SUPERVISOR'S DECLARATION

We hereby declare that we have checked this project and in our opinion this project is satisfactory in terms of scope and quality for the award of the degree of Bachelor of Mechanical Engineering with Manufacturing

Signature
Name of Supervisor: MR. MOHD FADZIL FAISAE AB. RASHID.
Position:
Date:

Signature
Name of Panel: DR, WAN AZHAR BIN WAN YUSOFF.
Position:
Date:

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature
Name: MOHAMAD NAZRI BIN SEMOIN.
ID Number: ME06078
Date:

DEDICATION

To my beloved father,

Semoin bin Suradi

My beloved mother,

Suhawati binti Bibit

and

All my beloved family members

Thank you for every single thing

ACKNOWLEDGEMENTS

In the name of Allah, the Most Merciful and the Most Beneficent. It is with the deepest senses gratitude of the almighty that gives strength and ability to complete this thesis successfully.

First of all, I would like to dedicate my sincere appreciation to my supervisor, Mr Mohd Fadzil Faisae bin Ab. Rashid and also lecturers at Universiti Malaysia Pahang for allowed taking me under their supervision. All of them have given me critics, encouragement, guidance, and valuable advices in order to complete this project. Without their continued support and interest, this thesis would not have been the same as presented here.

My fellow colleagues should also be recognized for their support and friendship. My deeply thanks also goes to others who have provided assistance at various occasions that invite whether direct or indirectly in the completion of my project. Last, but certainly not least, my special thanks also extends to my family for the continual encouragement and support.

ABSTRACT

This project proposed a new optimization technique based on the ant colony algorithm for solving single-pass turning optimization problems. The cutting process has focus on roughing stages. There are enough handbooks to provide recommended cutting parameters and not consider the economic aspects of machining. The cost of machining on these machines is sensitive to the machining variable. The project objectives are to develop Ant Colony Optimization (ACO) algorithm for CNC turning process and to optimize turning parameters for minimized production cost per unit. Method used for this project is Ant Colony Optimization. This method consists of many steps will elaborate detail in this thesis. The machining parameters are determined by minimized production cost per unit, subject to various practical machining constraints. The results indicate that the proposed ant colony framework is effective to optimized turning parameter. Lastly, ACO algorithm was successfully optimize depth of cut, cutting speed, feed rate and minimized production cost per unit.

ABSTRAK

Projek ini mencadangkan teknik baru berdasarkan kepada “Ant Colony Optimization” untuk meyelesaikan masalah permulaan untuk mengunakan mesin larik. Process pemotongan hanya difokuskan kepada process permulaan. Terdapat pelbagai buku-buku panduan untuk proses pemotongan yang mencadangkan pembolehubah permotongan dan tidak mempertimbangkan aspek ekonomi pada mesin. Kos mesin ini adalah sensetif dengan pembolehubah mesin. Objektif projek ini adalah untuk menghasilkan “Ant Colony Optimization” untuk proses mesin larik dan mengurangkan kadar kos seunit. Kaedah yang digunakan untuk projek ini adalah “Ant Colony Optimization”. Keadah ini merangkumi pelbagai langkah yang telah ceritakan secara lebih lanjut di dalam laporan ini. Pembolehubah untuk mesin digunakan untuk mengurangkan kadar kos untuk satu unit, mengikut kepada pelbagai pembolehubah yang tetap. Keputusan yang diperolehi akan mencadangkan bahawa “Ant Colont optimization” ini bersesuaian untuk mengoptimumkan kadar pembolehubah untuk mesin larik. Akhirnya, “Ant Colony Optimization” ini telah berjaya mengoptimumkan kedalaman permotongan, kelajuan permotongan, kadar suapan dan megurangkan kadar kos seunit.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

D	Diameter of workpiece (mm)
L	Length of the workpiece (mm)
V	Cutting speed (m/min)
f	Feed rate (mm/rev)
$f_{\min}, f_{\max}$	Minimum and maximum allowable feed rates
R_a	Surface roughness (μm)
$R_{a,\max}(r), R_{a.\max}(f)$	Maximum surface roughness of rough and finish cut, respectively
P	Power of the machine (kW)
F	Cutting force (N)
θ	Temperature of tool-workpiece interface (^{o}C)
doc	Depth of cut (mm)
$doc_{\min}(r), doc_{\max}(r)$	Minimum and maximum allowable depth of cut (rough)
$doc_{\min}(f), doc_{\max}(f)$	Minimum and maximum allowable depth of cut (finish)
a_1, a_2, a_3, K	Constant used in tool life equation
T	Tool life (min)
t_m	Machining time (min)
t_{cs}	Tool change time (min/edge)
t_R	Quick return time (min/pass)
t_h	Loading and unloading time (min/pass)
T_u	Total production time (min)
C_o	Operating cost (RM/piece)
C_t	Tool cost per cutting edge (RM/edge)
C_T	Total production cost (RM/piece)
C_P	Power cost (RM/min)
P1, P2	Parent 1, Parent 2

m	Level of mutation
r	Random number
X	Fitness value
ph	Pheromone value
age	Age
ph_{ave}	Pheromone average
$\lim step$	Limiting step
X_{inew}, X_{iold}	Fitness value iteration new, fitness value iteration old
ph_{inew}	Pheromone value iteration new
C_w	Workpiece cost (RM)

LIST OF ABBREVIATIONS

CNC	Computer Numerical Control
ACO	Ant Colony Optimization

CHAPTER 1

INTRODUCTION

1.1 PROJECT BACKGROUND

Machining parameters optimization has significant practical importance, particularly for operating Computer Numerical Control (CNC) turning machines. Due to the high cost of these machines, an economic analysis needs to be performed to operate them as efficiently as possible in order to obtain the required return on investment. Because the costs of machining on these machines are sensitive to the machining variable, the optimum values must be determined before a part is put into production. The operating parameters in this context are cutting speed, feed rate, depth of cut, and so on that do not violate any of the constraints that may apply to the process and satisfy the objective criterion, such as minimum time, minimum production cost, or maximum production rate (Saravanan, 2006).

The analysis will be done by using Ant Colony Optimization (ACO) method. This method is a metaheuristic approach to tackling a hard problem that was first proposed in the early 1990s by Dorigo, Maniezzo and Colorni. Fascinated by the ability of the almost blind ants to establish the shortest route from their nests to the food source and back, researchers found that these ants secrete a substance called pheromones and use its trails as a medium for communicating information among themselves. Also, they are capable of adapting to change in the environment, such as finding the new shortest path when the old one is no longer available due to new obstacle (Wei Gao, 2007).

The purpose of this study is to developed Ant Colony Optimization (ACO) algorithm for CNC turning process. The main objective is to optimize CNC turning parameters by minimizing production cost per unit.

The analysis will be done by collecting data. Then the data will be solving by applying Ant Colony Optimization method that developed in Matlab software. Nowadays, this technique is used to make sure the selection of machine parameters can minimize production cost per unit.

1.2 PROBLEM STATEMENT

In an early work, analysis of single and multi-pass turning under practical constraints has been done using minimum production cost or time criteria. The output of the product is usually high in order to increase the interest of the industry. However it is not easy to achieve that goal if there is no well plan by the industry.

Optimization of operating parameters is an important step in machining, particularly for operating CNC machine tools. Although there are enough handbooks to provide recommended cutting parameters, they do not consider the economic aspects of machining (Vijayakumar et. al, 2003).

Due to the high cost of these machines, an economic need exists to operate them as efficiently as possible to obtain the required return on investment. Because the cost of machining on these machines is sensitive to the machining variable, the optimum values must be determined before a part is put into production (Saravanan, 2006).

Selecting proper values for machining parameters such as cutting speed, feed rate, and depth of cut directly affects the machining economics in metal cutting process. Several cutting constraints must be considered in machining operations. A turning cutting operation involves several roughing cuts and a finishing cut. That makes the problem of determining the optimal cutting conditions more difficult and complicated.

Machining parameters can be determined based on the machine operator's experience or by following the cutting handbook supplied by the equipment manufacturer. However, those data are not guaranteed to be optimal or even good for a particular cutting environment (Yi-Chi Wang, 2007).

Therefore, developing optimization algorithm for single-pass turning operations has become a useful tool to optimize turning parameters for minimizes production cost per unit.

1.3 PROJECT OBJECTIVES

The purposed of these projects are to study and analyzed the CNC turning machines parameters and find the suitable value. The objectives of the project are:

1. To develop Ant Colony Optimization (ACO) algorithm for CNC turning process.
2. To optimize turning parameters for minimized production cost per unit.

1.4 PROJECT SCOPES

This scope is created to make sure this project running well in the limited boundary. The scopes of this project are:

1. The algorithm is developed for single past turning.
2. This project considers three main parameters such as feed rates, cutting speed, and depth of cut that mostly used in previous research.
3. All of the constant parameters are adapted from references. For optimization with difference tool material and machine, it must be appropriate with the difference parameter value.
4. The algorithm is developing by using Matlab software.
5. Only consider roughing cut.

1.5 PROJECT PLANNING

The planning for “Optimization Turning Parameters Using Ant Colony Optimization” is presented in this section. This planning consists of Flow chart PSM, Gantt chart PSM 1 and Gantt chart PSM 2, which is shown in Appendix B and Appendix C.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Optimization of operating parameters is an important step in machining, particularly for operating Computer Numerical Control (CNC) machine tools. Although there are enough handbooks to provide recommended cutting parameters, they do not consider the economic aspects of machining. Machining parameters problem have been dealt with several researchers.

This chapter introduce to the step of optimization turning parameters using ant colony optimization. Now days, CNC machine is commonly used in industry. The operation of this machine is an expensive because it has many parameters to consider. However, the optimizations technique can be used to minimize production cost per unit.

2.2 TURNING PROCESS

A Lathe produces parts by "turning" rod material and feeding a single-point cutter into the turning material. Cutting operations are performed with a cutting tool fed either parallel or at right angles to the axis of the workpiece. The tool may also be fed at an angle relative to the axis of the workpiece for the machining tapers and angles. The workpiece may originally be of any cross-section, but the machined surface is normally straight or tapered. Have many possible shape can produce in CNC turning such as variety of plain, taper, contour, fillet and radius profiles plus threaded surfaces. CNC turning also can be used to create shafts, rods, hubs, bushes and pulleys.

Figure 2.1 CNC turning machine.

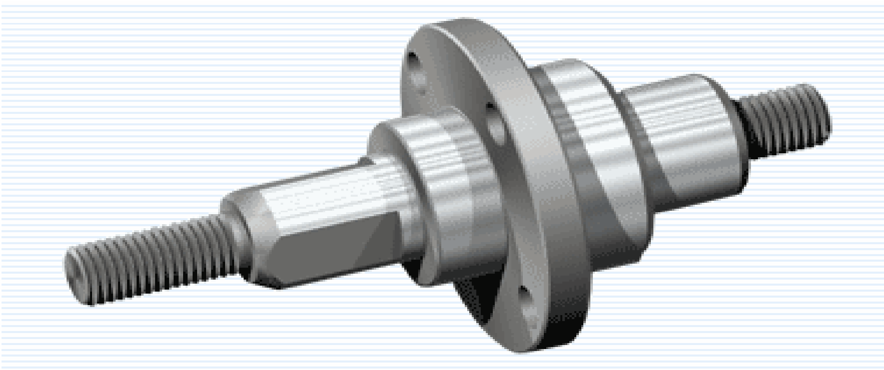


Figure 2.2 CNC turning possible shapes.

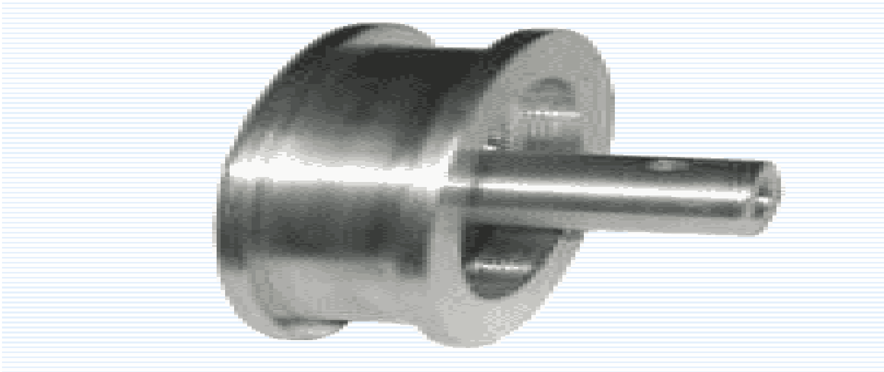


Figure 2.3 CNC turning example part.

2.2.1 Important to optimize turning parameters

Selecting proper values for machining parameters such as cutting speed, feed rate, and depth of cut directly affects the machining economics in metal cutting process. Several cutting constraints must be considered in machining operations. In turning operations, a cutting process can possibly be complicated with a single pass or by multiple passes. Multi pass turning is preferable over single pass turning in the industry for economic reasons. That makes the problem of determining the optimal cutting conditions more difficult and complicated. Machining parameters can be determined base on machine operator's experience or by following the cutting handbook supplied by the equipment manufacturer. However, those data are not guaranteed to be optimal or even good for a particular cutting environment. Other wise, developing mathematical models for single pass turning operations has become useful tool for determining the optimal cutting conditions (Yi-Chi Wang, 2007).

2.3 PREVIOUS RESEARCH OPTIMIZATION TURNING PARAMETERS

Optimization of operating parameters is an important step in machining, particularly for operating Computer Numerical Control (CNC) machine tools. Although there are enough handbooks to provide recommended cutting parameters, they do not consider the economics aspects of machining. Machining parameters optimization problem have been performed with by several researchers.

New optimization techniques based on the ant colony algorithm for solving multi-pass turning optimization problems are proposed. The cutting process has roughing and finishing stages. The machining parameters are determined by minimizing the unit production cost, subject to various practical machining constraints. In this paper, the Ant Colony Algorithm (ACO) algorithm is completely generalized and problem independent so that it can be easily modified to optimize this turning operation under various economic criteria, and numerous practical constraints (Vijayakumar et. al, 2003).

An article by Vijayakumar et al. [Optimization of Multi-pass Turning Operations Using Ant Colony System] proposed an ant colony optimization methodology for determining the machining parameters in a multi-pass turning operation model. By using the problem of Chen and Tsai [A Simulated Annealing Approach for Optimization of Multi-pass Turning Operations], they concluded that their ant colony approach outperformed the other optimization techniques proposed by other researchers. This journal discusses an illustrative multi-pass turning problem, which was used in several literatures and demonstrates that the optimal solution as found by Vijayakumar et al. is not valid (Yi-Chi-Wang, 2007).

Machining parameters optimization has significant practical importance, particularly for operating CNC machines. Because the cost of machining on these machines is sensitive to the machining variables, the optimum value must be determined before a part put into production. At the end of this analysis, has presented the initial simplex informed by considering the minimum limit of speed and feed rate. The accuracy of this result is dependent upon the chosen initial simplex. The results are obtained for the following four simplexes and the best one is selected (Saravanan, 2006).

In the 1997, analysis of [Design Optimization of Cutting Parameters for Turning Operations Base on Taguchi Method] is used to find the optimal cutting parameters for turning operations. As shown in this study, the Taguchi method provides a systematic and efficient methodology for the design optimization of the cutting parameters with far less effect than would be required for most optimization technique (W. H. Yang, Y. S. Tarng, 1997).

An optimization analysis, strategy for the selection of economic cutting conditions in single pass turning operations are presented using a deterministic approach. From this paper, the detailed optimization analysis assisted by the feed-speed diagrams has provide an in-depth understanding of economic characteristics and the influence of the constraint and machining performance data, which was resulted in a clearly defined optimization strategy that ensures the global optimum solution (Wang, 2002).

