 28)   0.308747 106.841270 0.000000  3.37957  5.62554  437.56677  758.91052
 29)   0.295972  84.874825 0.576167  2.65817  7.65031  394.01959  753.37616
 30)   0.307866  85.610725 0.580168  2.76251  7.85524  398.71323  774.52789
 31)   0.306833  87.536331 0.000000  2.81099  7.56851  402.15323  770.90381
 32)   0.528038  85.591568 0.000000  4.20701 13.50666  446.12814 1145.29480
 33)   0.295972  84.929802 0.576129  2.65972  7.64279  394.12628  753.32672
 34)   0.274069  84.880081 0.566843  2.50359  7.08132  387.74551  712.52600
 35)   0.296031  85.559639 0.575701  2.67788  7.55895  395.36197  752.87177
 36)   0.549972  84.874825 0.000000  4.30996 14.25100  448.34924 1180.59778
 37)   0.306085  74.304802 0.000011  2.42091  9.68562  375.31580  782.42371
 38)   0.264222  74.258118 0.000014  2.15731  8.36399  363.82489  703.33398
 39)   0.306085  84.883118 0.580083  2.72900  7.91161  396.81241  771.94507
 40)   0.312747  85.539909 0.582002  2.79453  7.99032  399.88733  783.47693
 41)   0.307866  85.610725 0.580168  2.76251  7.85524  398.71323  774.52789
 42)   0.305666  84.553482 0.580169  2.71655  7.94760  396.05420  771.48273
 43)   0.308684  85.591270 0.580485  2.76767  7.87891  398.89639  776.03693
 44)   0.288035  85.040840 0.572821  2.60698  7.42226  392.10690  738.52710
 45)   0.293988  84.550484 0.575583  2.63517  7.64317  392.83652  750.00153
 46)   0.305837  84.905952 0.579972  2.72793  7.90194  396.78979  771.47046
 47)   0.295972  42.374825 0.000002  1.42259 21.98969  294.11670  808.29718
 48)   0.272215  84.592499 0.566139  2.48277  7.06959  386.64578  709.27136
 49)   0.272215  41.101032 0.000002  1.29660 21.17787  285.21817  763.07605
 50)   0.295972  87.580826 0.574005  2.73432  7.29392  399.22195  750.98383
 51)   0.312685  84.875847 0.000015  2.77457  8.08393  398.57077  783.98297
 52)   0.270405 169.885590 0.000000  4.62614  2.43328  516.03961  657.71228
 53)   0.305836  85.562622 0.579456  2.74690  7.80991  398.06909  770.86652
 54)   0.312653  84.211540 0.000015  2.75480  8.18023  397.25583  784.55035
 55)   0.273142  84.929802 0.566396  2.49830  7.05099  387.56573  710.73572
 56)   0.273141  84.928513 0.566396  2.49825  7.05113  387.56299  710.73499
 57)   0.295972  84.929802 0.576129  2.65972  7.64279  394.12628  753.32672
 58)   0.295972  84.874825 0.576167  2.65817  7.65031  394.01959  753.37616
 59)   0.274087  85.546364 0.566486  2.52139  6.99811  389.02112  711.99518
 60)   0.312743  85.581085 0.581965  2.79571  7.98438  399.96671  783.43170

max=0.582002 min=0.000000 avg=0.411754 sum=24.705212 no_mutation=17561 no_cross=4344

 FEED     0.312747
 SPEED   85.539909
 FITNESS  0.582002
```

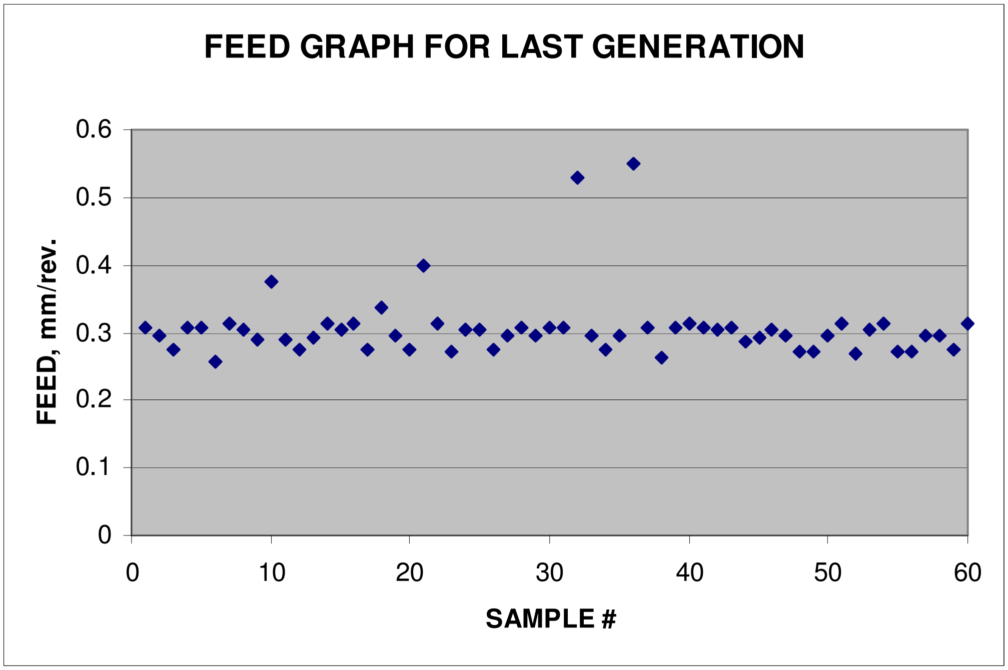


FIG. 5.4(c)

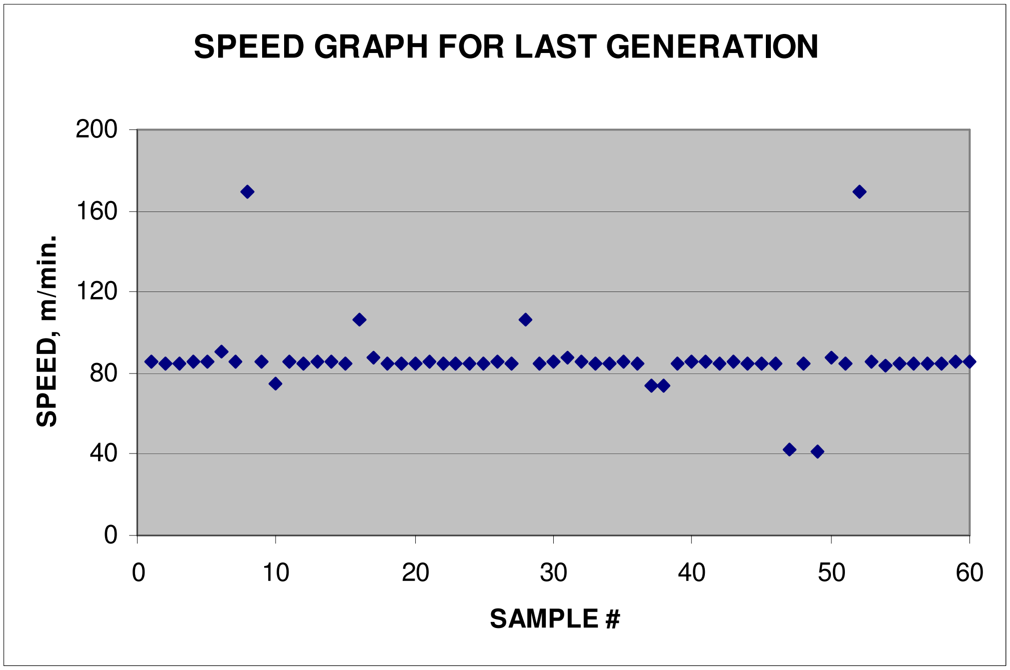


FIG. 5.4(d)

Fig. 5.4 Speed and feed graphs for last generation with doc = 5.08mm

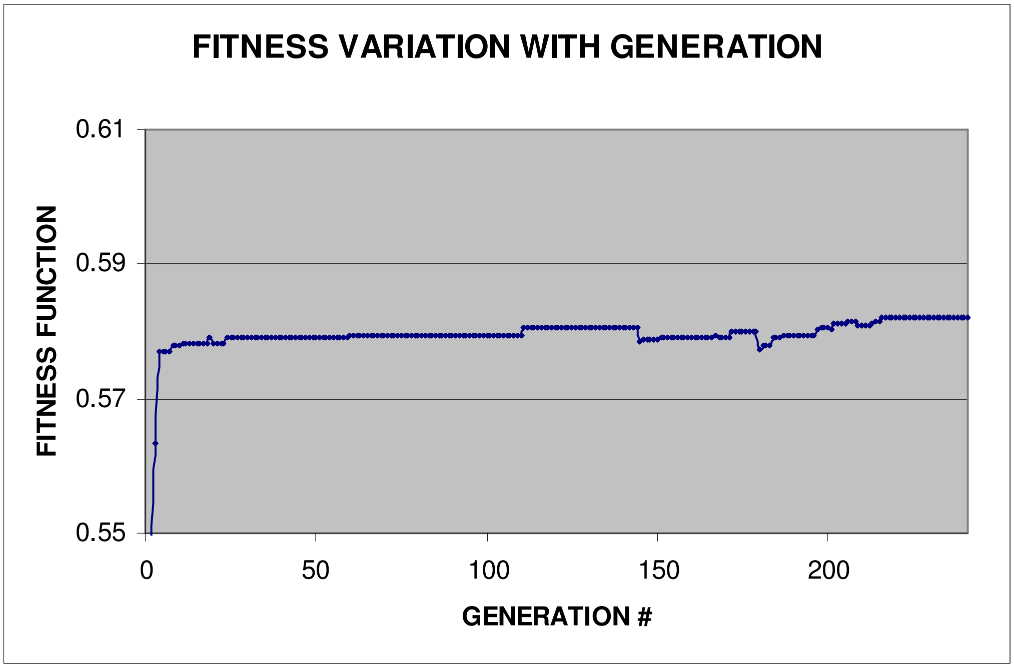


FIG. 5.4(e)

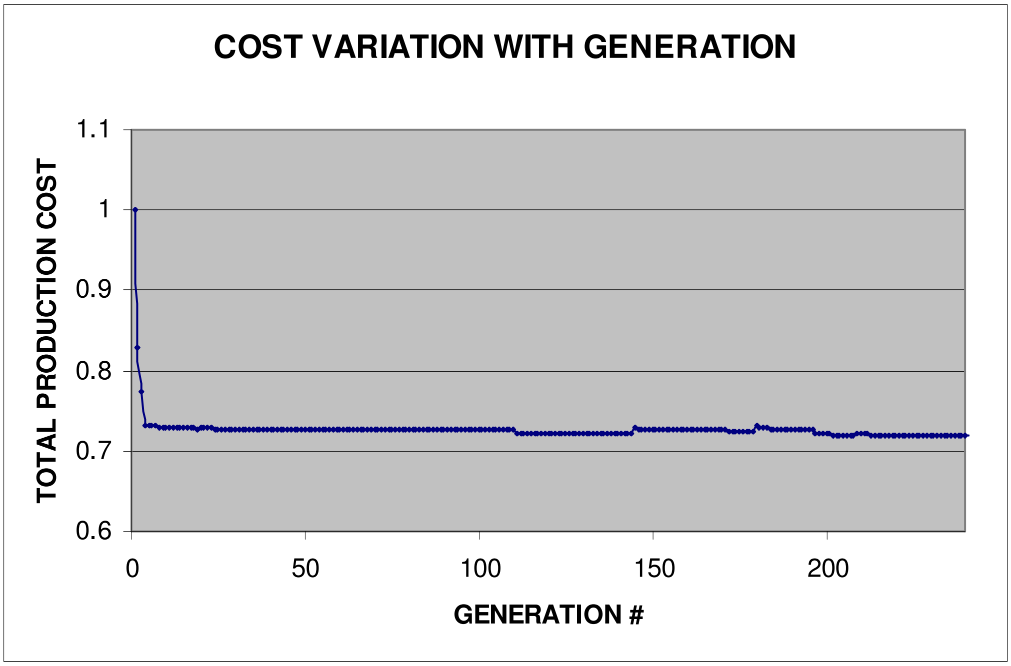


FIG. 5.4(f)

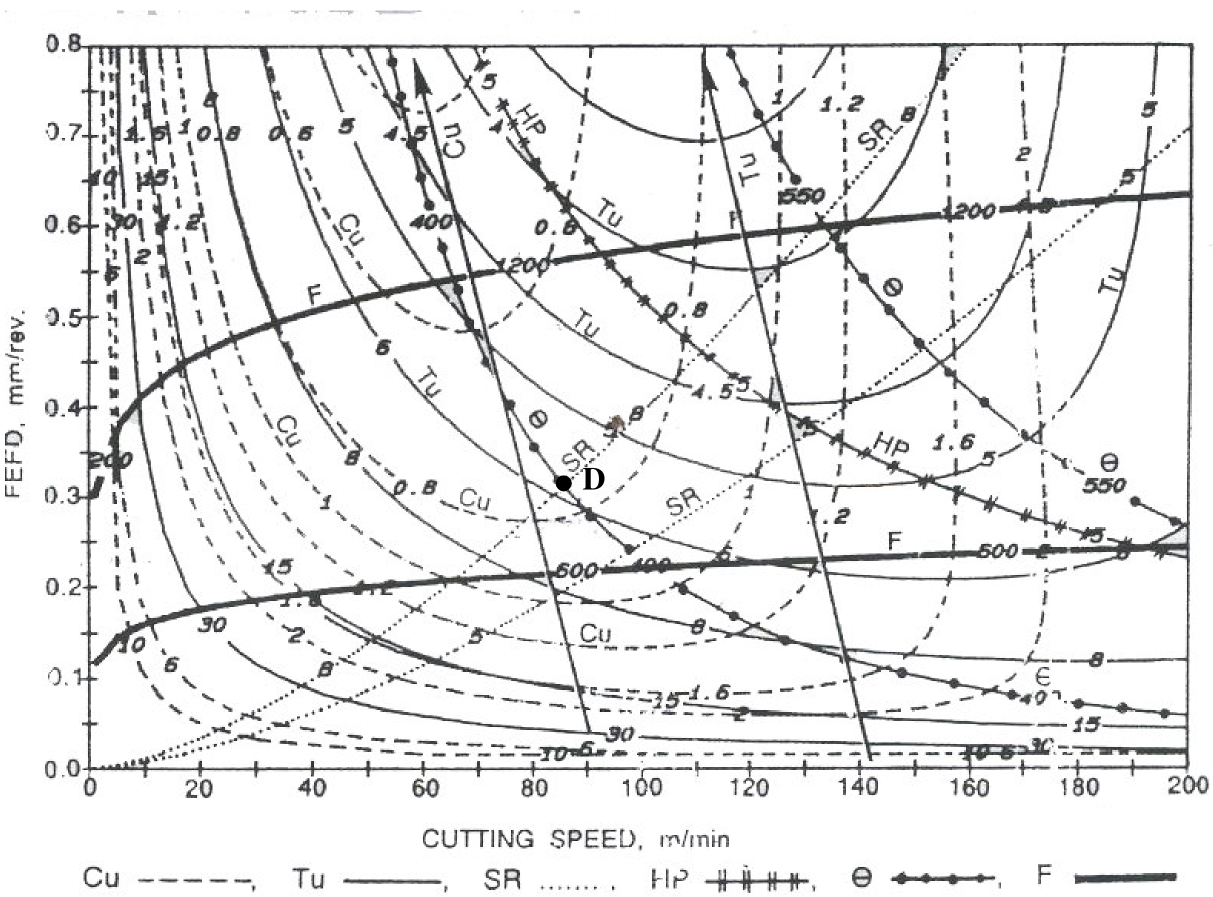


FIG. 5.4(g) Location of best result found by GA on the contours given by Agapiou [6], for doc = 5.08 mm and the constraints of power, force, surface roughness, and temperature.

In the section 5.5, the combined objective function is taken that includes both the total production cost and the total production time with weight coefficients 0.6 and 0.4 respectively, and depth of cut, doc = 2.54 mm having the constraints force, surface roughness, temperature and power.

Machining parameters used for the example are::

Parameter	**Value**
SR_{max}	2 μm
HP_{max}	5 KW
F_{max}	600 N
T_{max}	500 deg C
λ	0.212

Fig. 5.5(e) and fig. 5.5(f), shows the variation of best fitness function and the objective function values with generation, respectively. These figures clearly show how the values of both these functions get improved with generation, after applying the GA operations. From fig. 5.5(f), the best value of the objective function is approximated to be 0.91696537. It is clear from table 5.5(b) that the constraints surface roughness = 1.99998 μm and temperature = 467.68423 deg C have their extreme limits for the calculated best objective function value. Here we get the results at early generation because other constraints are no more acting due to their high values.

From fig. 5.5(g), that we have from J. S. Agapiou [6], the value of objective function lies at point E (less than 1) that justifies our objective function value.

5.5 COMBINED OBJECTIVE FUNCTION with doc = 2.54mm

TABLE 5.5(a)

```
Accumulated Statistics for Generation ---> 0
*********************************************Calculated Constraints************
        FEED      SPEED    FITNESS    POWER  ROUGNESS    TEMP       FORCE
*******************************************************************************

  1)    0.509809 143.570511 0.000000  3.87678  4.99491  508.99881  629.98132
  2)    0.684130 180.702728 0.000000  5.99806  4.73067  594.76294  761.75049
  3)    0.680531 116.590492 0.000012  4.02673  9.15969  495.73724  793.28534
  4)    0.448951 116.046623 0.000027  2.89880  6.07570  453.86099  587.03625
  5)    0.564842  90.154854 0.000008  2.76266 11.23004  428.59384  711.33557
  6)    0.563974  33.474651 0.000000  1.13126 50.55053  282.44473  785.55566
  7)    0.343484 166.771347 0.000140  3.26025  2.67581  499.05298  466.01419
  8)    0.746385 126.624870 0.000000  4.66136  8.86453  522.87634  841.16431
  9)    0.641327 138.249557 0.000000  4.48181  6.66084  525.47351  746.88055
 10)    0.410470 191.418243 0.000000  4.24110  2.59507  548.16132  522.91589
 11)    0.342245 165.294357 0.000137  3.22515  2.70242  496.84512  465.21420
 12)    0.747566 123.108353 0.000000  4.55029  9.26695  516.98846  844.53473
 13)    0.520237 195.282150 0.000000  5.19478  3.19366  580.39441  619.68274
 14)    0.602754 106.890472 0.000012  3.38757  9.25344  466.30637  732.88861
 15)    0.421478 191.924759 0.000000  4.33987  2.65427  551.76666  532.90344
 16)    0.703031  33.467255 0.000000  1.34317 63.09134  295.94736  921.68030
 17)    0.328929  70.889549 0.000011  1.45948  9.40406  346.10699  492.49600
 18)    0.368603 132.485016 0.000060  2.80027  4.07531  460.29648  502.05032
 19)    0.439069  30.772434 0.000001  0.86269 44.68136  258.32669  660.77411
 20)    0.738818 154.090332 0.000000  5.51812  6.51063  565.85309  818.53223
 21)    0.308129 179.644821 0.000000  3.20272  2.14292  503.21408  427.49051
 22)    0.445688 153.749023 0.000000  3.71287  3.93280  509.26993  567.53284
 23)    0.288519 186.980835 0.000000  3.15419  1.88762  504.70557  405.93958
 24)    0.421008  69.434967 0.000006  1.73660 12.43421  361.36340  590.23657
 25)    0.476449 187.077835 0.000000  4.66667  3.12093  559.96246  583.94733
 26)    0.750772 160.908707 0.000000  5.80968  6.19497  577.95892  824.48865
 27)    0.603141 197.564240 0.000000  5.89113  3.63994  601.21942  688.99530
 28)    0.727499  40.298916 0.000000  1.63052 49.23381  322.46783  927.21381
 29)    0.749536 177.998032 0.000000  6.35398  5.30506  602.29999  815.12720
 30)    0.458599  30.759521 0.000000  0.89214 46.70675  260.69382  681.98431
 31)    0.440820  48.830963 0.000002  1.31124 22.23571  314.60492  632.40045
 32)    0.757005  74.974197 0.000000  2.94071 19.94248  421.86475  896.16071
 33)    0.661191 154.092651 0.000000  5.06049  5.82384  553.04108  755.23535
 34)    0.562687 147.456573 0.000000  4.28884  5.29576  525.27051  674.87878
 35)    0.398529 118.924271 0.000037  2.70045  5.19375  447.29044  537.12659
 36)    0.698850  85.290886 0.000000  3.10289 15.12932  437.79086  834.72968
 37)    0.569022 156.931808 0.000000  4.57588  4.87191  540.21692  676.10010
 38)    0.419369 110.029816 0.000026  2.62010  6.15206  437.67075  561.75238
 39)    0.619395 103.571518 0.000010  3.36345  9.97700  462.84756  749.89478
 40)    0.254020 145.111496 0.000168  2.27333  2.44191  442.44751  379.76648
 41)    0.686146  46.946667 0.000000  1.78724 36.80891  339.68875  875.05695
 42)    0.702958 156.939896 0.000000  5.39633  6.02326  564.32501  788.06769
 43)    0.411141 148.619888 0.000069  3.38155  3.81874  493.84186  537.13037
 44)    0.671093 176.531952 0.000000  5.78583  4.80782  586.73712  752.97729
 45)    0.525717 119.736664 0.000022  3.37229  6.78827  475.12216  656.12781
 46)    0.323726 151.894196 0.000118  2.86194  2.90607  474.22729  450.66794
 47)    0.663476  84.374069 0.000000  2.95084 14.59837  431.13733  804.75903
 48)    0.382964  51.526951 0.000003  1.23319 17.79214  312.40414  567.97369
 49)    0.459375 197.565643 0.000000  4.76392  2.76926  568.41254  565.56610
 50)    0.690213 177.411804 0.000000  5.94053  4.90813  591.35358  768.08405
 51)    0.333952 145.988678 0.000099  2.82940  3.18451  469.50580  462.79968
 52)    0.302949 177.848083 0.000218  3.13218  2.13919  499.36685  422.69839
 53)    0.689907 130.841187 0.000000  4.51503  7.79334  521.45355  791.88965
 54)    0.638884 134.475082 0.000000  4.35854  6.92051  519.08136  746.90863
 55)    0.650006 122.476387 0.000000  4.06122  8.11631  501.16907  763.50677
 56)    0.340853 191.176865 0.000000  3.66461  2.15747  527.23523  457.05573
 57)    0.620774  95.129494 0.000008  3.12109 11.37879  446.98267  757.60236
 58)    0.637174 197.961075 0.000000  6.15995  3.83445  608.55511  716.82190
 59)    0.580563  98.608284 0.000010  3.05956 10.07364  447.43503  719.07678
 60)    0.284390  60.076977 0.000009  1.12260 10.45017  313.02869  450.68423