Therefore, summary from previous research optimization turning parameter, this is shown in Table 2.1 below.

Table 2.1: Summary from previous research optimization turning parameter.

Author	Journals / books	Year	Method	Parameters considered
R. Saravanan	Optimization of operating parameters for CNC machine tools	2006	Nelder-mead simplex method	Feed rate, cutting speed, depth of cut
K. Vijayakumar, G. Prabhaharan, P.Asokan, R. Saravanan	Optimization of multi-pass turning operation using ant colony method	2003	Ant colony optimization	Cutting speed, feed rate, depth of cut, number of rough cuts
Yi – Chi Wang	A note on optimization of turning operations using ant colony method	2007	Ant colony optimization	Cutting speed, feed rate, depth of cut, tool life, cutting force, cutting power, surface roughness
W. H. Yang, Y. S. Tarng	Design optimization of cutting parameters for turning operation based on the Taguchi method	1997	Taguchi method	Cutting speed, feed rate, depth of cut
J. Wang, T. Kuriyagawa, X. P. Wei, D. M. Guo	Optimization cutting condition for single-pass turning operation using a deterministic approach	2002	Deterministic approach	Cutting speed, feed rate, cutting power, surface roughness, tool life

From Table 2.1, three main parameters that were considered in previous studies were feed rate, cutting speed and depth of cut. Therefore, in this study these three main parameters will be considered.

2.3.1 Operating parameters

2.3.1.1 Feed rate

The maximum allowable feed has pronounced effect on both optimum spindle speed and production rate. Feed changes have a more significant impact on tool life than depth of cut change. The system energy requirement reduces with feed because the optimum speed becomes lower. Therefore, the largest possible feed consistent with allowable machine power and surface finish is desirable for a machine to be fully utilized. Obtaining much higher metal removal rates without reducing tool life is often possible by increasing the feed and decreasing the speed. In general, the maximum feed in a roughing operation is limited by the force that the cutting tool, machine tool, workpiece, and fixture are able to withstand. The maximum feed in a finish operation is limited by the surface finish requirement and often can be predicted to a certain degree based on the surface finish and tool nose radius.

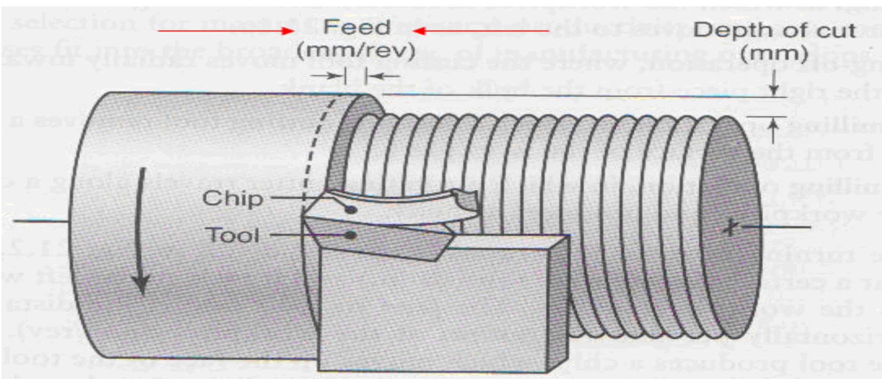


Figure 2.4 Feed rates (Saravanan, 2006).

2.3.1.2 Cutting speed

Cutting speed has a greater effect on a tool life than either depth of cut feed, when compared with depth of cut and feed, the cutting speed has only a secondary effect on chip breaking when it varies in the conventional speed ranges. Certain combination of speed, feed, and depth of cut are preferred for easy chip removal and are dependent mainly on the type of tool and workpiece material. Charts providing the feasible region for chip breaking as a function of feed versus depth of cut are sometimes available by the tool manufacturers for a specific insert or tool and can be incorporated into the optimization system.

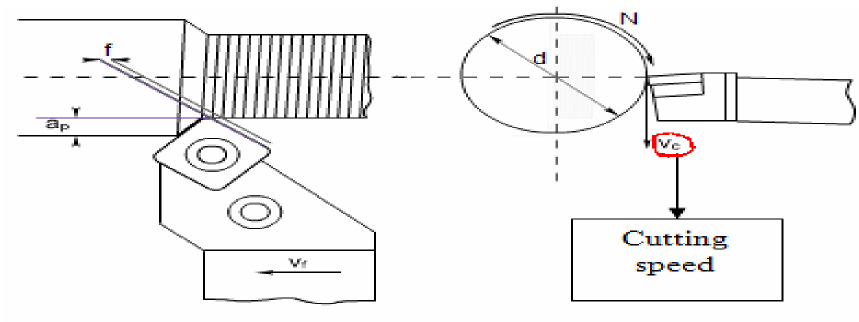


Figure 2.5 V_f = cutting speed (Saravanan, 2006).

2.3.1.3 Depth of cut

Allocation of depth of cut is a major task in machining process planning since the total depth of cut imposes the greatest constraint in single-pass turning operations, among all other constraints. It is generally done by assuming that all passes, or all passes except the finish pass, have equal depth of cut. This usually gives sub-optimal result.

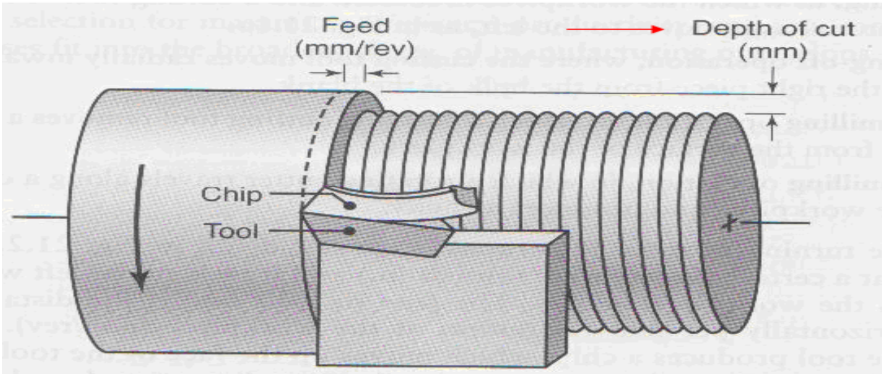


Figure 2.6 Depth of cut (Saravanan, 2006).

2.4 ANT COLONY OPTIMIZATION (ACO)

Ant colony optimization is a metaheuristic approach to tackling a hard CO problem that was first proposed in the early 1990s by Dorigo, Maniezzo and Colorni. Fascinated by the ability of the almost blind ants to establish the shortest route from their nests to the food source and back, researchers found that these ant secrete a substance called pheromones and use its trails as a medium for communicating information among themselves. Also, they are capable of adapting to changes in the environment, such as finding a new shortest path when the old one is no longer available due to a new obstacle in (Figure 2.7), ants are moving on a straight line that connects a food source to their nest.

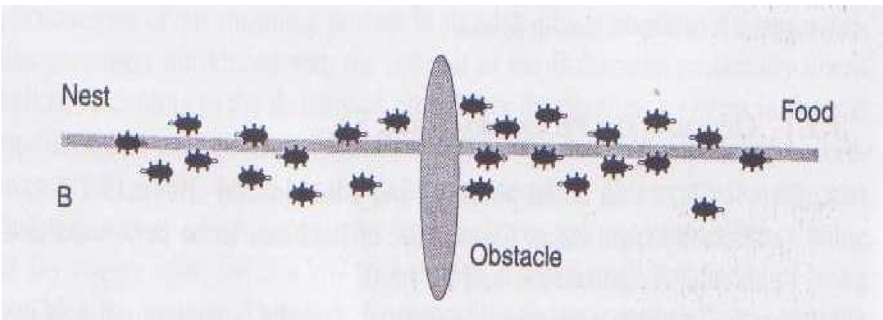


Figure 2.7 Real ants follow a path between nest and food source (Saravanan, 2006).

The primary, well known means for ants to form and maintain the line is a pheromone trail, Ants deposit a certain amount of pheromone while walking and each ant probabililistically prefers to follow a direction rich in pheromone. This elementary behavior of real ants can explain how they can find shortest path that reconnects a broken line after the sudden appearance of an unexpected obstacle in the initial path (Figure 2.8).

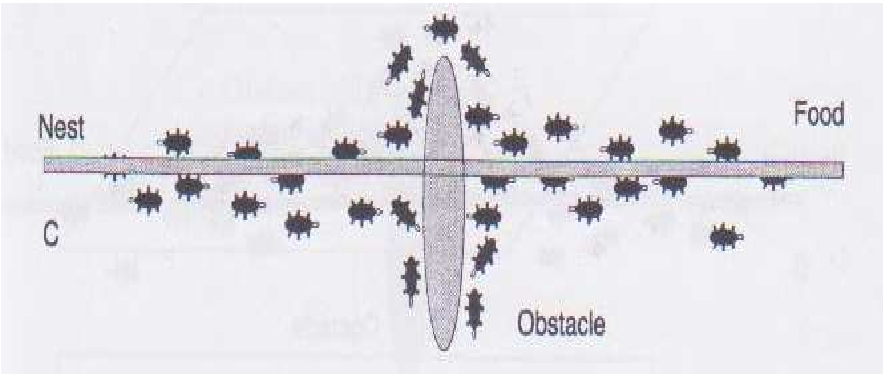


Figure 2.8 An obstacle appears on the path: Ants choose whether to turn left or right (Saravanan, 2006).

Once the obstacle has appeared, ant's right in front of the obstacle cannot continue to follow the pheromone trail in the straight line. In this situation, some ants choose to turn right and other choose to turn left (Figure 2.9).

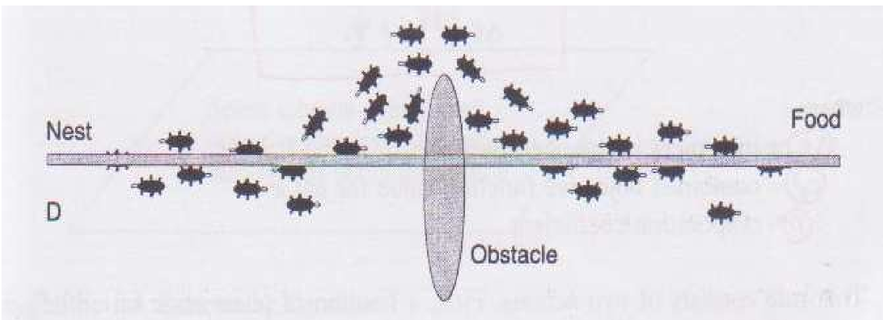


Figure 2.9 Pheromone is deposited more quickly on the shorter path (Saravanan, 2006).

The ants that choose, by chance, the shorter path around the obstacle will more rapidly reconstitute the interrupted pheromone trail compared to those that choose the longer path. Thus the shorter path will receive a greater amount of pheromone per unit and, in turn, a larger number of ants will choose the shorter path. Due to this positive feedback (autocatalytic) process, all the ants will rapidly choose the shorter path (Figure 2.10). The most interesting aspect of this autocatalytic process that finding the shortest path around the obstacle seems to be an emergent property of the interaction between the obstacle shape and ants' distributed behavior.

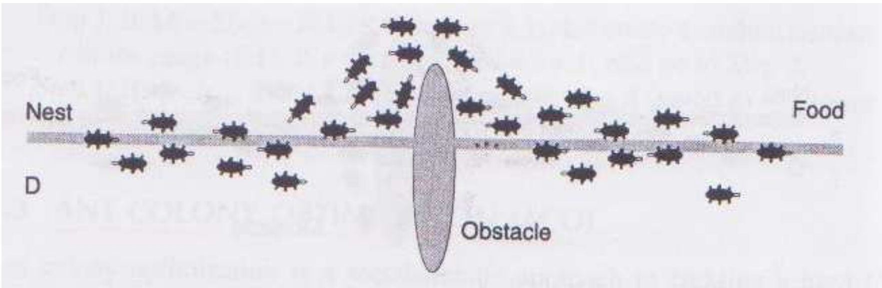


Figure 2.10 All ants have chosen the shorter path (Saravanan, 2006).

Although all ants move at approximately the same speed and deposit a pheromone trail at approximately the same rate, the fact that it takes longer to contour obstacles on their longer side than on their shorter side makes the pheromone trail accumulate more quickly on the shorter side. The ant's preference for higher pheromone trail levels makes this accumulation even quicker on the shorter path. A similar process used in a simulated world inhabited by artificial ants can solve a hard CO problem.

The artificial ants used to mimic the behavior of real ants in ACO differ in few respects:

These ants are not completely blind.

They have some memory.

They live in environment where time is discrete.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter includes the steps from the beginning till the end of process of developing this optimization turning parameters using ant colony optimization. This chapter also explains about how this method applied in this project and also justify the method is suitable for this project.

3.2 TYPES OF CUT

In turning process more common cutting process is used. Turning, in which the workpiece is rotated and cutting tool remove a layer of material as its moves to the left. This analysis is about straight turning process which is shown in Figure 3.1. Straight is one of common process used in turning to produce product. Material will used for this analysis is carbon steel. Diameter of raw material is 203mm and length of material is 152mm which is shown in Figure 3.2. Finish product for this analysis which is shown in Figure 3.3.

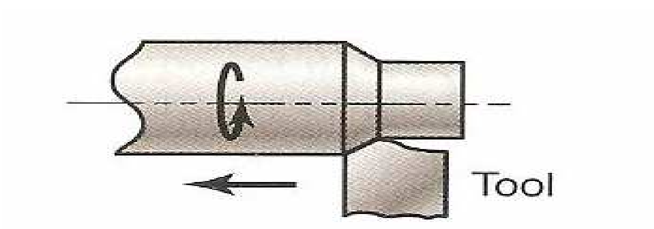


Figure 3.1 Straight turning.

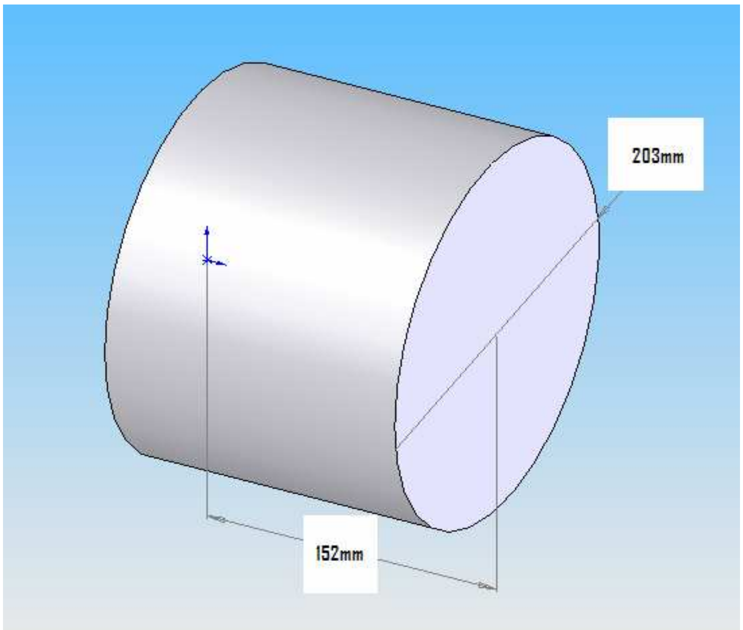


Figure 3.2 Raw material.