max=0.000218 min=0.000000 avg=0.000020 sum=0.001214 no_mutation=0 no_cross=0

 FEED     0.302949
 SPEED  177.848083
 FITNESS  0.000218
```

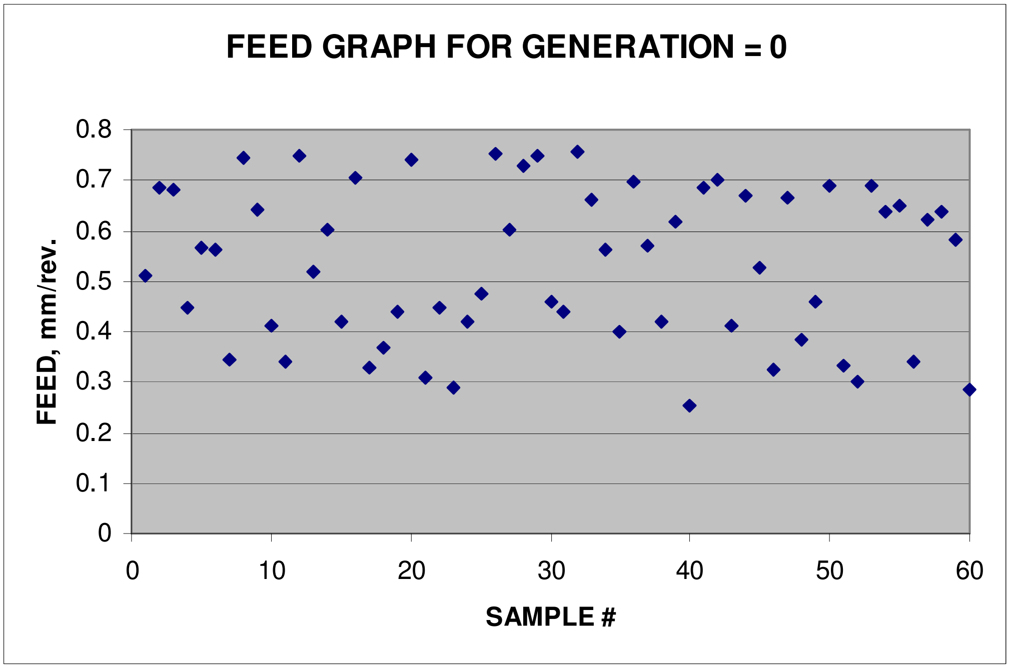


FIG. 5.5(a)

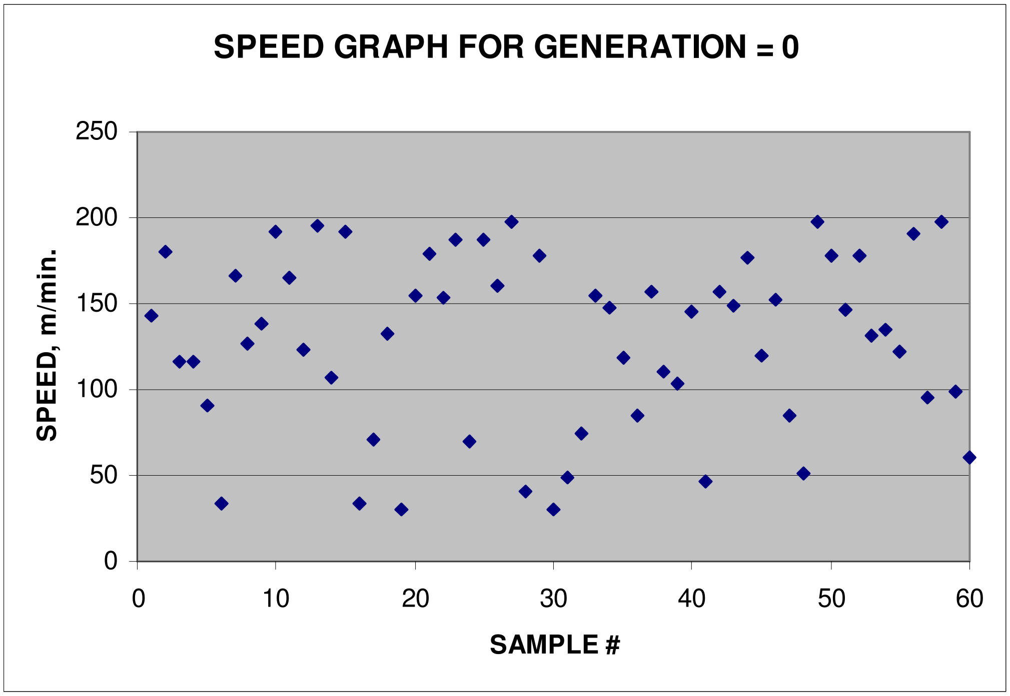


FIG. 5.5(b)

Fig. 5.5 Speed and feed graphs for starting generation with doc = 2.54mm

TABLE 5.5(b)

```
Accumulated Statistics for Generation ---> 240

*********************************************Calculated Constraints************
        FEED      SPEED    FITNESS    POWER   ROUGNESS    TEMP       FORCE
*******************************************************************************

  1)   0.254953 166.342361 0.521014  2.71700  1.81225  480.30862  372.59769
  2)   0.262269 178.028809 0.507461  2.81335  1.97273  483.51126  390.29193
  3)   0.254336 176.711060 0.507453  2.74384  1.78825  482.38657  372.61282
  4)   0.262269 176.700684 0.509370  2.78273  1.86917  483.36621  380.99033
  5)   0.262416 176.718658 0.509379  2.78420  1.86994  483.44290  381.14157
  6)   0.254477 172.054337 0.513767  2.56366  1.99864  467.61911  375.01410
  7)   0.254446 165.678223 0.521647  2.56470  1.99970  467.66995  375.15677
  8)   0.317979 173.132172 0.000183  3.17501  2.33939  498.81601  438.99414
  9)   0.264756 176.699875 0.509944  2.80328  1.88698  484.31232  383.60675
 10)   0.258921 178.708466 0.505712  2.78314  1.81380  484.34454  377.02673
 11)   0.258960 165.678329 0.000241  2.60013  2.03532  469.38058  379.97058
 12)   0.254417 178.448532 0.505026  2.74171  1.78607  482.29379  372.31467
 13)   0.255687 178.433792 0.505351  2.75218  1.79525  482.77563  373.66461
 14)   0.354479 165.678299 0.000007  3.72099  2.73063  521.95825  499.99963
 15)   0.271633 178.677322 0.508540  2.88870  1.90371  489.14548  390.36411
 16)   0.254479 166.074554 0.521192  2.57048  1.99271  468.14670  375.10104
 17)   0.255826 172.758301 0.513200  2.67440  1.88669  476.39893  375.03833
 18)   0.256441 176.345337 0.508481  2.72943  1.83307  480.71716  374.90967
 19)   0.254478 166.341064 0.520877  2.57419  1.98785  468.45819  375.03934
 20)   0.262040 176.366104 0.509793  2.77610  1.87292  482.89905  380.82254
 21)   0.254478 166.341064 0.520877  2.57419  1.98785  468.45819  375.03934
 22)   0.271825 176.697418 0.511513  2.86146  1.93761  486.96268  391.00626
 23)   0.254336 176.711060 0.507453  2.71700  1.81225  480.30862  372.59769
 24)   0.254483 165.678452 0.521658  2.56500  1.99998  467.68423  375.19595
 25)   0.256461 176.700684 0.507989  2.73455  1.82762  481.12671  374.85495
 26)   0.254336 176.711060 0.507453  2.71700  1.81225  480.30859  372.59766
 27)   0.262911 176.698135 0.509523  2.78801  1.87381  483.60864  381.66739
 28)   0.271825 176.365387 0.511998  2.85662  1.94316  486.58307  391.08075
 29)   0.254587 176.699860 0.507531  2.71894  1.81422  480.39453  372.86737
 30)   0.256463 165.678299 0.000246  2.58055  2.01561  468.43735  377.31064
 31)   0.254478 176.302002 0.508056  2.71252  1.81967  479.90274  372.83630
 32)   0.262415 166.403534 0.000238  2.63750  2.04893  471.52789  383.46979
 33)   0.398314 176.207367 0.000000  3.84531  2.85562  526.44177  515.95184
 34)   0.270831 123.842369 0.000091  2.07215  3.31358  419.74664  404.26477
 35)   0.262393 172.754486 0.514880  2.72774  1.93537  478.90582  381.99384
 36)   0.259790 175.410797 0.510598  2.74404  1.87217  480.95123  378.65659
 37)   0.264756 172.056625 0.516407  2.73690  1.96493  478.99240  384.64294
 38)   0.322414 177.362778 0.000000  3.27999  2.28668  505.27603  442.34155
 39)   0.254479 133.707352 0.000130  2.11486  2.77043  427.78397  383.42960
 40)   0.258293 174.034576 0.512121  2.71240  1.88375  478.80798  377.37439
 41)   0.264757 176.699875 0.509944  2.80328  1.88699  484.31241  383.60706
 42)   0.254479 176.690811 0.507517  2.71791  1.81359  480.34186  372.75409
 43)   0.254467 166.052368 0.521215  2.57009  1.99302  468.11633  375.09387
 44)   0.258921 176.698715 0.508585  2.75496  1.84525  482.07803  377.45892
 45)   0.254448 177.912628 0.505794  2.73456  1.79447  481.70462  372.46085
 46)   0.258797 178.067047 0.506608  2.77312  1.82286  483.57480  377.03311
 47)   0.383440 176.693405 0.000000  3.74211  2.73708  522.91034  501.77011
 48)   0.256463 176.355728 0.508472  2.72976  1.83307  480.73764  374.93121
 49)   0.258921 176.698715 0.508585  2.75496  1.84525  482.07803  377.45892
 50)   0.271825 176.697418 0.511513  2.86146  1.93761  486.96268  391.00626
 51)   0.254480 176.690811 0.507517  2.71792  1.81360  480.34222  372.75507
 52)   0.254479 165.678299 0.521657  2.56497  1.99996  467.68265  375.19202
 53)   0.254943 166.065933 0.521339  2.57402  1.99651  468.31387  375.59930
 54)   0.255827 172.758301 0.513200  2.67441  1.88669  476.39926  375.03931
 55)   0.254479 176.192764 0.508208  2.71101  1.82139  479.77972  372.86066
 56)   0.274169 176.684677 0.512032  2.88049  1.95460  487.81528  393.44971
 57)   0.254479 165.678299 0.521657  2.56497  1.99996  467.68262  375.19199
 58)   0.262415 176.366104 0.509880  2.77920  1.87562  483.04233  381.21771
 59)   0.254477 172.054337 0.513767  2.65362  1.88838  475.07074  373.75748
 60)   0.254479 165.678299 0.521657  2.56497  1.99996  467.68262  375.19199

max=0.521658 min=0.0000 avg=0.435470 sum=26.128193 no_mutation=17695 no_cross=4362

 FEED     0.254483
 SPEED  165.678452
 FITNESS  0.521658
```

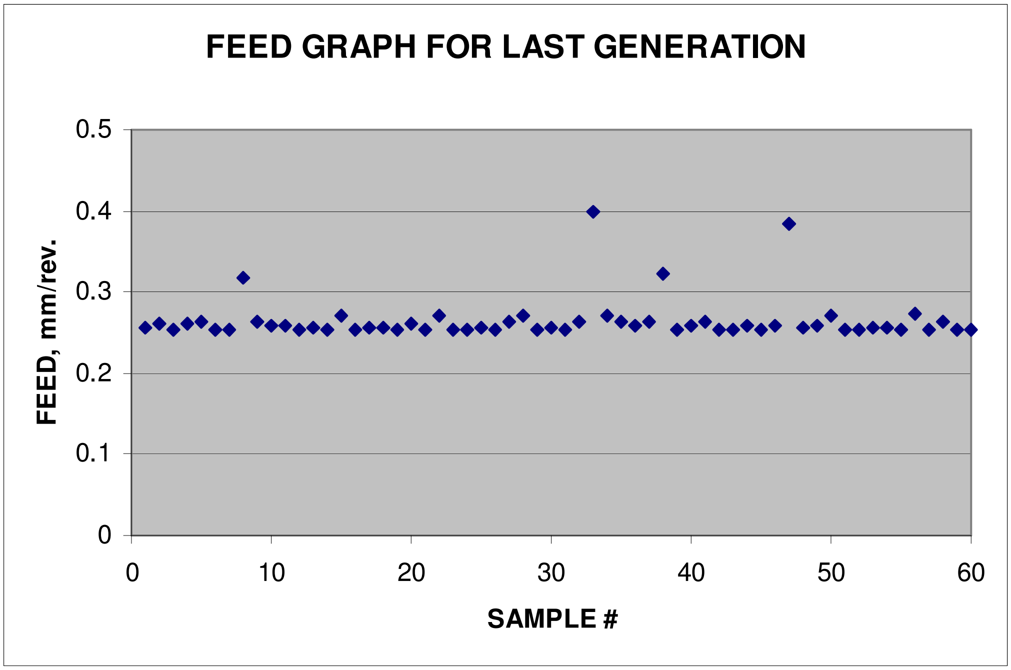


FIG. 5.5(c)

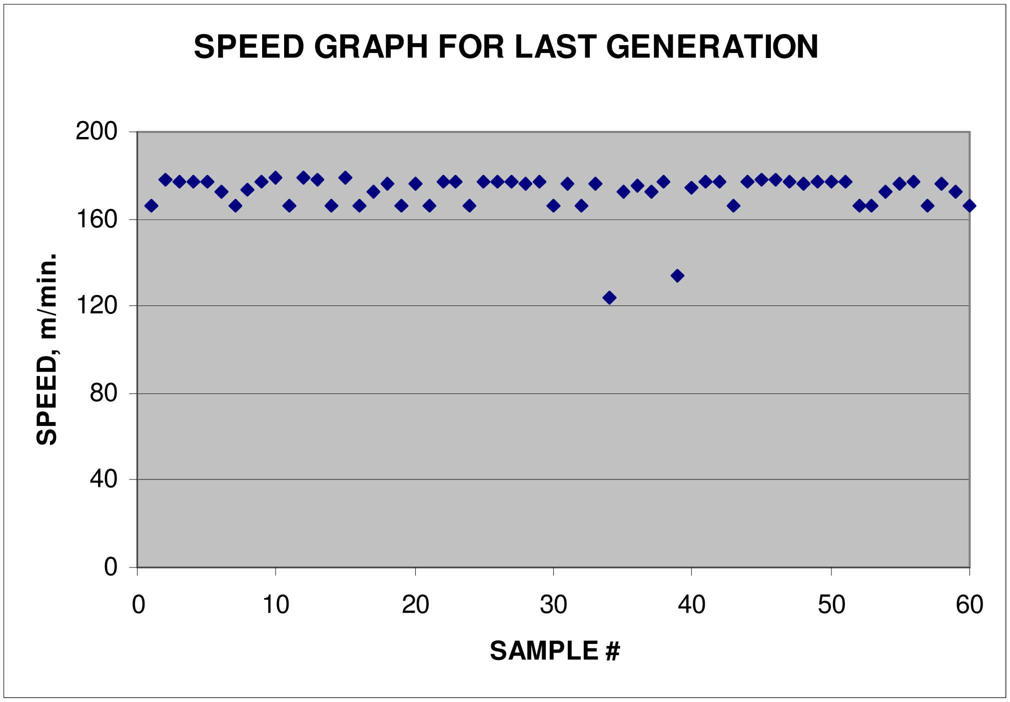


FIG. 5.5(d)

Fig. 5.5 Speed and feed graphs for last generation with doc = 2.54mm

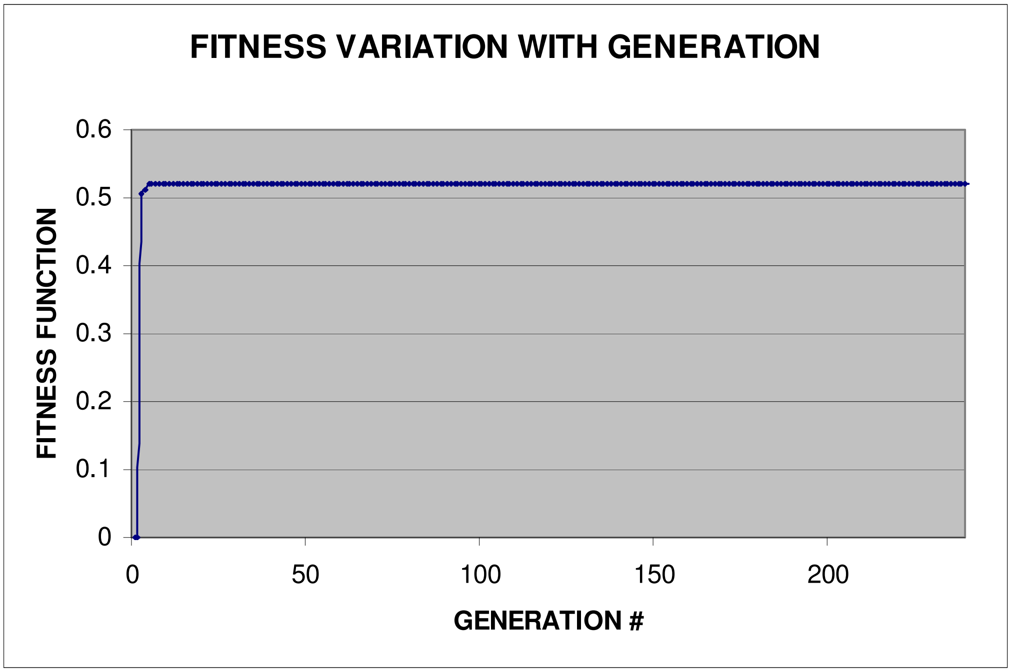


FIG. 5.5(e)

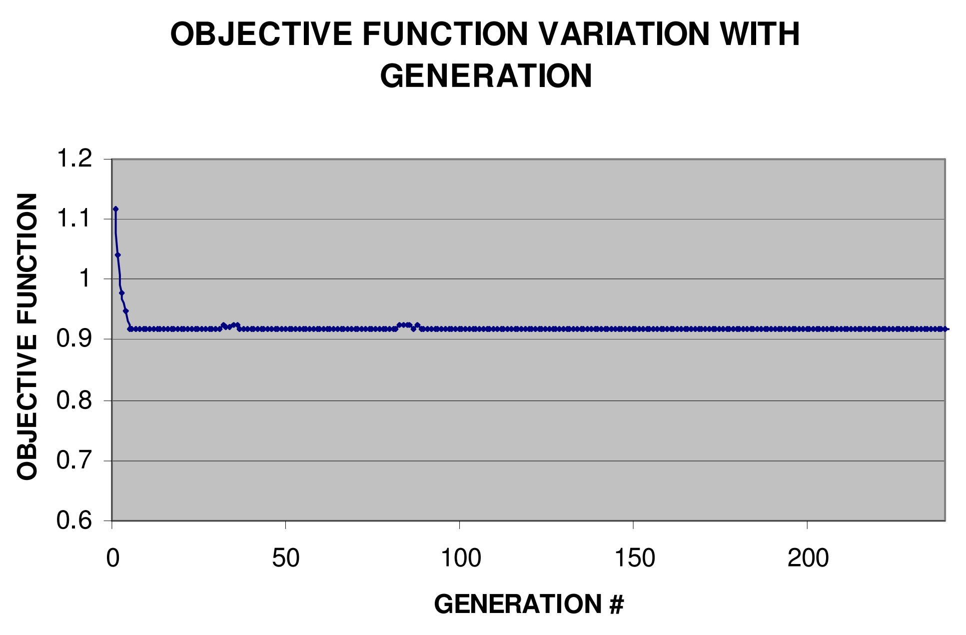


FIG. 5.2(f)

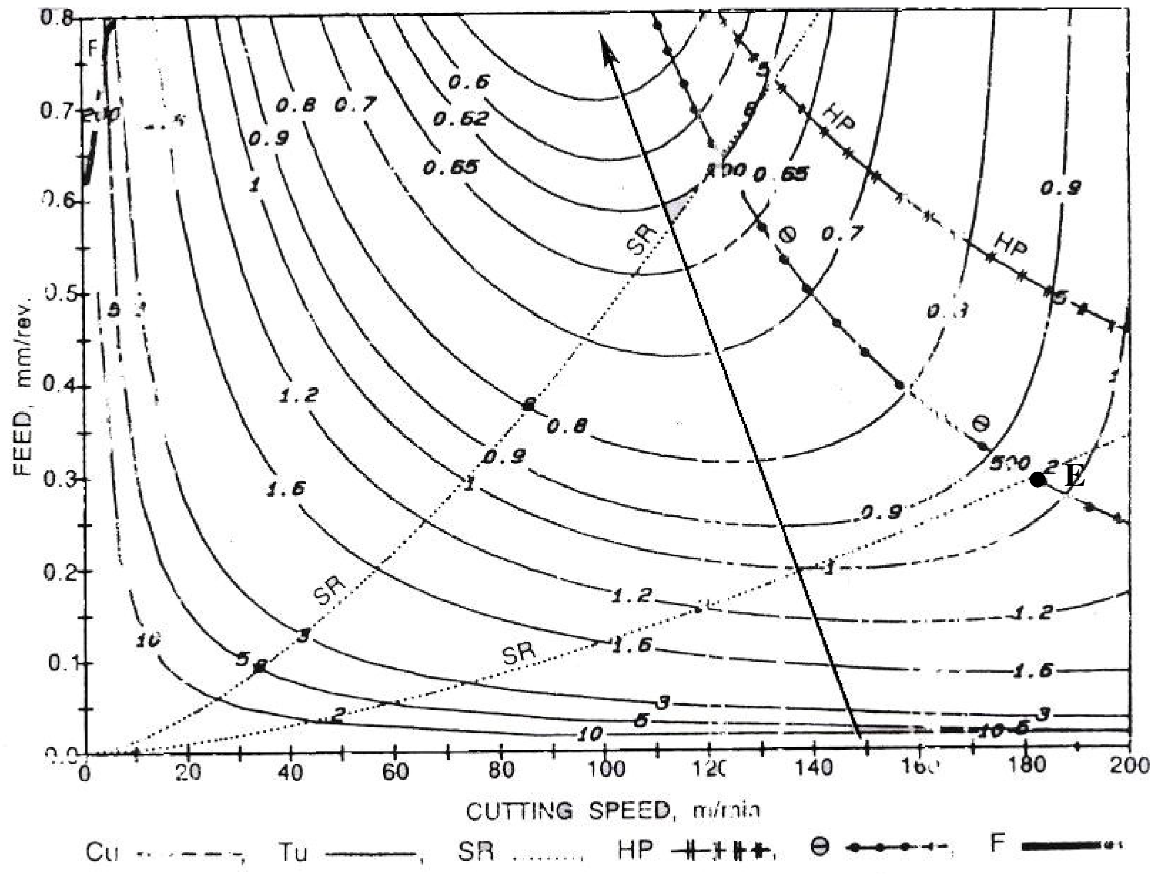


FIG. 5.5(g) Location of best result found by GA on the contours given by Agapiou [6], for doc = 2.54 mm and the constraints of power, force, surface roughness, and temperature.

In the section 5.6, the combined objective function is taken that includes both the total production cost and the total production time with weight coefficients 0.6 and 0.4 respectively, and depth of cut, doc = 5.08 mm having the constraints force, surface roughness, temperature and power.

Machining parameters used for the example are::

Parameter	**Value**
SR_{max}	8 μm
HP_{max}	5 KW
F_{max}	1200 N
T_{max}	500 deg C
λ	0.244

Fig. 5.6(e) and fig. 5.6(f), shows the variation of best fitness function and the objective function with generation, respectively. These figures clearly show how the values of both these functions get improved with generation after applying the GA operations. From fig. 5.6(f), the best value of the objective function is approximated to be 0.96076685. It is clear from table 5.6(b) that the constraints surface roughness = 7.98187 μm, force = 1002.70300 N and temperature = 429.58472 deg C have their extreme limits for the calculated best objective function value. From fig. 5.6(g), that we have from J. S. Agapiou [6], the value of objective function lies at point F (nearer to 0.9) that justifies our objective function value.

5.6 COMBINED OBJECTIVE FUNCTION with doc = 5.08mm

TABLE 5.6(a)