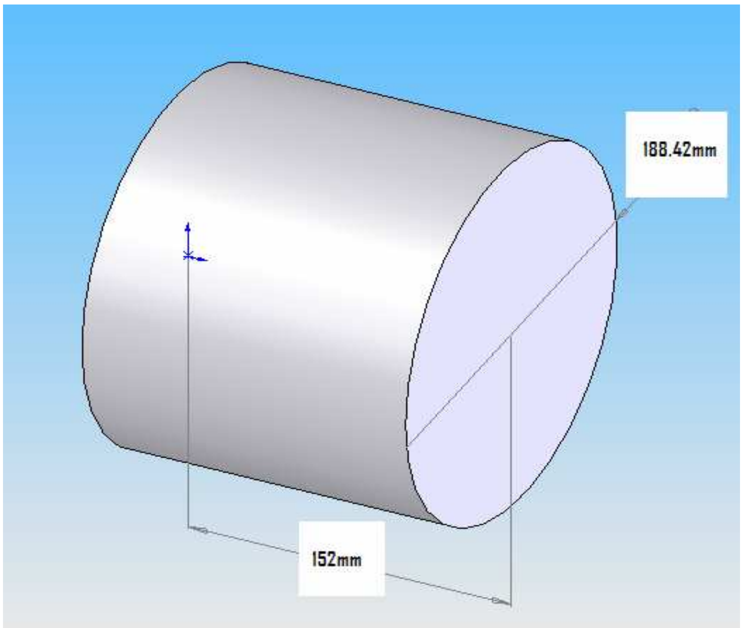


Figure 3.3 Finish product

3.3 ACO ALGORITHM

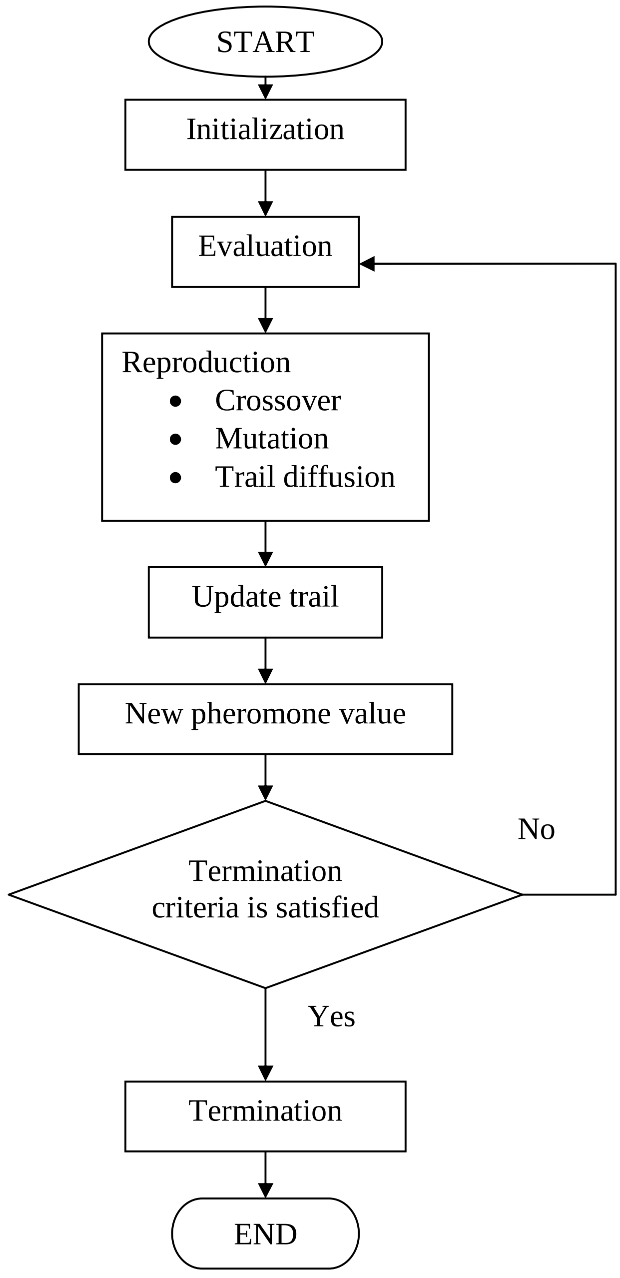


Figure 3.4 ACO algorithms.

3.4 IMPLEMENTATION OF ANT COLONY OPTIMIZATION

The flow chart shows about the steps used for this project. The steps to implementation of continuous ant colony optimization start with initialization, evaluation, reproduction, update trail, new pheromone value and lastly termination. Reproduction step is divided by three method, crossover, mutation and trail diffusion.

3.4.1 Initialization

In the first step, 20 random solutions are generated randomly within parameter bounds satisfying the constraint. The random solution is applied for depth of cut, cutting speed and feed rate. The random solution is shown in Table 3.1. After 20 random solutions are generated, the solution must convert or code into binary number Table 3.2. Combined all three parameter (binary number) to find the solution, this is shown in Figure 3.5.

Table 3.1: 20 random solutions.

Number	$doc(m)$	$f(m/rev)$	$V(m/\min)$
1	0.00258	0.000422	42
2	0.00352	0.000255	65
3	0.00201	0.000652	32
4	0.00303	0.000436	49
5	0.00452	0.000759	82
6	0.00489	0.000659	112
7	0.00366	0.000332	156
8	0.00335	0.000351	47
9	0.00223	0.000267	198
10	0.00345	0.000543	182
11	0.00476	0.000599	136
12	0.00264	0.000679	124
13	0.00231	0.000621	97
14	0.00355	0.000321	62
15	0.00433	0.000649	39
16	0.00226	0.000482	76
17	0.00332	0.000449	146
18	0.00487	0.000523	128
19	0.00264	0.000481	175
20	0.00238	0.000335	148

Table 3.2: Convert decimal to binary number.

Parameter	Range	Value	Binary number	Decimal number
$doc(m)$	Max	0.002	00000	0
		0.00335	01010	5
	Min	0.005	11110	20
$f(m/rev)$	Max	0.000254	00000	0
		0.000351	01100	8
	Min	0.000762	11110	20
$V(m/\min)$	Max	30	00000	0
		97	10011	10
	Min	200	11111	20

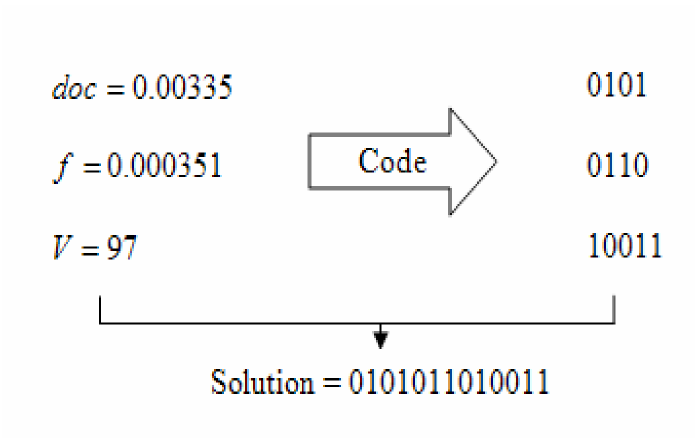


Figure 3.5 Three parameters combined to find the solution.

3.4.2 Evaluation

This section used to calculate the objective function. Decode the solution from initialization to decimal number and find the parameters value. The parameters value used to find objective function. The objective function is equal to fitness value, which is shown in Figure 3.6. Sort ascending a fitness values and divided in two section to find the superior and inferiors solution. 60% solution is superior and 40% solution is inferior, which is shown in Figure 3.7 After sort ascending, the objective function value which is shown in Table 3.3.

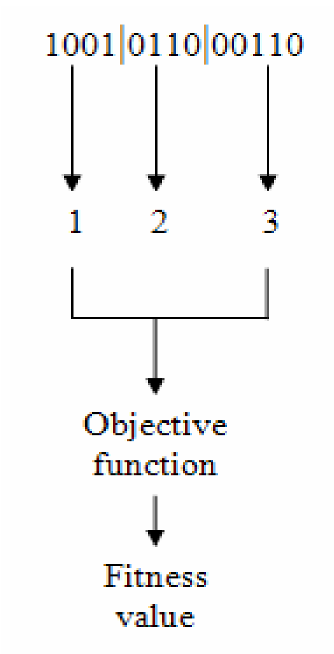


Figure 3.6 Evaluation.

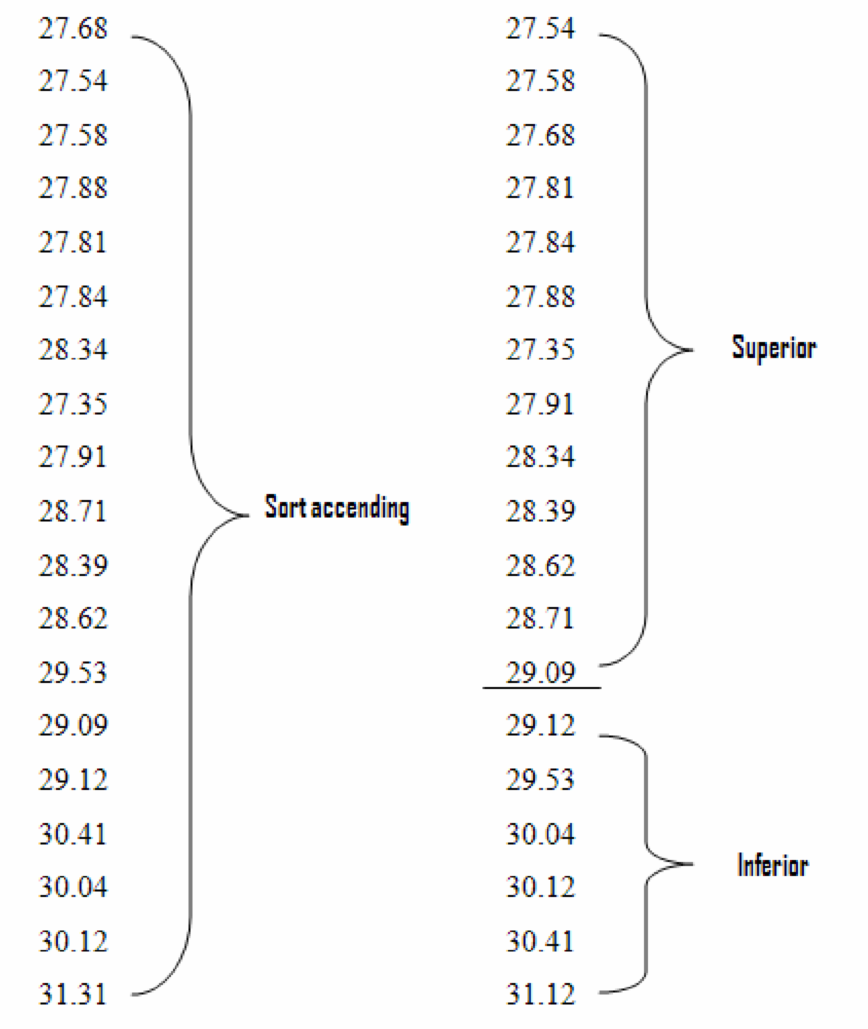


Figure 3.7 Superior and inferior.

Table 3.3: After sort ascending.

Region	Number	Objective function value (RM)
Superior	1	27.54
	2	27.58
	3	27.68
	4	27.81
	5	27.84
	6	27.88
	7	27.35
	8	27.91
	9	28.34
	10	28.39
	11	28.62
	12	28.71
Inferior	13	29.09
	14	29.12
	15	29.53
	16	30.04
	17	30.12
	18	30.41
	19	31.12
	20	31.31

3.4.2.1 Objective function

Both of the production cost and time are considered as objective functions. The production cost per component for a machining operation is comprised of the sum of the cost per component for tooling, machining, tool change time, handling time, and quick return time, which is given below:

$$C_u = C_o t_m + \left(\frac{t_m}{T}\right)\times\left(C_o t_{cs} + C_t\right) + C_o\left(t_h + t_R\right) + C_P + C_w \quad \text{(Eq 3.1)}$$

Where the cutting time per pass is

$$t_m = \frac{DL}{1000Vf} \quad \text{(Eq 3.2)}$$

Power cost is

$$C_P = 0.0373\times V^{0.91} f^{0.78} doc^{0.75} \times \frac{t_m}{60} \times RM\,0.20 \quad \text{(Eq 3.3)}$$

The total time required to machine a part is the sum of the times necessary for machining, tool changing, tool quick return, and workpiece handling.

$$T_u = t_m + t_{cs}\left(\frac{t_m}{T}\right) + t_R + t_h \quad \text{(Eq 3.4)}$$

Taylor's tools life equation is represented in terms of V, f, doc, and T:

$$Vf^{al} doc^{a2} T^{a3} = K \quad \text{(Eq 3.5)}$$

Where, a1, a2, a3 and K are the constants. This equation is valid over a region of speed and feed by which the tool life (T) is obtained.

3.4.2.2 Constraints

Maximum and minimum permissible feed rate, cutting speed, and depth of cut:

$$f_{\min} \leq f \leq f_{\max} \quad \text{(Eq 3.6)}$$

$$V_{\min} \leq V \leq V_{\max} \quad \text{(Eq 3.7)}$$

$$doc_{\min} \leq doc \leq doc_{\max} \quad \text{(Eq 3.8)}$$

Power limitation:

$$0.0373 \times V^{0.91} f^{0.78} doc^{0.75} \leq P_{\max} \quad \text{(Eq 3.9)}$$

The above constraints were taken from three independent sources and available in (Saravanan, 2006).

3.4.2.3 Data of problem

This section is shown about data of problem used in this project. This data are taken from experiment of turning plain carbon steel without coolant.

$$L = 203\text{mm}$$
$$D = 152\text{mm}$$
$$T = 35\,\text{min}$$
$$V_{\min} = 30\text{m/min},\ V_{\max} = 200\text{m/min}$$
$$f_{\min} = 0.254\text{mm/rev},\ f_{\max} = 0.762\text{mm/rev}$$
$$R_{a,\max}(r) = 12\text{m},\ R_{a,\max}(f) = 8\text{m}$$
$$P_{\max} = 5\text{kW}$$
$$F_{\max} = 900\text{N}$$
$$\theta_{\max} = 500\,^{o}\text{C}$$
$$doc_{\min}(r) = 2.0\text{mm},$$
$$doc_{\max}(r) = 5.0\text{mm}$$
$$a_1 = 0.29,\ a_2 = 0.35,\ a_3 = 0.25$$
$$K = 193.3$$
$$t_{cs} = 0.5\ \text{min/edge}$$
$$t_R = 0.13\text{min/pass}$$
$$t_h = 1.5\text{min/piece}$$
$$C_o = \text{RM}0.28\text{/min}$$
$$C_t = \text{RM}1.41\text{/edge}$$
$$C_w = \text{RM}1.65$$

The above data of problem were taken from three independent sources and available in (Saravanan, 2006).

3.4.3 Reproduction

This is third step apply for this project. This step consist of four part with is crossover, mutation, trail diffusion and local search.

3.4.3.1 Crossover

Crossover is divided in three sections. This step used to repair the inferior solution from 13 to 18. To apply crossover, firstly generate 1 random number from 1 to 12. Example solution is “7”, than code the solution to binary number. The solution is called “Parent 1”, which is shown in Figure 3.4. Choose “13” as an inferior solution, than code the solution to binary number. This solution is called “Parent 2”, which is shown in Figure 3.8. Second step in crossover is, generate another random number from 1 to 13 and code the solution into binary number. As an example, “10” is represented as the solution and modify the random number and fitness value of “10” to find “Child 1” and “Child 2”, which is shown in Figure 3.9. Lastly, evaluate the child to find the fitness value. To apply this step “Child 1” and “Child 2” must be decode or changed to decimal number.