```
Accumulated Statistics for Generation ---> 0
*********************************************Calculated Constraints************
        FEED      SPEED    FITNESS    POWER   ROUGNESS    TEMP      FORCE
*******************************************************************************

  1)   0.612423 122.740631 0.000000  6.53275  9.06201  534.21497 1229.52637
  2)   0.678436 119.421120 0.000000  6.90330 10.47029  539.48773 1327.93176
  3)   0.273445 192.543091 0.000000  5.22325  2.03434  544.71033  654.70929
  4)   0.455292  63.933033 0.000003  2.88221 18.13459  382.96829 1059.44971
  5)   0.264219  76.250244 0.000015  2.20930  8.03401  367.88660  701.44403
  6)   0.650119  48.781765 0.000000  2.98313 39.11834  368.43069 1409.74121
  7)   0.433076 196.774963 0.000000  7.62429  3.12294  604.27960  911.73737
  8)   0.408484  78.017464 0.000000  3.16808 12.01642  406.87524  959.76056
  9)   0.339722 143.846878 0.000000  4.75875  3.94027  504.99344  789.24060
 10)   0.627434  80.237244 0.000000  4.54096 17.71715  450.16055 1306.36230
 11)   0.736122 186.309845 0.000000 10.97842  5.78025  658.95209 1346.80322
 12)   0.607295 154.015030 0.000000  7.96095  6.36392  585.66266 1194.27600
 13)   0.312628  95.397766 0.000000  3.08186  6.76701  418.48077  774.65448
 14)   0.358075 190.530197 0.000000  6.38519  2.70977  573.38794  796.91309
 15)   0.699571 116.513130 0.000000  6.91533 11.21008  537.41553 1361.19263
 16)   0.352511 163.008270 0.000000  5.48138  3.38135  535.88605  800.46680
 17)   0.698644 163.512863 0.000000  9.37179  6.68838  617.84869 1313.99219
 18)   0.726852 117.241928 0.000000  7.16488 11.53912  543.07697 1398.58960
 19)   0.263216 192.980759 0.000000  5.08058  1.95119  540.94476  636.71417
 20)   0.705009  88.909630 0.000000  5.45447 17.04012  481.27261 1406.86670
 21)   0.681566  66.276970 0.000000  4.07821 25.74292  422.94955 1414.26025
 22)   0.624260  44.120346 0.000000  2.64038 43.75054  350.22220 1382.86108
 23)   0.671521 102.112976 0.000000  5.94821 13.14777  504.58365 1339.17883
 24)   0.725819  79.240639 0.000000  5.03044 20.90057  461.59131 1453.70898
 25)   0.624508  66.033371 0.000000  3.79676 23.71192  414.67221 1327.89282
 26)   0.582016 194.911255 0.000000  9.51939  4.26314  639.65643 1130.72607
 27)   0.279201  56.053886 0.000005  1.74851 13.55525  327.00296  753.17596
 28)   0.288975 105.644295 0.000000  3.17716  5.35482  429.55847  724.18250
 29)   0.387278 186.852264 0.000000  6.66980  3.01985  578.06677  845.18927
 30)   0.382324 174.342209 0.000000  6.20391  3.31220  560.28607  843.23529
 31)   0.670497 100.350288 0.000000  5.84875 13.47974  500.79956 1340.06018
 32)   0.612311 168.627701 0.000000  8.69322  5.59082  608.97437 1190.43860
 33)   0.678965 199.990311 0.000000 10.98640  4.78551  667.24371 1261.06421
 34)   0.640633  41.220367 0.000000  2.53433 49.79124  342.20776 1418.79980
 35)   0.295305 176.637543 0.000000  5.13207  2.50538  534.06476  698.32422
 36)   0.514001 191.465591 0.000000  8.50237  3.86649  618.95789 1035.17371
 37)   0.340393 160.476776 0.000000  5.25924  3.34326  528.59454  781.66071
 38)   0.329614 158.465591 0.000000  5.07099  3.29962  522.35754  764.61200
 39)   0.432540 116.725510 0.000000  4.76054  6.89867  486.88077  960.40643
 40)   0.631769  68.513145 0.000000  3.96040 22.68150  422.11276 1334.07825
 41)   0.460225 110.510033 0.000000  4.75647  7.97882  482.11932 1010.17548
 42)   0.577692 122.926353 0.000000  6.25044  8.52649  528.13702 1178.39026
 43)   0.394405 175.138046 0.000000  6.38240  3.39371  564.95477  862.07220
 44)   0.715556  77.832420 0.000000  4.89523 21.17315  456.81143 1441.39258
 45)   0.551581  64.535622 0.000000  3.37584 21.67542  400.21024 1216.39380
 46)   0.687958  76.484352 0.000000  4.67326 20.90110  449.81314 1403.35254
 47)   0.475531 104.925789 0.000000  4.65693  8.92141  475.07175 1039.87097
 48)   0.263354  55.109398 0.000006  1.64525 13.11735  320.68817  723.17828
 49)   0.514130 180.512177 0.000000  8.06491  4.22975  604.16010 1041.55981
 50)   0.272018  60.149509 0.000007  1.82558 11.86289  335.02359  733.81616
 51)   0.291924 150.163879 0.000000  4.39467  3.16985  498.17111  704.00317
 52)   0.640047  89.955063 0.000000  5.11181 15.19166  474.01068 1310.09033
 53)   0.544707 178.785309 0.000000  8.36397  4.54831  608.97345 1087.16907
 54)   0.478318  76.765381 0.000000  3.53130 14.43017  417.66650 1077.87280
 55)   0.255900  78.322708 0.472239  2.20750  7.46931  369.55963  683.50269
 56)   0.431421 197.652527 0.000000  7.63206  3.08999  604.91095  908.80121
 57)   0.389283 170.275650 0.000000  6.15958  3.49593  556.92419  856.38147
 58)   0.386755  99.719345 0.000000  3.78624  7.83301  445.58746  899.81903
 59)   0.683114  38.688187 0.000000  2.51670 58.47891  337.72919 1495.98108
 60)   0.451550  62.037014 0.000003  2.78717 18.82708  377.51828 1056.34741

max=0.472239 min=0.000000 avg=0.007871 sum=0.472280 no_mutation=0 no_cross=0

 FEED     0.255900
 SPEED   78.322708
 FITNESS  0.472239
```

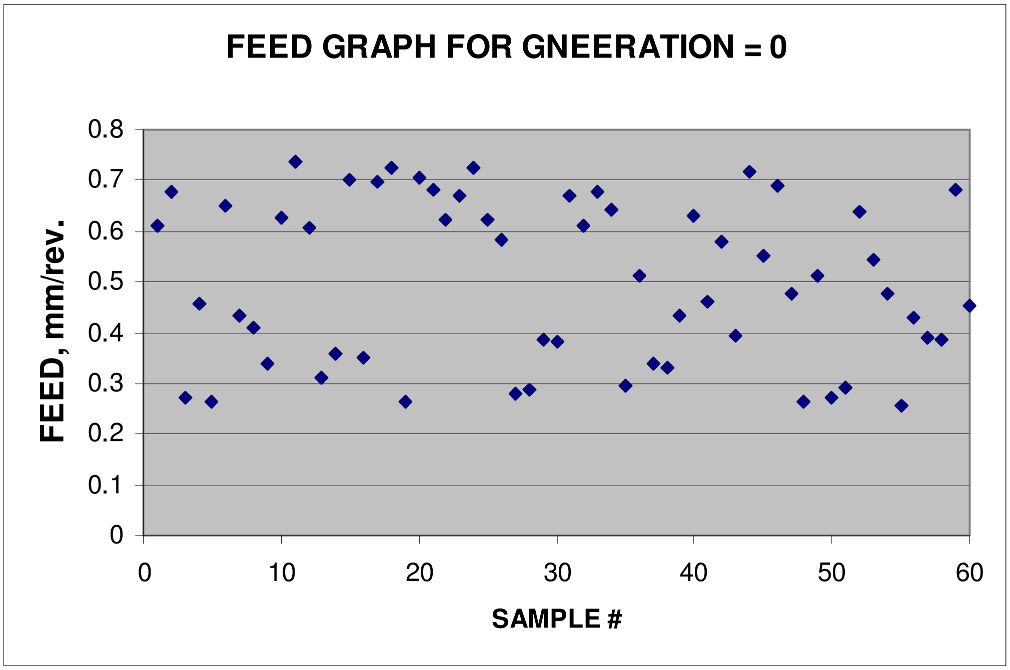


FIG. 5.6(a)

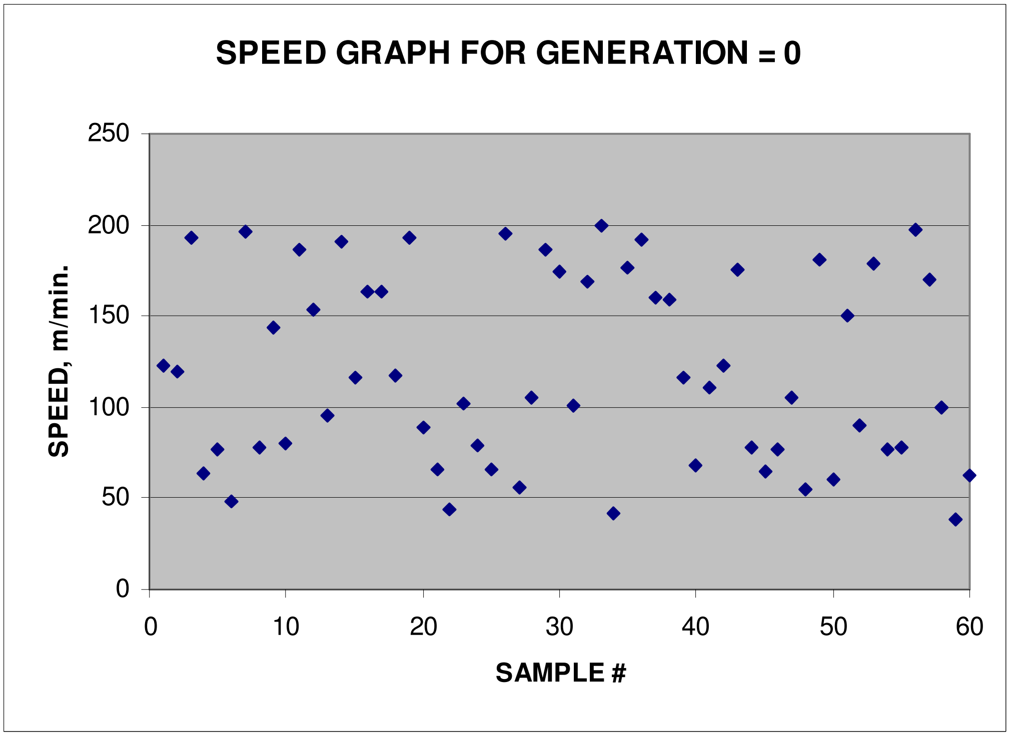


FIG. 5.6(b)

Fig. 5.6 Speed and feed graphs for starting generation with doc = 5.08mm

TABLE 5.6(b)

```
Accumulated Statistics for Generation ---> 240

*********************************************Calculated Constraints************
        FEED      SPEED    FITNESS    POWER   ROUGNESS    TEMP       FORCE
*******************************************************************************

  1)   0.311328  85.665413 0.509459  2.78831  7.93622  399.75247  780.78235
  2)   0.312612  90.499306 0.000061  2.79940  8.00041  400.09976  784.47687
  3)   0.286935  90.739250 0.000608  2.77527  8.01234  398.84055  781.64014
  4)   0.309826  85.168518 0.508495  2.76331  7.96793  398.37927  778.50842
  5)   0.286757  85.571175 0.497406  2.61254  7.31972  392.76245  735.68793
  6)   0.311450  41.224251 0.000002  1.44407 24.13330  293.86212  841.06909
  7)   0.311450  85.737190 0.509553  2.79127  7.92924  399.92514  780.93823
  8)   0.311330  85.665359 0.509460  2.78832  7.93628  399.75287  780.78595
  9)   0.302805  85.572502 0.505409  2.72593  7.73084  397.26059  765.31067
 10)   0.302805  85.333282 0.505270  2.71907  7.76381  396.79611  765.52795
 11)   0.286881  84.139290 0.496457  2.57403  7.51314  390.03564  737.17688
 12)   0.312612  84.420120 0.000015  2.76065  8.14844  397.65570  784.27814
 13)   0.312612  85.582314 0.510005  2.79484  7.98084  399.93411  783.19263
 14)   0.311454  85.737190 0.509555  2.79130  7.92934  399.92618  780.94525
 15)   0.281688  85.665359 0.494807  2.57900  7.17779  391.48053  726.15497
 16)   0.298927  85.415611 0.503439  2.69421  7.65277  395.88840  758.33435
 17)   0.312612  85.499306 0.509961  2.79240  7.99263  399.77200  783.26959
 18)   0.311082  85.665359 0.509346  2.78659  7.92993  399.68631  780.33490
 19)   0.311454  85.415611 0.509385  2.78187  7.97476  399.29877  781.24255
 20)   0.302892  85.333282 0.505312  2.71968  7.76606  396.82004  765.68805
 21)   0.312612  85.499306 0.509961  2.79240  7.99263  399.77200  783.26959
 22)   0.312612  85.582314 0.510005  3.24943  7.98187  429.58472 1002.70300
 23)   0.255185  85.739250 0.479759  2.38955  6.49138  383.61005  675.89508
 24)   0.373078  85.618568 0.000000  3.20938  9.52513  415.04495  890.27832
 25)   0.311454  85.737190 0.509555  2.79130  7.92934  399.92618  780.94525
 26)   0.311328  85.500641 0.509372  2.78349  7.95948  399.43103  780.93463
 27)   0.311546  85.623856 0.509538  2.69077  7.72258  395.47272  760.28381
 28)   0.311330  85.665359 0.509460  2.78832  7.93628  399.75287  780.78595
 29)   0.306779  85.760201 0.507398  2.75923  7.80667  398.70929  772.40820
 30)   0.286881  84.388313 0.496643  2.58088  7.47947  390.51804  736.95624
 31)   0.311327  85.483437 0.509363  2.78298  7.96190  399.39731  780.94971
 32)   0.310865  85.636757 0.509231  2.78424  7.92840  399.57230  779.96649
 33)   0.302805  85.572502 0.505409  2.72593  7.73084  397.26059  765.31067
 34)   0.310748  85.664673 0.509192  2.78424  7.92148  399.59537  779.72839
 35)   0.312612  85.499306 0.509961  2.79240  7.99263  399.77200  783.26959
 36)   0.306779  85.760201 0.507398  2.75923  7.80667  398.70929  772.40820
 37)   0.311314  85.483444 0.509357  2.78288  7.96155  399.39368  780.92499
 38)   0.311330  85.636787 0.509445  2.78749  7.94030  399.69717  780.81238
 39)   0.302752  88.376053 0.000000  2.80579  7.35987  402.63278  762.71942
 40)   0.311454 106.987183 0.000000  3.40684  5.66331  438.61160  763.62347
 41)   0.310834  85.665359 0.509232  2.78486  7.92358  399.61975  779.88385
 42)   0.309826  85.665314 0.508767  2.77781  7.89780  399.34866  778.05005
 43)   0.309826  85.621216 0.508743  2.77653  7.90398  399.26276  778.09064
 44)   0.311206  85.664665 0.509403  2.78744  7.93320  399.71826  780.56122
 45)   0.311330  85.001297 0.000016  2.76886  8.03071  398.45514  781.40167
 46)   0.311454  84.409065 0.000015  2.75235  8.11975  397.32568  782.18103
 47)   0.439550  42.999306 0.000001  1.96231 31.98939  321.77167 1075.10632
 48)   0.311546  85.168571 0.000016  5.07521  2.70499  529.33575  713.50610
 49)   0.311330  85.636787 0.509445  2.78749  7.94030  399.69717  780.81238
 50)   0.311454  85.621216 0.509494  2.78790  7.94567  399.70007  781.05231
 51)   0.301888  89.721672 0.000000  2.83788  7.17215  404.94019  759.97705
 52)   0.286877  85.126953 0.497169  2.60118  7.38095  391.94281  736.29895
 53)   0.302805  85.572830 0.505409  2.72594  7.73079  397.26123  765.31036
 54)   0.311546  85.623856 0.509538  2.78862  7.94767  399.73001  781.21796
 55)   0.302892  85.737190 0.505544  2.73126  7.71052  397.60394  765.32190
 56)   0.305501  85.333282 0.506557  2.73793  7.83320  397.53168  770.46283
 57)   0.311327  85.483437 0.509363  2.78298  7.96190  399.39728  780.94946
 58)   0.311454  85.737190 0.509555  2.79130  7.92934  399.92618  780.94525
 59)   0.301888  85.168571 0.504730  2.70794  7.76298  396.22476  763.99707
 60)   0.311546  85.167267 0.000016  2.77523  8.01252  398.83798  781.64124