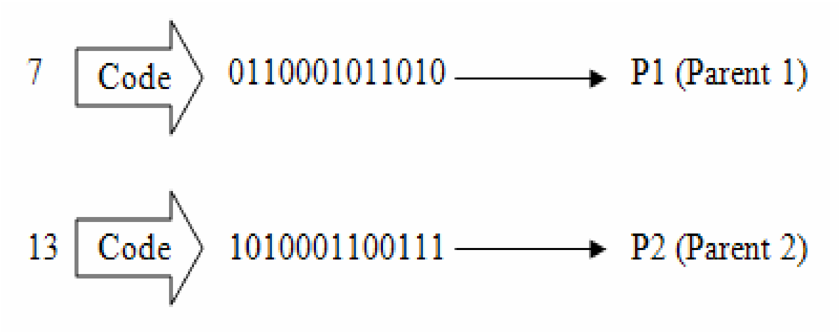


Figure 3.8 Parent 1 and Parent 2.

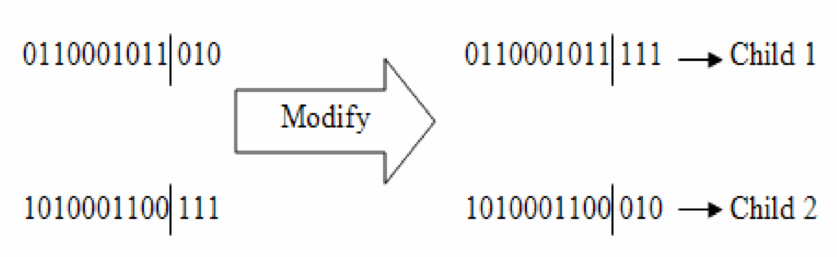


Figure 3.9 Child 1 and Child 2.

3.4.3.2 Mutation

This step is used to repair the part of the solution from 13 to 18. Firstly, level of mutation (m) is randomly generated from 0 to 1. As an example, 0.45 is the value of level of mutation. Than, generate a random number (r) from 0 to 1 an increasing 0.01. From this 0.40 is the value is applied for solution 13. Before apply the level of mutation and random value, the mutation rules must be considered. The mutation rules are $(r \geq m)$ and $(r < m)$. The random number (r) is considered when random number is bigger than equal of level of mutation (m). The (r) is not considered when random number is smaller than level of mutation (m).

Second step in mutation is, as an example 0.6 is apply for fitness value 16. Than generate 1 random number from 0 to 13, example value 6 is appearing. From behind of fitness value in binary number (Parent 1), swap the binary number to get Child 1. This second step which shown in Figure 3.10.

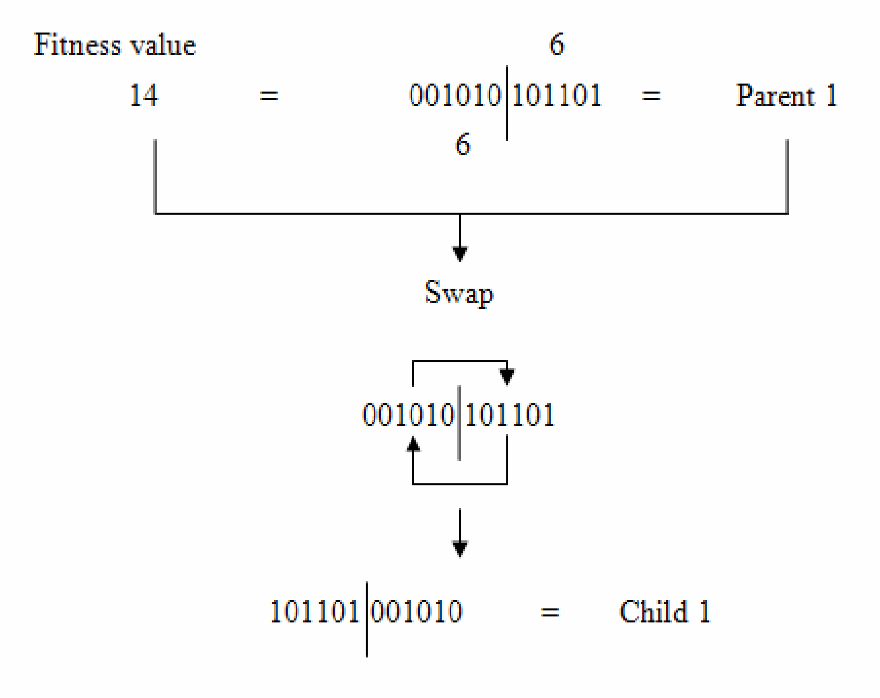


Figure 3.10 Second step in mutation.

3.4.3.3 Trail diffusion

Trail diffusion is the third step applied in reproduction. Firstly, generate 2 different numbers from 1 to 12. As an example, 8 and 11 are finding, and code to binary number. Than take the period from superior and decode to decimal. Then the value for X_{P1} and X_{P2} are finding. This step which shown in Figure 3.11.

Second step in trail diffusion is generated random number of alpha (α) from 0.0 to 1.0. Before accept the α value, the rules must be considered. First rule is, if α smaller than 0.5 $(\alpha < 0.5)$,

$$X_{child} = \alpha X_{P1} + (1-\alpha) X_{P2} \quad \text{(Eq 3.10)}$$

Calculate value of X_{child} and code to binary number. After calculated, replace the value for fitness solution. Second rule of α is, if α is bigger than $0.5\,(\alpha > 0.5)$,

$$X_{child} = X_{P2}\,. \qquad \text{(Eq 3.11)}$$

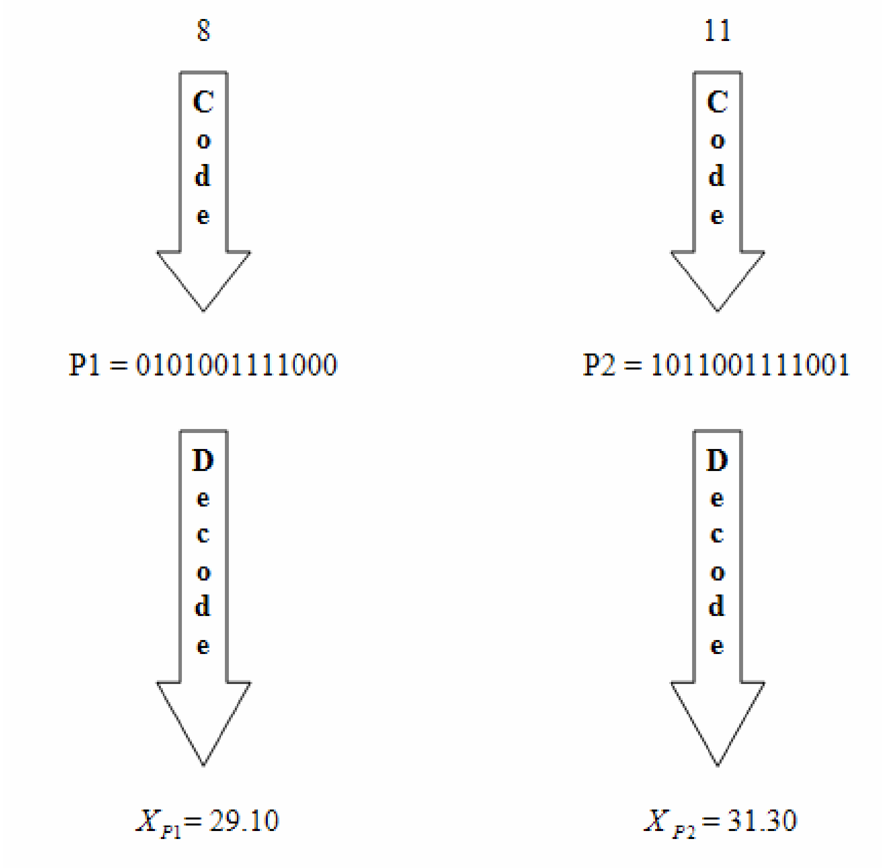


Figure 3.11 Value of X_{P1} and X_{P2}.

3.4.4 Update trail

Method used for this step is local search. Update trail is used to solve solution for superior region from 1 to 12. Pheromone value and age value for each value of superior must be set for before change value of superior region. As an example, pheromone value is equal to 1.0, $(ph = 1.0)$ and age value is equal to 10, $(age = 10)$. The value of ph and age is apply for all value of superior region, which is shown in Table 3.4.

Table 3.4: Update trail for superior solution.

Superior solution	ph	age
27.54	1.0	10
27.58	1.0	10
27.68	1.0	10
27.81	1.0	10
27.84	1.0	10
27.88	1.0	10
27.35	1.0	10
27.91	1.0	10
28.34	1.0	10
28.39	1.0	10
28.62	1.0	10
28.71	1.0	10

Secondly in update trail, calculate the pheromone average; Number of solution superior region is equal to 12.

$$(ph_{ave} = \sum ph / \text{Number of solution superior region}). \qquad \text{(Eq 3.12)}$$

Third step in update trail is limiting step, $\lim step$. Firstly state the constant values of limiting step $k1 = 0.1$ and $k2 = 0.01$. The rule for limiting step is $k1$ is bigger than $k2$, $(k1 > k2)$. Than apply the values in limiting step formula, to find the limiting step value.

$$\lim step = k1 - (age \times k2) \quad \text{(Eq 3.13)}$$

Lastly, find the value of X iteration new, (X_{inew}). Before that, firstly generate random, r from 0 to 1. Value of r is considered base on two rules, if r is bigger than 0.5 solution and

$$X_{inew} = X_{iold} + \lim step\,. \quad \text{(Eq 3.14)}$$

If r is smaller than 0.5,

$$X_{inew} = X_{iold} = \lim step\,. \quad \text{(Eq 3.15)}$$

All X or solution value must be decode before applied at this two formula. The last step which shown in Figure 3.12.

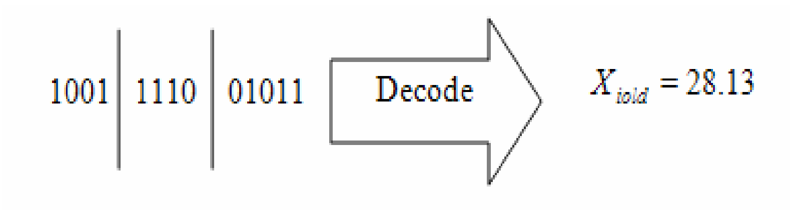


Figure 3.12 Find value of X_{iold}.

3.4.5 New pheromone value

To apply this method, fitness value for new solution $F(X_{inew})$ and fitness value for old solution $F(X_{iold})$ are considered. This two fitness values is used to find age_{new}. Value of age_{new} is find to base on two rules. If $F(X_{inew})$ is smaller than $F(X_{iold})$, the value of age new, $age_{new} = age + 1$. This step which shown in Figure 3.13.

$$F(X_{inew}) = 29.28 < F(X_{iold}) = 30.30$$

$$\downarrow$$

$$age_{new} = 10 + 1 = 11$$

$$\downarrow$$

$$ph_{new}$$

Figure 3.13 $F(X_{inew})$ is smaller than $F(X_{iold})$.

If $F(X_{inew})$ is bigger than $F(X_{iold})$, the value of age new, $age_{new} = age - 1$. This step which shown in Figure 3.14.

$$F(X_{inew}) = 29.28 > F(X_{iold}) = 30.30$$

$$\downarrow$$

$$age_{new} = 10 - 1 = 9$$

$$\downarrow$$

$$ph_{new}$$

Figure 3.14 $F(X_{inew})$ is bigger than $F(X_{iold})$.

The values of $F(X_{inew})$ and $F(X_{iold})$ is also used to find value for pheromone iteration new, ph_{inew}. Formula to find pheromone iteration new is

$$ph_{inew} = \frac{F(X_{inew}) - F(X_{iold})}{F(X_{iold})} + ph_{iold}. \qquad \text{(Eq 3.16)}$$

3.4.6 Termination

Termination is last step apply for ACO method. This step is used to set the number of iteration, example $iter = 100$ and also used to set required of fitness value, F.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 INTRODUCTION

This chapter will describe about simulation setup and assumption and also describe result from Initialization, Evaluation, Reproduction, Mutation, Trail Diffusion, Local Search, Update trail, New Pheromone value and lastly result about Termination process.

This chapter also describe about graph optimum production cost versus generation and graph best overall production cost versus generation. To make sure result from simulation is correct, this chapter also describe about verification and validation.

Before doing an analysis, fixed parameters and constraints from other experiment were use to identify the result. All of the parameters and constraints were taken from the experiment of turning plain carbon steel without coolant. After that step of ACO method is applied using Matlab software.

This thesis is create to identify the best value of feed rate, cutting speed, depth of cut and production cost per unit. This result is generating based on probability in ACO step from hundred iterations.

4.2 SIMULATION OBJECTIVES

The simulation objective must be considered in this simulation. This is because the right setups of machining parameters are required.

The ACO used for the work reported here, was tested along with the data provided by R. Saravanan (2006). The steps to implement of ACO start with initialization, evaluation, reproduction, update trail, new pheromone value and lastly termination.

Straight turning is one of the processes to produce product. Many parameters involve in this process. This thesis will identify the best parameters value for depth of cut, cutting speed and feed rate by using ACO technique

Production cost is an important factor to consider before start the production. It is important to develop this technique for the industry to gain profits.

In CNC turning process, a few factor such as prices of material and parameters setup will effect to total production cost. Therefore to gain more profits, the right setups of machining parameters are required.

4.3 SIMULATION SETUP AN ASSUMPTION

In single-pass optimization problems, specification of the personal computer is considered because it will influence the simulation result. To make simulation run well the initial setup of computer also important. Other reason is to know the time taken for hundred iterations. This simulation using a 512 MB of Ram, 30 GB hard disc and Intel Pentium 4 (2.40 GHz) as a processor.

Simulation was performed by the idleness of several assumptions in accordance with ACO procedure. Any assumption for constant parameter and objective function was taken from “Manufacturing Optimization through Intelligent Technique” book as a reference. All of the data used for this work was tested along with the data provided by R. Saravanan (2006). The data also was taken from India because the experiment to get the data was occurred in that country. Therefore, all the result is valid for this analysis.

However, when we apply to the local problem, few changes must be considered such as coat of material and currency. The original data were directly being applied to this problem to compare the final result that acquired through ACO and the result from reference.

4.4 RESULT

The result that acquired from ACO are according to it steps. By using Matlab software for simulation or analysis and considered objective functions also constraints and constant parameters the optimal value for depth of cut, cutting speed, feed rate and production cost was successfully generate for this thesis.

4.4.1 Initialization result

Initialization step is used to create 20 solutions in binary number. Random solutions are generated for depth of cut, cutting speed and feed rate. Lastly initialization procedure must obtained 20 solutions for new chrome in binary number, which is shown in Table 4.1.

Table 4.1: Initialization result

Number	Chrome	Number	Chrome
1	110111001111110	11	000110100010101
2	100000110000101	12	100000010011011
3	111100010011000	13	010010101111101
4	110111110111110	14	100000011000000
5	011111000110111	15	101000110111101
6	011110000100011	16	100000100001000
7	001110011111001	17	011001011101110
8	110111111001011	18	100011111011001
9	011000101101100	19	001100011101010
10	000110011010110	20	100011010000001

4.4.2 Evaluation result

Evaluation is the second procedure in ACO. 20 solution from initialization is taken to be use in this procedure. This procedure must used objective function to get the 20 fitness value. The solution also call the chrome inform in binary number, which is shown in Table 4.2.

Table 4.2: Evaluation result

Number	Chrome	Number	Chrome
1	110011111011001	11	001000010011000
2	101010101110111	12	110100001010101
3	100001100101101	13	100010010010010
4	001111010101101	14	101111100100011
5	100010110010000	15	101100100001010
6	000111010101100	16	001100011100111
7	101111100101001	17	111001000100000
8	001011100001010	18	001010011100101
9	010011101001000	19	010100001100111
10	101101100000111	20	100010101000000

4.4.3 Reproduction result

Reproduction consists of three procedure crossover, mutation and trail diffusion

4.4.3.1 Crossover result

To build crossover, must consider the formula and limitation to identify child and parent need to be considered. This procedure used to generate 20 new solutions from evaluation chrome. This solution also encoded in binary number, which is shown in Table 4.3.

Table 4.3: Crossover result

Number	Chrome	Number	Chrome
1	101001010011100	11	000010111001111
2	110011000011101	12	010010110001101
3	000001010011101	13	011000111001011
4	111010111111100	14	000110110010010
5	010100110011101	15	100000111010000
6	101101110001111	16	110110011111110
7	100010100010110	17	011001011000111
8	011100111101111	18	000001010011100
9	001000011010111	19	000100101100001
10	000011001001101	20	000010000100100

4.4.3.2 Mutation result

Mutation procedure used to find new chrome from crossover. Mutation procedures consist of many conditions such as probability of mutation and level of mutation. Chrome from this procedure which is shown in Table 4.4.

Table 4.4: Mutation result

Number	Chrome	Number	Chrome
1	101111111011000	11	000000010110100
2	110001101010111	12	111100111100110
3	011111001110110	13	100010010010011
4	101101001110101	14	001110001001001
5	000101000111000	15	001001110001001
6	001011001010011	16	010100110111010
7	110001011101110	17	100111000100100
8	010011000010000	18	101111100100110
9	110111100100110	19	101110000101011
10	111010001010101	20	001010111100000

4.4.3.3 Trail diffusion result

After mutation procedure, trail diffusion procedure is applied. This procedure applied to repair the child and change the parent using a condition. Then other 20 chrome is generated in form of binary number, which is shown in Table 4.5.

Table 4.5 Trail diffusion result

Number	Chrome	Number	Chrome
1	100011011011110	11	010010110001111
2	111101101110111	12	001010000011100
3	110011010010011	13	001000011111011
4	000110110011001	14	010110101111011
5	111000011011101	15	110000011110110
6	011010100011011	16	111101100100001
7	000001001110001	17	011110010110011
8	100101100101100	18	100001111101100
9	000001100001110	19	100111010101000
10	111010010011011	20	000101010011100

4.4.4 Local search result

Local search procedure is applied after trail diffusion. It is also used to find the new chrome for changing the chrome identities with trail diffusion procedure. Chrome for local search which is shown in Table 4.6.

Table 4.6: Local search result

Number	Chrome	Number	Chrome
1	100101101011010	11	111000100101101
2	100011001011110	12	101110001110010
3	110000110111101	13	100011101111000
4	010011100110101	14	111011001000011
5	111011000010011	15	111110110010000
6	110101000110001	16	111111100101100
7	111010110010100	17	001000101010100
8	000101101001100	18	111000100101101
9	000010100110100	19	101110100101011
10	110000111101011	20	101011110011001

4.4.5 Update trail

Update trail procedure consist the new pheromone value as a procedure to identify the new solutions.

4.4.5.1 New Pheromone value

New pheromone value is procedure to find Ph average. The initial value for pheromone for this procedure is 1.000.

4.4.6 Termination result

This is last procedure applied in ACO method. From this procedure the best solution for production cost per unit and also optimum parameters were identified, it is shown in Table 4.7. The final results from this procedure were represented in graphs. First graph is the optimum production cost versus generation and second graph is the best overall production cost versus generation. The discussion for the graphs will elaborate in the next sub-topic.

Table 4.7: Termination result

	$doc(mm)$	$V\left(\frac{mm}{\min}\right)$	$f\left(\frac{mm}{rev}\right)$	Production cost (RM/piece)
Best result (roughing cut)	3.6452	1.94520	0.7620	2.1440

4.5 RESULTS AND DISCUSSION

The Ant Colony Optimization used for this thesis was tested along with the dat provided by R. Saravanan. This analysis was generated the optimum parameter for CNC turning machine. The optimum parameter used to identify minimum production cost per unit. To make sure decision produce to be exact and optimum, 100 iteration were used. These analyses also generate two graphs. First graph is show the optimum production cost versus generation which is shown in Figure 4.1. The second graph is show the best overall production cost versus generation which is presented in Figure 4.2.

4.5.1 Optimum production cost versus generation graph

In this single pass optimization problem, the first iteration graph is generated for rough cut. This graph shows the optimum cost versus generation which is shown in Figure 4.1. From that graph, it shows different decisions in different generations. All of the optimum production cost for this graph is shown in Table 4.8. The minimum value for production cost that was obtained from graph is RM 2.1440 and the maximum value is RM2.1676. The result was obtained is acceptable because the value is within the range that were obtained from reference. The result also not far differences from sample calculation.

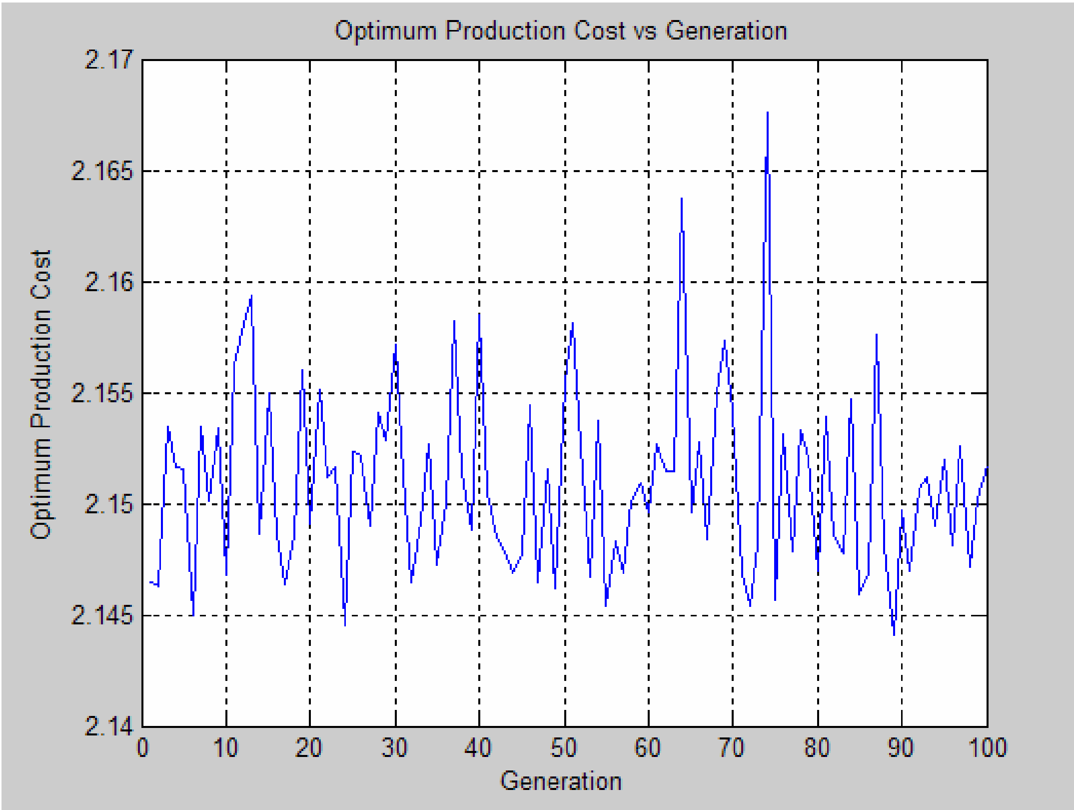


Figure 4.1 Optimum production cost versus generation graph.

Table 4.8: Optimum production cost for graph optimum production cost versus generation

Iter	Cost (RM)	Iter	Cost (RM)	Iter	Cost (RM)	Iter	Cost (RM)	Iter	Cost (RM)
1	2.1464	21	2.1551	41	2.1504	61	2.1527	81	2.1539
2	2.1463	22	2.1512	42	2.1484	62	2.1515	82	2.1485
3	2.1534	23	2.1516	43	2.1478	63	2.1515	83	2.1478
4	2.1516	24	2.1445	44	2.1469	64	2.1637	84	2.1547
5	2.1515	25	2.1523	45	2.1478	65	2.1496	85	2.1459
6	2.1449	26	2.1522	46	2.1544	66	2.1528	86	2.1469
7	2.1534	27	2.1490	47	2.1464	67	2.1484	87	2.1576
8	2.1502	28	2.1541	48	2.1515	68	2.1549	88	2.1481
9	2.1534	29	2.1529	49	2.1461	69	2.1573	89	2.1440
10	2.1468	30	2.1572	50	2.1555	70	2.1542	90	2.1497
11	2.1562	31	2.1513	51	2.1581	71	2.1469	91	2.1470
12	2.1581	32	2.1465	52	2.1526	72	2.1454	92	2.1507
13	2.1593	33	2.1492	53	2.1467	73	2.1481	93	2.1512
14	2.1486	34	2.1527	54	2.1537	74	2.1676	94	2.1490
15	2.1550	35	2.1472	55	2.1454	75	2.1457	95	2.1520
16	2.1486	36	2.1500	56	2.1483	76	2.1531	96	2.1481
17	2.1463	37	2.1582	57	2.1469	77	2.1478	97	2.1526
18	2.1486	38	2.1514	58	2.1501	78	2.1533	98	2.1472
19	2.1560	39	2.1488	59	2.1510	79	2.1520	99	2.1502
20	2.1491	40	2.1584	60	2.1496	80	2.1470	100	2.1517

4.5.2 Best overall production cost versus generation graph

The graph in Figure 4.2 show about best overall production cost versus generation. This graph was obtained when ACO was run for 100 iterations. The best production cost is selected to produce this graph for first iteration until 100 iterations. The best result for every iteration which is shown in Table 4.9. The graph also shows more iteration required to produce best solution for production cost. The maximum production cost is RM 2.1464 and the minimum production cost is RM 2.1440. The minimum value for this analysis was obtained from iteration 89 until 100. That mean here RM 2.1440 is the best result for production cost in this analysis. From the result it is clear that the proposed ACO result significantly within that range from reference and almost equal within sample calculation.

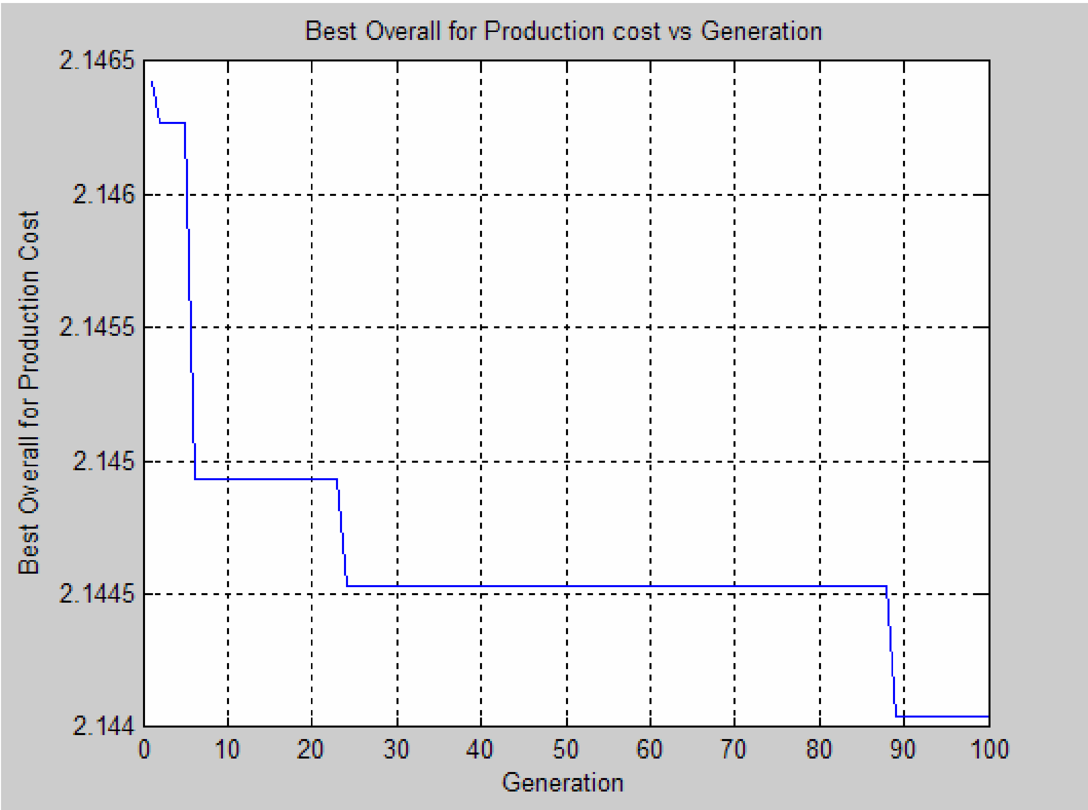


Figure 4.2 Best overall production cost versus generation graph.

Table 4.9: The best result for every iteration in graph best overall production cost versus generation.

Iter	Cost (RM)	Iter	Cost (RM)	Iter	Cost (RM)	Iter	Cost (RM)	Iter	Cost (RM)
1	2.1464	21	2.1449	41	2.1445	61	2.1445	81	2.1445
2	2.1463	22	2.1449	42	2.1445	62	2.1445	82	2.1445
3	2.1463	23	2.1449	43	2.1445	63	2.1445	83	2.1445
4	2.1463	24	2.1445	44	2.1445	64	2.1445	84	2.1445
5	2.1463	25	2.1445	45	2.1445	65	2.1445	85	2.1445
6	2.1449	26	2.1445	46	2.1445	66	2.1445	86	2.1445
7	2.1449	27	2.1445	47	2.1445	67	2.1445	87	2.1445
8	2.1449	28	2.1445	48	2.1445	68	2.1445	88	2.1445
9	2.1449	29	2.1445	49	2.1445	69	2.1445	89	2.1440
10	2.1449	30	2.1445	50	2.1445	70	2.1445	90	2.1440
11	2.1449	31	2.1445	51	2.1445	71	2.1445	91	2.1440
12	2.1449	32	2.1445	52	2.1445	72	2.1445	92	2.1440
13	2.1449	33	2.1445	53	2.1445	73	2.1445	93	2.1440
14	2.1449	34	2.1445	54	2.1445	74	2.1445	94	2.1440
15	2.1449	35	2.1445	55	2.1445	75	2.1445	95	2.1440
16	2.1449	36	2.1445	56	2.1445	76	2.1445	96	2.1440
17	2.1449	37	2.1445	57	2.1445	77	2.1445	97	2.1440
18	2.1449	38	2.1445	58	2.1445	78	2.1445	98	2.1440
19	2.1449	39	2.1445	59	2.1445	79	2.1445	99	2.1440
20	2.1449	40	2.1445	60	2.1445	80	2.1445	100	2.1440

Three parameters were selected to optimize turning parameter for minimized production cost per unit. The parameters selected are depth of cut, cutting speed and feed rate. Result optimize turning parameter for minimize production cost per unit using ACO which is shown in Table 4.10 for depth of cut, cutting speed and feed rate. The result also shown, the probability taken by ACO to selected the different value of parameter with that range. The result was obtained by using ACO for depth of cut, cutting speed, and feed rate within range that prescribed.

Table 4.10: 100 Iteration result for depth of cut, cutting speed, feed rate and cost.