max=0.510005 min=0.000000 avg=0.413583 sum=24.814966 no_mutation=21544 no_cross=5411

 FEED     0.312612
 SPEED   85.582314
 FITNESS  0.510005
```

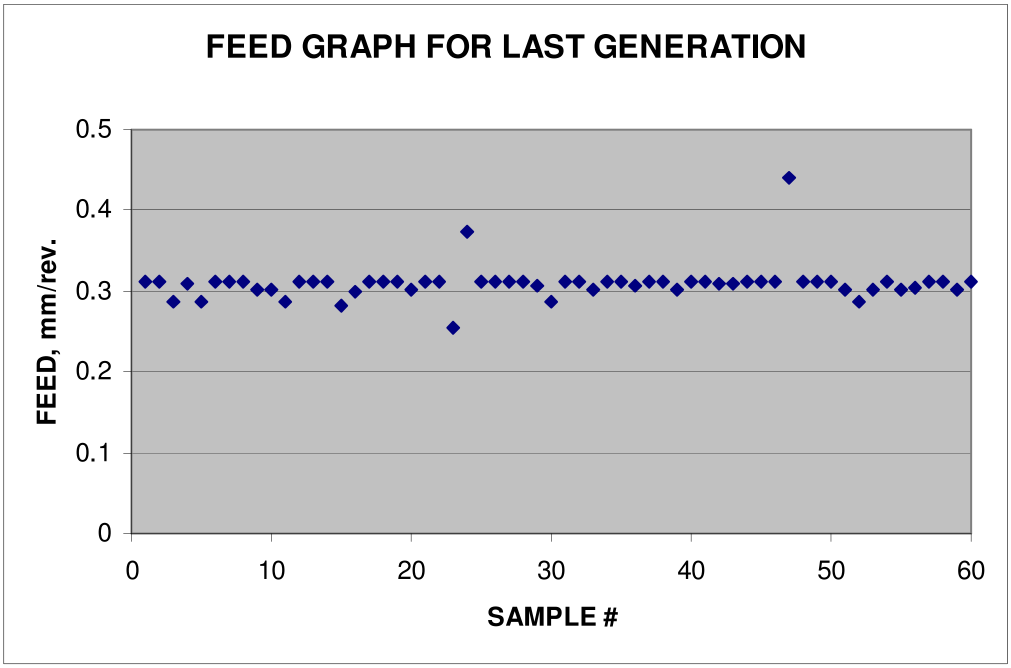


FIG. 5.6(c)

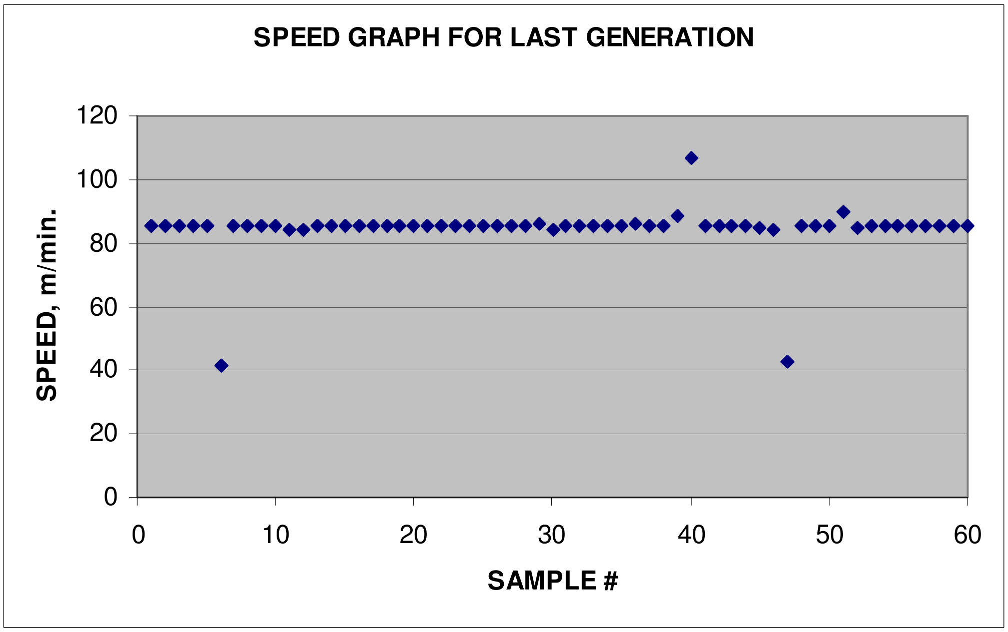


FIG. 5.6(d)

Fig. 5.6 Speed and feed graphs for last generation with doc = 5.08mm

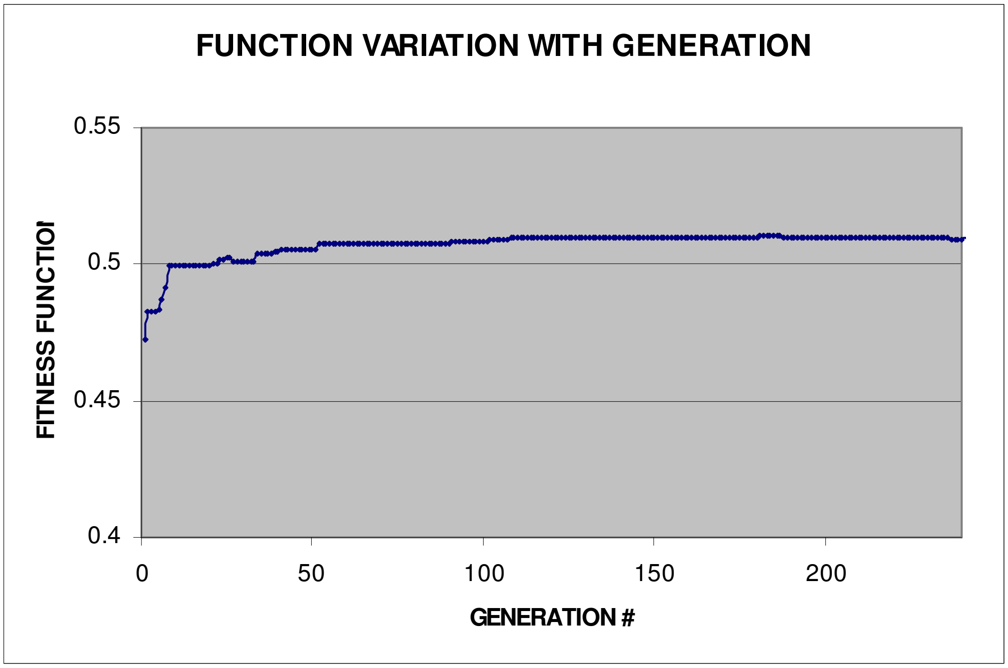


FIG. 5.6(e)

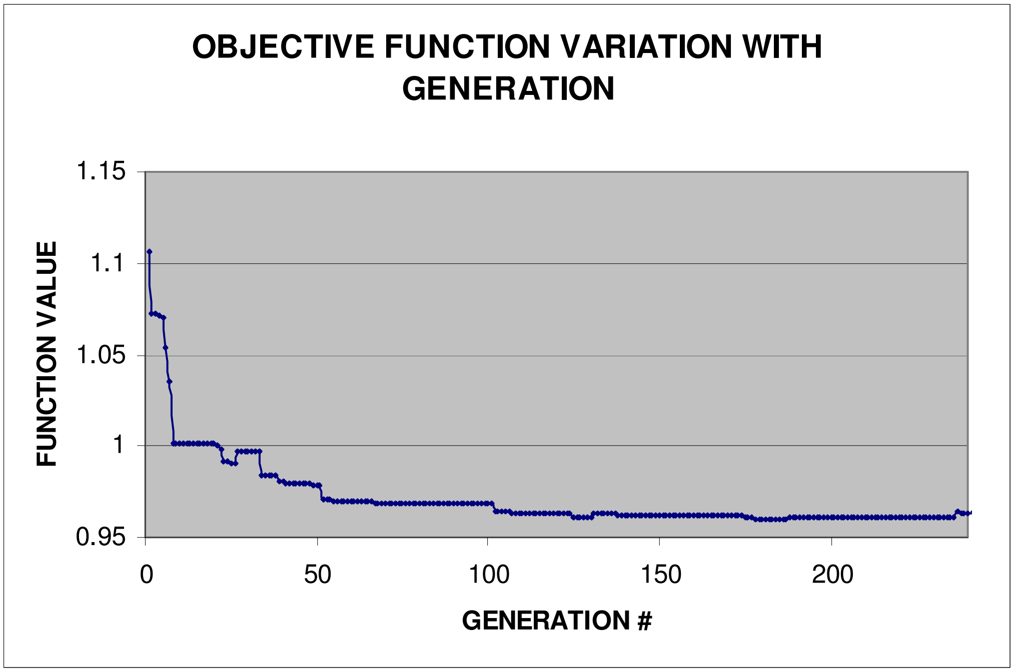


FIG. 5.6(f)

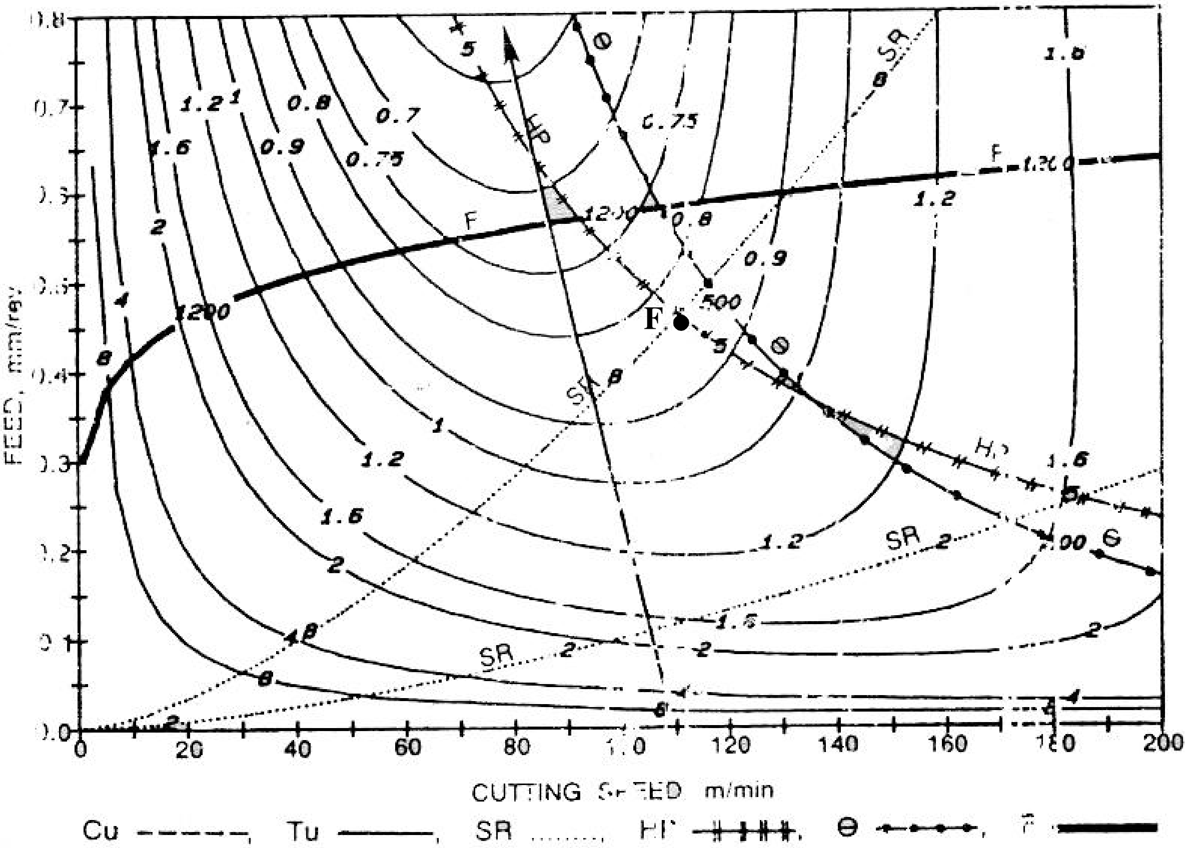


FIG. 5.6(g) Location of best result found by GA on the contours given by Agapiou [6], for doc = 5.08 mm and the constraints of power, force, surface roughness, and temperature.

CHAPTER 6

SCOPE OF FURTHER WORK

In the present work, the cutting conditions such as feed and speed are optimized based on the total production cost, total production time and combined of these two, as objective functions by taking into consideration the various constraints such as feed, cutting speed, power, cutting force, temperature, and surface roughness. Where as, there are other constraints also such as dimensional accuracy, rigidity and reliability of the system etc. that could also be considered.

Here the optimization is done only for the single-pass turning operation that can also be further extended to the multi-pass operations.

Tool selection, machine selection, process selection and tool path selection are the other important areas for optimization in process planning. Besides this, the sequencing and scheduling optimization are the other important areas for the GA implementation. Also the other advanced GA operators for reproduction, crossover and mutation could be applied to the optimization problem.

REFERENCES

1. Armarego and Brown, “The machining of metals”. Prentice Hall, Inc. Englewood Cliffs, New Jersey, (1980).

2. Yen, D. W. And Wright, R. K., “Adaptive control in machining-a new approach based on the physical constraints of tool wear mechanisms”. ASME J. Engg. Ind. Vol. 105, pp 31-38, (1983).

3. Chang T. C., Wysk, R. A., “An Introduction to automated process planning system”. Englewood Cliffs, NJ: Prentice Hall, (1985).

4. Ghosh Amitabh and Malik Ashok Kumar, “Manufacturing science”. New Delhi: Affiliated East-West Press (P) Ltd., (1985).

5. Sunderam R. Meenakshi and Cheng Tajen, “Microcomputer based process planning using geometric programming”. Int. J. Prod. Res., Vol. 24, pp 119-127, (1986).

6. J.S.Agapiou, “The optimization of machining operations based on a combined criterion; Part 1: The use of combined objectives in single pass operations”. Computers Ind. Trans. ASME 114, 500-507, (1992).

7. J.S.Agapiou, “The optimization of machining operations based on a combined criterion; Part 2: Multi-pass operations”. Computers Ind. Trans. ASME 114, 508-513, (1992).

8. R. Gupta, J. L. Batra and G. K. Lal, "Determination of optimal subdivision of depth of cut in multi-pass turning with constraints". Int. J. Prod. Res. 33(9), 2555-2565, (1995).

9. Rao S. S., "Engineering optimization theory and practice". Third Edition, New Delhi: New Age International (P) Ltd., (1996).

10. Juneja B. L. and Sekhon G. S., "Fundamentals of metal cutting and machine tools". Third Edition. New Delhi: New Age International (P) Ltd., (1998).