Iteration	$doc(mm)$	$V\left(\frac{mm}{\min}\right)$	$f\left(\frac{mm}{rev}\right)$	Cost (RM)
1	2.0968	106770	0.5162	2.1464
2	2.0968	112260	0.5817	2.1463
3	2.0968	117740	0.5981	2.1463
4	2.0968	117740	0.5981	2.1463
5	2.1935	123230	0.6145	2.1463
6	2.1935	128710	0.6145	2.1449
7	2.1935	128710	0.6145	2.1449
8	2.2903	134190	0.6309	2.1449
9	2.3871	139680	0.6309	2.1449
10	2.3871	139680	0.6309	2.1449
11	2.3871	139680	0.6309	2.1449
12	2.3871	139680	0.6309	2.1449
13	2.3871	145160	0.6309	2.1449
14	2.4839	145160	0.6309	2.1449
15	2.4839	145160	0.6309	2.1449
16	2.5806	150650	0.6473	2.1449
17	2.5806	150650	0.6473	2.1449
18	2.5806	150650	0.6473	2.1449
19	2.5806	150650	0.6637	2.1449
20	2.6774	150650	0.6637	2.1449
21	2.6774	150650	0.6637	2.1449
22	2.6774	150650	0.6637	2.1449
23	2.6774	150650	0.6637	2.1449
24	2.6774	150650	0.6637	2.1445
25	2.6774	150650	0.6637	2.1445
26	2.7742	156130	0.6637	2.1445
27	2.7742	156130	0.6637	2.1445

28	2.8710	161610	0.6801	2.1445
29	2.8710	161610	0.6801	2.1445
30	2.9677	161610	0.6801	2.1445
31	2.9677	161610	0.6801	2.1445
32	3.0645	161610	0.6801	2.1445
33	3.0645	161610	0.6801	2.1445
34	3.0645	167100	0.6965	2.1445
35	3.1613	167100	0.6965	2.1445
36	3.1613	172580	0.6965	2.1445
37	3.1613	172580	0.6965	2.1445
38	3.2581	172580	0.6965	2.1445
39	3.2581	172580	0.6965	2.1445
40	3.2581	172580	0.6965	2.1445
41	3.2581	172580	0.6965	2.1445
42	3.2581	172580	0.6965	2.1445
43	3.3548	172580	0.6965	2.1445
44	3.3548	172580	0.6965	2.1445
45	3.3548	172580	0.7128	2.1445
46	3.3548	172580	0.7128	2.1445
47	3.4516	172580	0.7128	2.1445
48	3.5484	172580	0.7128	2.1445
49	3.5484	178060	0.7128	2.1445
50	3.5484	178060	0.7128	2.1445
51	3.6452	178060	0.7128	2.1445
52	3.6452	178060	0.7128	2.1445
53	3.6452	178060	0.7128	2.1445
54	3.6452	178060	0.7128	2.1445
55	3.6452	178060	0.7128	2.1445
56	3.7419	178060	0.7128	2.1445
57	3.7419	178060	0.7128	2.1445
58	3.7419	178060	0.7128	2.1445
59	3.8387	178060	0.7128	2.1445
60	3.8387	178060	0.7292	2.1445
61	3.9355	178060	0.7292	2.1445
62	3.9355	183550	0.7292	2.1445
63	3.9355	183550	0.7292	2.1445
64	3.9355	183550	0.7292	2.1445
65	3.9355	183550	0.7292	2.1445
66	3.9355	183550	0.7292	2.1445
67	3.9355	183550	0.7292	2.1445
68	3.9355	183550	0.7292	2.1445
69	3.9355	183550	0.7292	2.1445
70	4.0323	183550	0.7292	2.1445
71	4.0323	183550	0.7292	2.1445

72	4.0323	189030	0.7292	2.1445
73	4.0323	189030	0.7456	2.1445
74	4.0323	189030	0.7456	2.1445
75	4.0323	189030	0.7456	2.1445
76	4.1290	189030	0.7456	2.1445
77	4.2258	189030	0.7456	2.1445
78	4.2258	189030	0.7456	2.1445
79	4.2258	189030	0.7456	2.1445
80	4.3226	189030	0.7456	2.1445
81	4.3226	189030	0.7456	2.1445
82	4.3226	194520	0.7456	2.1445
83	4.3226	194520	0.7456	2.1445
84	4.4194	194520	0.7456	2.1445
85	4.5161	194520	0.7456	2.1445
86	4.5161	194520	0.7456	2.1445
87	4.5161	194520	0.7620	2.1445
88	4.5161	194520	0.7620	2.1445
89	4.6129	194520	0.7620	2.1440
90	4.6129	194520	0.7620	2.1440
91	4.6129	194520	0.7620	2.1440
92	4.6129	194520	0.7620	2.1440
93	4.6129	194520	0.7620	2.1440
94	4.7097	194520	0.7620	2.1440
95	4.8065	194520	0.7620	2.1440
96	4.8065	194520	0.7620	2.1440
97	4.9032	194520	0.7620	2.1440
98	4.9032	194520	0.7620	2.1440
99	4.9032	194520	0.7620	2.1440
100	4.9032	200000	0.7620	2.1440

Therefore, the result clearly show that the proposed ACO approach is significant when compare to the result from R.Saravanan (2006) and simple calculation. The graph also shows the best result for production cost per unit is RM 2.1440. After that, the optimum parameters also obtained. The best parameters are 3.6452mm for depth of cut, 194520 mm/min for cutting speed and 0.7620mm/rev for feed rate. Stop watch is used to know about time taken for this analysis. The best result for parameters and production cost per unit which is shown in Table 4.13.

Table 4.11: The best result for this analysis

	$doc(mm)$	$V\left(\frac{mm}{\min}\right)$	$f\left(\frac{mm}{rev}\right)$	Production cost (RM/piece)
Best result (roughing cut)	3.6452	194520	0.7620	2.1440

Finally, the ACO method described in this thesis has proven a useful tool for improving the production cost. ACO also, automatically allocate the optimal parameters to the turning machine. This algorithm also useful tool to optimize turning parameters for minimized production cost per unit.

4.6 VERIFICATION

This part will show of simple calculation. Simple calculation is used to make sure the result obtained from the program is accurate. Constant value for parameters must be used to make this sample calculation. Median value for parameters used in this calculation. Value for depth of cut is 3.5mm, cutting speed is 11500mm/rev and feed rate is 0.508mm/min. Simple calculation are used objective function that was taken from “Manufacturing Optimization through Intelligent Technique” book.

4.6.1 Simple calculation

First calculation is about cutting time per pass.

$$t_m = \frac{DL}{1000Vf} \qquad \text{(Eq 4.1)}$$

$$t_m = \frac{152 \times 203}{1000 \times 194520 \times 0.762}$$

$$t_m = 2.0817 \times 10^{-4} \text{ min}$$

Second calculation is about Taylor’s tool life. Taylor’s tools life equation is represented in terms of V, f, doc, and T.

$$Vf^{al} doc^{a2} T^{a3} = K \qquad \text{(Eq 4.2)}$$

$$T = a3\sqrt{\frac{K}{Vf^{a1} doc^{a2}}}$$

$$T = 0.25\sqrt{\frac{193.3}{194520 \times 0.762^{0.29} \times 3.6452^{0.35}}}$$

$$T = 6.5372 \times 10^{-3} \text{ min}$$

Third calculation is about Power limitation. Power limitation result is used to calculate Power cost.

$$P_{\max} = 0.0373 \times V^{0.91} f^{0.78} doc^{0.75} \qquad \text{(Eq 4.3)}$$

$$P_{\max} = 0.0373 \times 194520^{0.91} 0.762^{0.78} 3.6452^{0.75}$$

$$P_{\max} = 5174.6467 kW$$

Fourth calculation is about Power cost. Before calculate Power cost, the value for Power limitation must be obtained.

$$C_P = P_{\max} \times \frac{t_m}{60} \times RM\,0.20 \qquad \text{(Eq 4.4)}$$

$$C_P = 5174.6467 \times \frac{2.0817 \times 10^{-4}}{60} \times RM\,0.20$$

$$C_P = RM\,3.5907 \times 10^{-3}$$

Finally, take the value from Cutting time per pass, Taylor’s tool life equation and Power cost put inside Production cost per unit equation to the suitable value for comparison. This is the objective of this thesis. When identify this value using calculation, this value must use for comparison to make sure the analysis using software is verify.

$$C_u = C_o t_m + \left(\frac{t_m}{T}\right) \times \left(C_o t_{cs} + C_t\right) + C_o\left(t_h + t_R\right) + C_P + C_w \qquad \text{(Eq 4.5)}$$

$$C_u = \left(0.28 \times 2.0817 \times 10^{-4}\right) + \left(\frac{2.0817 \times 10^{-4}}{6.5372 \times 10^{-3}}\right) \times \left(\left(0.28 \times 0.5\right) + 1.41\right)$$
$$+ \left(0.28\left(1.5 + 0.13\right)\right) + \left(3.5907 \times 10^{-3}\right) + 1.65$$

$$C_u = RM\, 2.1591$$

4.7 VALIDATION

Validation greatly needed to know whether decision obtainable is suitable. Validation also used to compare the answer from the previous research. Data to compare acquired from the Table 6.1 in the book “Manufacturing Optimization through Intelligent Technique” as a reference. Reference used for this analysis is Saravanan's outcome of the research with use method Nelder-Mead Simplex which is shown in Table 4.14.

Table 4.14: Optimization of single pass using Nelder-Mead Simplex Method (rough cut) (Saravanan, 2006).

No	Cost (RM)	No	Cost (RM)	No	Cost (RM)
1	2.0072	11	2.0917	21	2.2066
2	2.0099	12	2.1000	22	2.2190
3	2.0244	13	2.1294	23	2.2263
4	2.0305	14	2.1272	24	2.2449
5	2.0340	15	2.1594	25	2.2600
6	2.0369	16	2.1971	26	2.2714
7	2.0397	17	2.1915	27	2.2850
8	1.9988	18	2.2034	28	2.2984
9	2.0750	19	2.2056	29	2.3072
10	2.0713	20	2.2246	30	2.3219

Form this Table 4.14 the smallest cost is RM 1.9988 and the highest cost is RM 2.3219. Therefore, result of the analyses must be inside this range so can determine that the analysis result obtainable to be exact and right.

CHAPTER 5

CONCLUSION

5.1 INTRODUCTION

This sub-topic will conclude the result was obtained from simulation and also give the recommendation for this thesis. In this chapter, every suggestion that have been made will be elaborate in detail.

In previous chapter, analysis using different parameters for CNC machine turning obtained value for production cost per unit. From the result was successfully told this analysis is successfully being optimized.

5.2 CONCLUSION

In this report, a cutting optimization algorithm for single-pass turning for roughing cut operations has been presented and an ACO has been applied to solve the machining optimization problem for straight turning. The ACO applied for this analysis was successfully being developed using Matlab software and was successfully being optimize depth of cut, cutting speed, and feed rate. The algorithm also obtained the result to minimized production cost per unit for CNC turning machine.

The results of the proposed approach are compared with results of Nelder-Mead Simplex Method (rough cut) and simple calculation. The ACO algorithm can obtain near optimal solution. The effectiveness of the ACO algorithm has been proved through this analysis.

The results was proposed in this report, suitable used for making a product and using types of cutting same as elaborate in methodology. The results obtained for this analysis also suitable for carbon steel material.

5.3 RECOMMENDATION

The effectiveness of the ACO algorithm to optimize turning parameter for minimized production cost per unit has been proved through this report. From the analysis three parameters has been consider to optimize. It's used to minimize production cost per unit for CNC turning machine.

For future planning, the ACO algorithm can be easily modified to optimize this turning operation under various economic criteria such as identify the production rate of product.

Therefore, identify the best result for analysis is important. This algorithm also can be easily modified to consider various parameters such as surface roughness.

Usually surface roughness is considered at finishing operation. Multi pass operation consideration is also recommendation for future planning using this intelligent technique. Other recommendations are making an experiment and use the data to compare with ACO method.

Lastly, ACO is one of intelligent technique is useful for any ideas. That mean, this algorithm can also be extended to other to other machining problem, such as milling operation and treading operations.

REFERENCES

1. Kalpakjian, S. and Schmid, S. 2006. *Manufacturing Engineering and Technology*, Singapore: Prentice Hall.
2. Lee, Y. B. and Tarng, S. Y. 1999. Cutting-Parameter Selection for Maximizing Production Rate or Minimizing Production Cost in Multistage Turning Operations. *Journal of Materials Processing Technology.* **105** (2000): 61-66.
3. Min Kong. Peng Tian and Yucheng Kao. 2007. A New Ant Colony Optimization Algorithm for the Multidimensional Knapsack Problem. *Computer & Operation Research.* **35** (2008): 2672-2683.
4. Nian, Y. C. Yang, H. W. and Tarng, S. Y. 1998. Optimization of Turning Operations with Multiple Perfomances Characteristics. *Journal of Materials Processing Technology*. **95** (1999) 90-96.
5. Saravanan, R. 2006. *Manufacturing Optimization Through Intelligent Technique*, USA: Taylor & Francis.
6. Vijayakumar, K. Prabhaharan, G. Asokan, P. and Saravanan, R. 2003. Optimization of Multi-pass turning Operation Using Ant Colony system. *International Journal of Machines Tools & Manufacture.* **43** (2003): 1633-1639.
7. Wang, J. Kuriyagawa, T. Wei, P. X. and Gao, M. D. 2002. Optimization of Cutting Conditions for Single-pass turning Operations Using a Deteministic Approach. *International Journal of Machines Tools & Manufacture.* **42** (2002): 1023-1033.
8. Wei Gao. 2007. Study on Immunized Ant Colony Optimization. *Third International Conference on Natural Computation (ICNC 2007).***55**: 2875-7695.
9. Yang, H. W. and Tarng, S. Y. 1997. Design Optimization of Cutting Prameters for Turning Operations Based on the Taguchi Method. *Journal of Materials Processing Technology*. **84** (11998): 122-129.
10. Yi-Chi Wang. 2007. A note on ‘optimization of multi-pass turning operations using ant colony system’. *International Journal of Machines Tools & Manufacture*. **47** (2007): 2057-2059.