11. Hati, S. K. and Rao S. S., "Determination of optimal cutting conditions using deterministic and probabilistic approaches". Journal of Engineering for Industry, Transactions of ASME, pp. 210-217, (1998).

12. S. V. Bhaskara Reddy, M. S. Shunmugam and T.T. Narendran, "Operation sequencing in CAPP using Genetic Algorithms." International Journal of Production Research.. Vol. 37, No. 5, pp. 1063-1074, (1999).

13. Goldberg, D. E., "Genetic algorithms in search, optimization and machine learning". Addison-Wesley Longman, Inc., (1999).

14. Sonmez A. Ihsan and Baykasoglu Adil, "Dynamic optimization of multi-pass milling operations via geometric programmimg". Int. J. Machine Tools and Manufacture, Vol. 39, pp 297-320, (1999).

15. Chaudhury, S. K. and Appa Rao I. V. K., "Optimization of cutting parameters for maximizing tool life". Int. J. Machine Tools and Manufacture, Vol. 39, pp 343-353, (1999).

16. Dereli, T. and Filiz, H.I, “Optimization of process planning functions by genetic algorithm”. Computers and Industrial Engineering, Vol. 36, pp 281-308, (1999).

17. Sidhu T. S., “Optimal selection of machining parameters by using goal programming technique in single pass turning”. Sixteenth National Convention of Mechanical Engineers and National Seminar on Future Trends in Mechanical Engineering, Research and Development Department of Mechanical and Industrial Engineering. U.O.R., Roorkee, Sept. 29-30, (2000).

18. Sharma P. C., “A Text Book of Production Engineering”. Fifth Edition. New Delhi: S. Chand and Company Ltd., (2000).

19. Varghese Biju, “Optimal cutting conditions for a single pass machining operation using genetic algorithm”. Sixteenth National Convention of Mechanical Engineers and National Seminar on Future Trends in Mechanical Engineering, Research and Development Department of Mechanical and Industrial Engineering. U.O.R., Roorkee, Sept. 29-30, (2000).

20. Arya P. K. and Purohit K., “Optimization of Machining Parameters Using Genetic Algorithm”. Sixteenth National Convention of Mechanical Engineers and National Seminar on Future Trends in Mechanical Engineering, Research and Development Department of Mechanical and Industrial Engineering. U.O.R., Roorkee, Sept. 29-30, (2000).

21. Meng Q., “Calculation of optimum cutting conditions for turning operations using a machining theory”. Int. J. Machine Tools and Manufacture, Vol. 40, pp 1709-1733 (2000).

22. M. S. Shunmugam and S.V. Bhaskara Reddy, T. T. Narendran "Selection of optimal conditions in multi-pass face-milling operation using a genetic algorithm". Int. J. Machine Tools and Manufacture, Vol. 40, pp 401-414 (2000).

23. Nafis Ahmad, Dr. A.F.M. Anwarul Haque, Dr. A. A. Hasin, "Current Trend In Computer Aided Process Planning". Proceedings of the 7th Annual Paper Meet and 2nd International Conference. The Institution of Engineers, Bangladesh Mechanical Engineering Division, Paper No.10, pp. 81-92, (2001).

24. Nafis Ahmad and Dr. A.F.M. Anwarul Haque, "Optimization of process planning parameters for rotational components by genetic algorithms". Proceedings of the 7th Annual Paper Meet and 2nd International Conference. The Institution of Engineers, Bangladesh Mechanical Engineering Division, Section VI, pp. VII 227-233, (2001).

25. Y. V. Hui, "Optimal machining conditions with costs of quality and tool maintenance for turning". Int. J. Prod. Res., Vol. 39, pp. 647-665, (2001).