APPENDIX A

SIMULATION PROGRAMMING

```
%Implementation of ant colony optimization
   %Initialization
disp('Step 1: INITIALIZATION');

%%---------------------------------------------------------------------

%20 random solution are generated within bounds satisfying the constraint.
   %The random solution is applied for:
      %doc(r)=depth of cut for roughing (parameter 1)
      %V=cutting speed (parameter 2)
      %f=feed rate (parameter 3)
doc = 3*rand(1,20)+2;
V = 170000*rand(1,20)+30000;
f = 0.508*rand(1,20)+0.254;

   disp('The random solution for depth of cut is:');
      doc
   disp('The random solution for cutting speed is:');
      V
   disp('The random solution for feed rate is:');
      f

%%---------------------------------------------------------------------

%Movement of solution
for i = 1:1:20;
      doc(1,i);
      V(1,i);
      f(1,i);

%Convert solution to decimal number
   %Formula for convert (interpolation)
      %dec=decimal
decdoc = 31*(doc-2)/3;      %Intrepolation
decV = 31*(V-30000)/170000;      %Intrepolation
decf = 31*(f-0.254)/0.508;  %Intrepolation
end

   disp('The decimal number for depth of cut:');
      decdoc
```

```
   disp('The decimal number for cutting speed:');
      decV
   disp('The decimal number for feed rate:');
      decf

%%-----------------------------------------------------------------------

%Convert decimal number to binary number
   %bin=binary
bindoc = dec2bin(decdoc);
binV = dec2bin(decV);
binf = dec2bin(decf);

   disp('The binary number for depth of cut is:');
      bindoc
   disp('The binary number for cutting speed is:');
      binV
   disp('The binary number for feed rate is:');
      binf

%%-----------------------------------------------------------------------

%combined parameter to get solution
for nchrom = 1:1:20;
      num = [bindoc(nchrom,:) binV(nchrom,:) binf(nchrom,:)];
         chrom(nchrom,:) = [num];
end

   disp('The 20 solution is:');
      chrom

%%-----------------------------------------------------------------------

%Evaluation
   %Load file from Initialization
disp('Initialization')
   Initialization

disp('Step 2: EVALUATION');

%%-----------------------------------------------------------------------

%Data of problem using in this project
   %Three independant source available in Saravanan 2006
disp('Data of problem is:')
   L = 203        %Length of workpiece (mm)
```

```
   D = 152        %Diameter of workpiece (mm)
   %T = 35         %Tool life (min)
   %Rmax = 12000   %Maximum surface roughness of rough (m)
   %Pmax = 5000    %Power of the machine (kW)
   %Fmax = 900     %Cutting force (N)
   %Omax = 500     %Temperature of toll workpiece interface (oC)
   a1 = 0.29      %Constatnts used in tool life equation
   a2 = 0.35      %Constatnts used in tool life equation
   a3 = 0.25      %Constatnts used in tool life equation
   K = 193.3      %Constatnts used in tool life equation
   tcs = 0.5      %Tool change time (min/edge)
   tR = 0.13      %Quick return time(min/pass)
   th = 1.5       %Tool production time (min)
   Co = 0.28      %Operating cost (RM/piece)
   Ct = 1.41      %Tool cost per cutting edge (RM/piece)
   Cw = 1.65      %Workpiece cost (RM)

%%---------------------------------------------------------------------

%Movement of solution
for i = 1:1:20;
      doc(1,i);
      V(1,i);
      f(1,i);

%Formula to find cutting parameter per pass
   %Using 1 and j for random sata
   for j = 1:1:20;
      tm = (D*L)/(1000*V(1,j)*f(1,j));
         CuttingTime(j,:) = [tm];
   end
end

   disp('The cutting time per pass is:');
      CuttingTime

%%---------------------------------------------------------------------

%Movement of cuting time
for j = 1:1:20;
   CuttingTime(j,:);

%Formula to find Taylors Tool Equation
   for j = 1:1:20;
      T = a3*sqrt((K/(V(1,j))*((f(1,j))^a1)*((doc(1,j))^a2)));
         TaylorsToolEquation(j,:) = [T];
```

```
   end
end

   disp('The Taylors Tool Equation is:');
      TaylorsToolEquation

%%------------------------------------------------------------------------

%Constraint
   %Formula for Power Limitation
for j = 1:1:20;
      maxP = (0.0373*V(1,j)^0.91*f(1,j)^0.78*doc(1,j)^0.75);
         PowerLimitation(j,:) = [maxP];
end

disp('The Power Limitation is:');
   PowerLimitation

%%------------------------------------------------------------------------

%Formula to find Power Cost
for j = 1:1:20;
      Cp = ((PowerLimitation(j,:)*(CuttingTime(j,:)/60)*(0.20)));
         PowerCost(j,:) = [Cp];
end

   disp('The Power Cost is:');
      PowerCost

%%------------------------------------------------------------------------

%Objective function
   %To optimize turning parameters for minimized production cost per unit
      %Formula Production Cost per unit
for j = 1:1:20;
      Cu =
(((Co*CuttingTime(j,:))+(CuttingTime(j,:)/TaylorsToolEquation(j,:)))*((Co*tcs)+Ct))+(
Co*(th+tR))+(PowerCost(j,:)+Cw);
         ProductCost(j,:) = [Cu];
end
   disp('Objective function');
   disp('The Production Cost per unit is:');
      ProductCost

%%------------------------------------------------------------------------
```

```
%Sort accending
ProductionCost = sort(ProductCost);

   disp('The sort accending of Production Cost value is:');
   disp('The Production cost per unit is:');
      ProductionCost

   disp('The chrom is:');
      chrom

%%----------------------------------------------------------------------

%Find new chrom
for i = 1:20
   for j = 1:20
      if ProductionCost(i,1) == ProductCost(j,1)
         Newchrom(i,:) = [chrom(j,1:15)];
         break
      end
   end
end


   disp('The Newchrom is:');
      Newchrom

%%----------------------------------------------------------------------

%Reproduction
   %Croosover, Mutation, and Trail diffusion
      %Load file from Evaluation
disp('Evaluation');
   Evaluation

disp('Step 3.0: REPRODUCTION');

%%----------------------------------------------------------------------

%Crossover
   %Load file from Reproduction
%Repair solution 13-18
   %6 times loop
for n = 1:6 %Start loop
disp('Reproduction');
   Reproduction
```

```
%Crossover is a first step applied in reproduction
disp('Step 3.1: CROSSOVER');

%%---------------------------------------------------------------------



%%---------------------------------------------------------------------

%Repair the infirior solution from 13 to 18
   %Generate 1 random number from 1 to 12
RandNum = randint (1,2,[1,12]);

   disp('The random number is:')
      RandNum

%%---------------------------------------------------------------------

%Generate solution
SoluGen = Newchrom(RandNum,:);

   disp('The solution ganerate is:')
      SoluGen

%%---------------------------------------------------------------------

%Generate another random number from 1 to 15
   %Generate for saperate solution
SepSol = randint(1,1,[1,14]);

   disp('The random number for Saperate solution is:')
      SepSol

%%---------------------------------------------------------------------

%Generate solution
SepSolgen = Newchrom(SepSol,:);

   disp('The Saperate solution is:')
      SepSolgen

%%---------------------------------------------------------------------

%Parent 1 and Parent 2 for combination
   %Parent 1
disp ('The Parent 1 is:');
```

```
   Parent1a = SoluGen(1,1:SepSol)
   Parent1b = SoluGen(1,SepSol+1:15)

   %Parent 2
disp ('The Parent 2 is:');
   Parent2a = SoluGen(2,1:SepSol)
   Parent2b = SoluGen(2,SepSol+1:15)

%%----------------------------------------------------------------------

%Combination to making Child 1 and Child 2
   %Child 1
disp ('The Child 1 is:');
   Child1 = [Parent1a Parent2b]

   %Child 2
disp ('The Child 1 is:');
   Child2 = [Parent2a Parent1b]

%%----------------------------------------------------------------------

%Compare the value of Child 1 and 2
   %saperate child 1 solution
for i = 5
   Child1sep = [Child1(1,1:i); Child1(1,i+1:10); Child1(1,16-i:15)]
end

%%----------------------------------------------------------------------

%Convert (Child 1 saperate) into binary number
Child1bin = bin2dec(Child1sep)

%%----------------------------------------------------------------------

%Convert decimal number into parameter value
Condecdoc = (3*(Child1sep(1,1))/31)+2
CondecV = (170000*(Child1sep(2,1))/31)+30000
Condecf = (0.508*(Child1sep(3,1))/31)+0.254

%%----------------------------------------------------------------------

   %Saperate child 2 solution
for i = 5;
   Child2sep = [Child2(1,1:i); Child2(1,i+1:10); Child2(1,16-i:15)]
end
```

```
%%----------------------------------------------------------------------

%Convert (Child 2 saperate) into binary number
Child2bin = bin2dec(Child2sep)

%%----------------------------------------------------------------------
            %masalh tak dlm range pun
%Convert decimal number into parameter value
Condec2doc = (3*(Child2sep(1,1))/31)+2
Condec2V = (170000*(Child2sep(2,1))/31)+30000
Condec2f = (0.508*(Child2sep(3,1))/31)+0.254

%%----------------------------------------------------------------------

%Total parameter value
TotalPar1 = Condecdoc+CondecV+Condecf
TotalPar2 = Condec2doc+Condec2V+Condec2f

%%----------------------------------------------------------------------

%Select the smallest parameter value
   %Smallest parameter value give smallest production cost
   %(objective function)
if TotalPar1 > TotalPar2;
   NewSol = Child2;
else TotalPar1 < TotalPar2;
   NewSol = Child1;
end

%%----------------------------------------------------------------------

%The best solution in Crossover
%End loop
Newsolution(n,:)=[NewSol]
end

%%----------------------------------------------------------------------

%Find old inferior solution
   %1-12 (Superior)
for i= 1:1:12;
   Superior=Newchrom(i,:);
   Superiorchrom(i,:)=[Superior];
end
Superiorchrom
```

```
%%----------------------------------------------------------------------

   %19-20 (Inferior)
   %Trail diffusion
for i=19:20;
   InfTra=Newchrom(i,:);
   InfTrachrom(i-18,:)=[InfTra];
end
InfTrachrom %InfTrachrom = Inferior for trail diffusion chrom

%%----------------------------------------------------------------------

%Combine new solution and old solution
for j=1:1:20;
  NewC=[Superiorchrom;Newsolution;InfTrachrom];
  NewChromAC=[NewC];   %NewChromAC = Newchrom after crossover
end

   disp('The new chrom after crossover is:');
      NewChromAC
%All solution has been replace with new solution

%%----------------------------------------------------------------------

%Mutation
   %Load file from Crossover
disp('Crossover');
   Crossover

%Mutation is a second step applied in reproduction
disp('Step 3.2: MUTATION');

%%----------------------------------------------------------------------

%Repair the infirior solution from 13 to 18
for j = 1:1:6;    %loop

%%----------------------------------------------------------------------

%Set the level of Mutation
   %Probability Mutation

disp('The Probability of Mutation is:');
   ProM = 0.45     %ProM = Probability of Mutation

%%----------------------------------------------------------------------
```

```
%Generate random number from 0 to 1 for level of Mutation
disp('The level of Mutation is:');
    LevM = rand(1,1)    %LevM = level of Mutation

%%----------------------------------------------------------------------

%Mutation condition
    %Two condition of Mutation
        %Level of mutation is lower than Probility of mutation (LevM < ProM)
            %Did not applied for mutation procedure
if LevM < ProM
    disp('Mutation procedure was not implemented')
   for i = 1:1:6;
       w = Newsolution(i,:);
       for i = j;
           z = Newsolution(j,:);
           ChildNew(j,:) = [z];
       end
   end
ChildNew

%%----------------------------------------------------------------------
        %Level of mutation is higher or equal Probility of mutation (LevM >= ProM)
            %Applied Mutation Procedure
else LevM >= ProM
    disp('Mutatin procedure was implemented')

  for r = 1:1:6;
       y = Newsolution(r,:);
       for r = j;
           z = Newsolution(j,:);
           SolChrom(r,:) = [z];
       end
  end
SolChrom

%%----------------------------------------------------------------------

%Generate randomly from 1-15 represent as the number of binary number

SepChrom = randint(1,1,[1,14]);   %SepChrom = Seperate chrom fror mutataion
procedure

disp('The random number to separate the solution is:');
SepChrom
```

```
Child1 = SolChrom(1,1:SepChrom)
Child2 = SolChrom(1,SepChrom+1:15)

%%-----------------------------------------------------------------------

%Arrange child2 at the front side and child1 at the back of solution

ChildNew(j,:) = [Child2 Child1];

disp('The new child is:');
ChildNew
end

ChildNewX(j,:) = ChildNew(j,:)

end

%%-----------------------------------------------------------------------

%Find the old inferior solution for mutation level

for i = 1:1:12;
   P = NewChromAC(i,:);
   Superiorchrom2(i,:) = [P];    %Superiorchrom2 = Superioe chrom (1to12)
end
Superiorchrom2
for i = 19:20;
   B = NewChromAC(i,:);
   InfTrachrom2(i-18,:) = [B];     %InfTrachrom2 = Infirior for trail diffusion chrom
end
InfTrachrom2

%%-----------------------------------------------------------------------

%Combine the old solution with the new solution

for j = 1:1:20;
  New = [Superiorchrom2;ChildNewX;InfTrachrom2];
  NewChromAM = [New];    %NewChromAM = New chrom at mutation
end

disp('The new chrom after mutation is:');
NewChromAM

%%-----------------------------------------------------------------------
```

```
%Trail diffusion
   %Load file from Mutation
%Repair the inferior solution 19 to 20
for a = 1:2;    %Loop
disp('Mutation');
   Mutation

%Trail diffusion is a third step applied in reproduction
disp('Step 3.3: TRAIL DIFFUSION');

%%----------------------------------------------------------------------



%%----------------------------------------------------------------------

%Ganerate 2 random number from 1 to 12
DifNu = randint (1,2,[1,12]);    %DifNu = diffrent number

   disp('The two diffrent number is:');
      DifNu

%%----------------------------------------------------------------------

%Take new solution from mutation
ChromTra = NewChromAM(DifNu,:);     %ChromTra = Chrom trail diffusion

   disp('The solution generate based on mutation is:');
      ChromTra

%%----------------------------------------------------------------------

%Convert the solution into binary number
P1Tra = bin2dec(ChromTra(1,:))      %P1Tra = Parent1 trail diffusion
P2Tra = bin2dec(ChromTra(2,:))      %P2Tra = Parent2 trail diffusion

%%----------------------------------------------------------------------

%Second step
   %Ganerate random number of alpha from 0.0 to 1.0
alpha = rand(1,1)   %Probability

%%----------------------------------------------------------------------

%Find the Xchild
%Rules
```

```
   %(alpha <= 0.5)
if alpha <= 0.5;
    Xchild = (alpha)*P1Tra+(1-alpha)*P2Tra;

   %(alpha > 0.5)
else  alpha > 0.5;
     Xchild = P2Tra;
end

   disp('The trail diffusion child, Xchild is:')
      Xchild

%%-----------------------------------------------------------------------

%Convert the child into binary number
Ch1(1,1:15) = dec2bin(Xchild,15);   %generate 15 bit binary number

   disp('The new child is:')
      Ch1

CTra(a,:) = [Ch1]    %CTra = Child trail diffusion
end

%%-----------------------------------------------------------------------

%Find the old inferior solution
for i = 1:1:18;
   NM = NewChromAM(i,:);
   Mchrom(i,:) = [NM];
end
Mchrom

%%-----------------------------------------------------------------------

%Combine inferior solution
   %ganerate solution for trail diffusion
for j=1:1:20;
  DT=[Mchrom;CTra];
  NewChromTD=[DT];      %NewChromTD = New chrom for trail diffusion
end

   disp('the new chrom is:');
      NewChromTD

%All solution has been replaced
```

```
%%------------------------------------------------------------------------

%Local Search
   %Load file from Trail Diffusion
disp('TrailDiffusion');
   TrailDiffusion

%Step applied after Trail Diffusion
   %Superior region only
disp('Step 4.0: LOCAL SEARCH');

%%------------------------------------------------------------------------

%The local search improves the 12 solutions in the superior region only
   %Initially the pheromone value (Ph) for every region is set 1.0
   %the age for every region is taken as 10
Ph = 1;     %Ph = Pheromone value
Age = 10;   %Age = Age for every region
n = 12;     %n = Solution

%%------------------------------------------------------------------------

%Find the old solution from Evaluation
for i=1:1:12
Xold(i,:) = Newchrom(i,:);  %Xold = Old solution from Evaluation
end

   disp('The old solution from Evaluation is:');
      Xold

%%------------------------------------------------------------------------

%Convert the old solution to decimal number
for i=1:1:12;
   Xold(i,:);
   for j=1:1:12;
      Old=bin2dec(Xold(j,:));
      XoldDec(j,:)=[Old];    %XioldDec = X iteration old in dec number
   end
end

   disp('The old solution in decimal number is:');
      XoldDec

%%------------------------------------------------------------------------
```

```
%Limiting step
K1 = rand(1,1)
K2 = rand(1,1)

%Rules for limiting step
if K1 > K2;
   LimSt = K1-(Age*K2);    %LimSt = Limiting step
else K1 < K2;
   LimSt = K2-(Age*K1);
end

   disp('The limiting step is:');
      LimSt

%%------------------------------------------------------------------------

%Ganerate random number between 0 to 1
r = rand(1,1)

%Find the new solution
if r > 0.5;
   Xinew = XoldDec+LimSt;   %Xinew = X iteration new
else r < 0.5;
   Xinew = XoldDec-LimSt;
end

   disp('The new solution,Xinew is:');
      Xinew

%Convert solution into binary number
Xinewchrom = dec2bin(Xinew);  %Xinewchrom = X iteration new chrom

   disp('The new superior chromozon is:');
      Xinewchrom

%%------------------------------------------------------------------------

%Find the old inferior solution from trail diffusion
for i = 13:1:20;
   N = NewChromTD(i,:);
   NTDchrom(i-12,:) = [N];   %NTDchrom = Chrom at trail difusion
end
NTDchrom

%%------------------------------------------------------------------------
```

```
%Combine the chrom from trail difusion with new solution
for j = 1:1:20;
  New =[Xinewchrom;NTDchrom];
  NewChromLS = [New];  %NewChromLS = New chrom at Local Search
end

   disp('The new chrom at Local Search is:');
      NewChromLS

%%-----------------------------------------------------------------------

%Update Trail
   %Load file from Local search
disp('LocalSearch');
   LocalSearch

%Step applied after Local Search
disp('Step 5.0: UPDATE TRAIL');

%%-----------------------------------------------------------------------

%New Pheromone Value
   %lLoad file from Update Trail
disp('UpdateTrail');
   UpdateTrail

%This step applied for update trail
disp('Step 5.1: NEW PHEROMONE VALUE');

%%-----------------------------------------------------------------------

%Consider the fitness value for solution 0 to 50
   %50 = fitness solution
   %32767 = 111111111111111 (15 value of 1 in decimal)
   %655.34 = 32767/50
FXinew = 50-((32767-Xinew)/655.34);   %FXinew = Fitness value for iteration new
solution
FXiold = 50-((32767-XoldDec)/655.34); %FXiold = Fitness value for iteration old
solution

   disp('The fitness value for new solution,FXinew is:');
      FXinew

   disp('The fitness value for old solution,FXiold is:');
      FXiold
```

```
%%----------------------------------------------------------------------

%Calculate the Age new
   %Two rules
for i=1:1:12
   if FXinew < FXiold;
      AgeNew(i,:) = Age+1;
   else FXinew > FXiold;
      AgeNew(i,:) = Age-1;
   end
end

   disp('the new age for the solution is:')
      AgeNew

%%----------------------------------------------------------------------

%The new pheromone is calculated by using the following expression
for i=1:1:12;
   PhiNew(i,:) = ((FXinew(i,:)-FXiold(i,:))/FXiold(i,:))+Ph;
end


   disp('The new Pheromone value is:');
      PhiNew      %PhiNew = Pheromone iteration new

%%----------------------------------------------------------------------

%Calculate the Pheromone Average using the Pheromone New value
PhAve = mean(PhiNew);

   disp('The new Pheromone Average is:');
      PhAve

%%----------------------------------------------------------------------

%Termination/Iteration
   %Load file from New Pheromone Value
for m = 1:100;  %100 Iteration

disp('NewPheromoneValue');
   NewPheromoneValue

%This is last step applied for Ant Colony Method
   %Use to set number of iteration
disp('Step 6.0: TERMINATION/ITERATION');
```

```
%%----------------------------------------------------------------------

%Saperate the solution from Local Search
for j = 1:20;
for i = 1:1:20;
      W = NewChromLS(i,:);
      for i = j;
         G = [NewChromLS(i,1:5);NewChromLS(i,5+1:10);NewChromLS(i,16-5:15)];
      end
end

%%----------------------------------------------------------------------

%Convert the binary number into decimal number
n = bin2dec(G);

%%----------------------------------------------------------------------

%Convert decimal number into parameter value
Pardoc = (3*(n(1,1))/31)+2          %Pardoc = Parameter value for depth of cut
ParV = (170000*(n(2,1))/31)+30000   %ParV = Parameter value for cutting speed
Parf = (0.508*(n(3,1))/31)+0.254    %Parf = Parameter value for feed rate

%%----------------------------------------------------------------------

%Change parameter into matrix position
Matdoc(1,i) = Pardoc;   %Matdoc = Matrix depth of cut
MatV(1,i) = ParV;       %MatV = Matrix cutting speed
Matf(1,i) = Parf;       %Matf = Matrix feed rate
end

%%----------------------------------------------------------------------

%Constant value from objective function
disp('The constant value is:')
   L = 203         %Length of workpiece (mm)
   D = 152         %Diameter of workpiece (mm)
   %T = 35          %Tool life (min)
   %Rmax = 12000    %Maximum surface roughness of rough (m)
   %Pmax = 5000     %Power of the machine (kW)
   %Fmax = 900      %Cutting force (N)
   %Omax = 500      %Temperature of toll workpiece interface (oC)
   a1 = 0.29       %Constatnts used in tool life equation
   a2 = 0.35       %Constatnts used in tool life equation
   a3 = 0.25       %Constatnts used in tool life equation
```

```
   K = 193.3      %Constatnts used in tool life equation
   tcs = 0.5      %Tool change time (min/edge)
   tR = 0.13      %Quick return time(min/pass)
   th = 1.5       %Tool production time (min)
   Co = 0.28      %Operating cost (RM/piece)
   Ct = 1.41      %Tool cost per cutting edge (RM/piece)
   Cw = 1.65      %Workpiece cost (RM)

%%-----------------------------------------------------------------------

%Movement of solution
for i = 1:1:20;
      Matdoc(1,i);
      MatV(1,i);
      Matf(1,i);

%Formula to find cutting parameter per pass
   %Using 1 and j for random sata
   for j = 1:1:20;
      tm = (D*L)/(1000*MatV(1,j)*Matf(1,j));
         CuttingTimeX(j,:) = [tm];
   end
end

   disp('The cutting time per pass is:');
      CuttingTimeX

%%-----------------------------------------------------------------------

%Movement of cuting time
for j = 1:1:20;
   CuttingTimeX(j,:);

%Formula to find Taylors Tool Equation
   for j = 1:1:20;
      T = a3*sqrt((K/(MatV(1,j))*((Matf(1,j))^a1)*((Matdoc(1,j))^a2)));
         TaylorsToolEquationX(j,:) = [T];
   end
end

   disp('The Taylors Tool Equation is:');
      TaylorsToolEquationX

%%-----------------------------------------------------------------------

%Constraint
```

```
   %Formula for Power Limitation
for j = 1:1:20;
      maxP = (0.0373*MatV(1,j)^0.91*Matf(1,j)^0.78*Matdoc(1,j)^0.75);
         PowerLimitationX(j,:) = [maxP];
end

disp('The Power Limitation is:');
   PowerLimitationX

%%-----------------------------------------------------------------------

%Formula to find Power Cost
for j = 1:1:20;
      Cp = ((PowerLimitationX(j,:)*(CuttingTimeX(j,:)/60)*(0.20)));
         PowerCostX(j,:) = [Cp];
end

   disp('The Power Cost is:');
      PowerCostX

%%-----------------------------------------------------------------------

%Objective function
   %To optimize turning parameters for minimized production cost per unit
      %Formula Production Cost per unit
for j = 1:1:20;
      Cu =
(((Co*CuttingTimeX(j,:))+(CuttingTimeX(j,:)/TaylorsToolEquationX(j,:)))*((Co*tcs)+
Ct))+(Co*(th+tR))+(PowerCostX(j,:)+Cw);
         ProductCostX(j,:) = [Cu];
end
   disp('Objective function');
   disp('The Production Cost per unit is:');
      ProductCostX

%%-----------------------------------------------------------------------

%Sort production cost
L = sort(ProductCostX);

   disp('The best PRODUCTION COST PER UNIT is:')
      ProductionCOST = L(1,1) %First solution is the best

%%-----------------------------------------------------------------------

%Find the solution for the best production cost
```

```
for i = 1:20;
   for j = 1:20;
      if L(1,1) == ProductCostX(j,:);
         Newc(1,:) = [NewChromLS(j,:)];
         break
      end
   end
end
BestChrom = Newc

%%----------------------------------------------------------------------

%Colect the production cost
PC(m,:) = [ProductionCOST];

   disp('The production cost for 100 generation is:');
      PC

%%----------------------------------------------------------------------

%Collect the best solution
BS(m,1:15) = [BestChrom];
   disp('The best solution for 100 generation is:');
      BS
end

%%----------------------------------------------------------------------

%Sort the production cost for easily take the best solution
BestPC = sort(PC);  %BestPC = Best production cost

%%----------------------------------------------------------------------

%Choose the best solution from 100 generation
for i = 1:100;
   for j = 1:20;
      if BestPC(1,1) == PC(i,:);
         BestSolution(1,:) = [BS(i,1:15)];
         break
      end
   end
end
BestSolution

%%----------------------------------------------------------------------
```

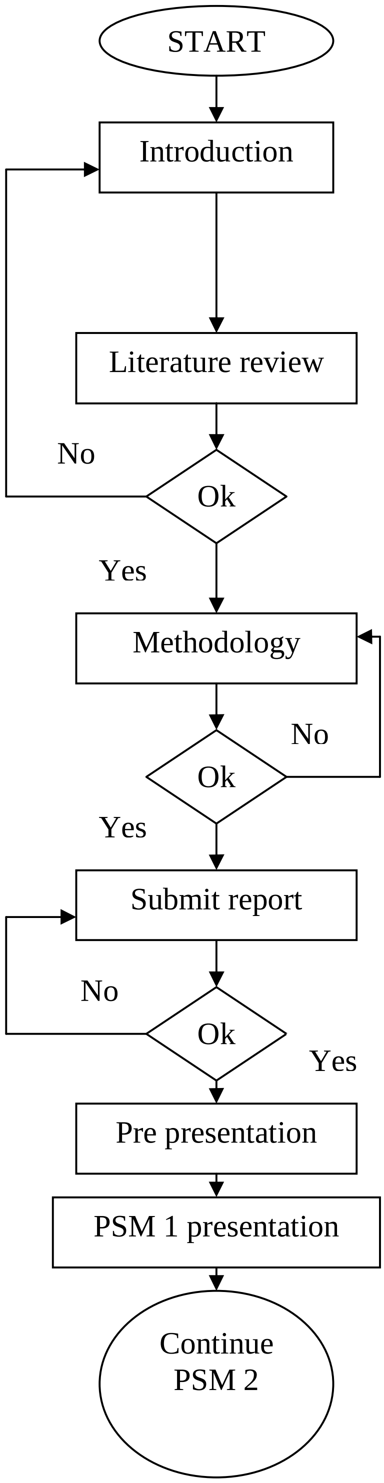
```
%Seperate the digit for the best solution
SepBS = [BestSolution(1,1:5);BestSolution(1,5+1:10);BestSolution(1,16-5:15)];

%%----------------------------------------------------------------------

%Convert the Seperate solution into decimal number
DX = bin2dec(SepBS);

%%----------------------------------------------------------------------

%Plot the graph
   %Graph 1
x = [1:100]
y = [PC]
plot(x,y),xlabel('Generation'),ylabel('Optimum Production Cost'),title('Optimum
Production Cost vs Generation')
grid on,

%%----------------------------------------------------------------------

   %Graph 2
for i = 1:100;
   if i == 1;
      BO(i,1) = PC(i,:);
   elseif PC(i,:) < BO(i-1,:);
      BO(i,1) = PC(i,:);
   else PC(i,:) >= BO(i-1,:);
      BO(i,:) = BO(i-1,:);
   end
end
BO  %Best Overall for Production Cost

x = [1:100]
y = [BO]
plot(x,y),xlabel('Generation'),ylabel('Best Overall for Production Cost'),title('Best
Overall for Production cost vs Generation')
grid on,

%%----------------------------------------------------------------------
for w = 1:100;

%%----------------------------------------------------------------------

%Saperate the digit of the best solution
Sep = [BS(w,1:5);BS(w,5+1:10);BS(w,16-5:15)];
```

```
%%----------------------------------------------------------------------

%Convert into decimal number
xn = bin2dec(Sep);

%%----------------------------------------------------------------------

%Convert into parameters value

DOCut = (3*(xn(1,1))/31)+2;
CSpeed = (170000*(xn(2,1))/31)+30000;
FRate = (0.508*(xn(3,1))/31)+0.254;

D(w,:) = [DOCut];
C(w,:) = [CSpeed];
F(w,:) = [FRate];
end

%%----------------------------------------------------------------------

%Sort the values
Depth_Of_Cut = sort(D);
Cutting_Speed = sort(C);
Feed_Rate = sort(F);

%%----------------------------------------------------------------------

%Display the values
Depth_Of_Cut
Cutting_Speed
Feed_Rate

%%----------------------------------------------------------------------
%Take the production cost value among 100 generation
disp('The best PRODUCTION COST is:');
   BestPC(1,1)
%%----------------------------------------------------------------------

%Convert the decimal number into parameter value
TheBestDepthOfCut = (3*(DX(1,1))/31)+2;
TheBestCuttingSpeed = (170000*(DX(2,1))/31)+30000;
TheBestFeedRate = (0.508*(DX(3,1))/31)+0.254;

   disp('The best Depth Of Cut is:');
      TheBestDepthOfCut
   disp('The best Cutting Speed is:');
```

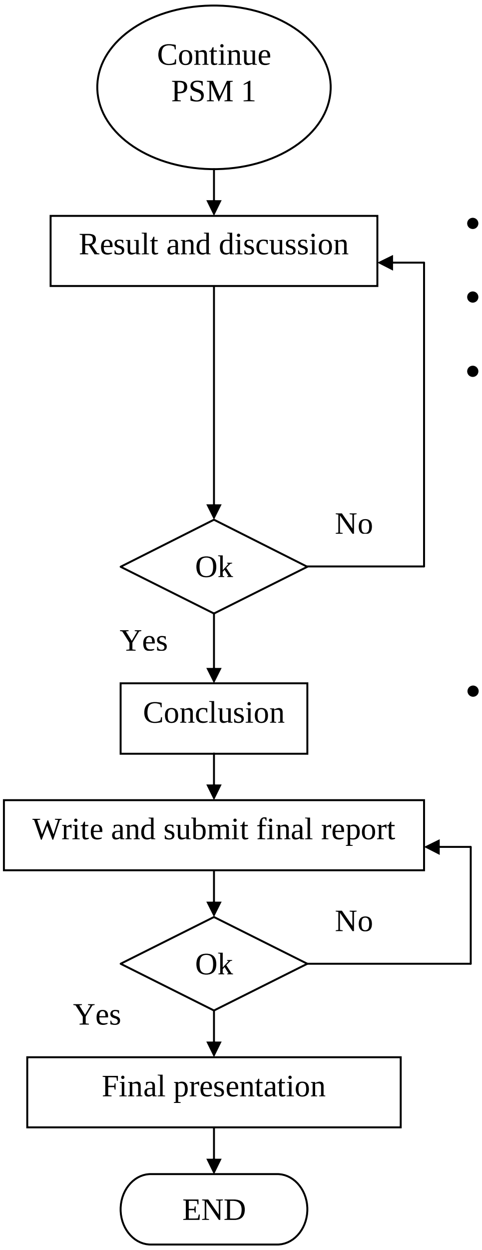
```
        TheBestCuttingSpeed
    disp('The best Feed Rate is:');
        TheBestFeedRate

%%----------------------------------------------------------------------

disp('Optimization Turning Parameters Using Ant Colony Method')
    disp('By: MOHAMAD NAZRI BIN SEMOIN')
    disp('Supervisor: MR. MOHD FADZIL FAISAE AB. RASHID')
    disp('University Malaysia Pahang')
```

APPENDIX B

FLOW CHART FOR PSM



- Set the objective and scope.
- Determine the problem statement,
- Determined the previous method used.
- Determine constant parameters and suitable formulas.
- Determined about ACO step, function and rules.
- Determined the objective function.



- Generate the ACO using Matlab to find the result.
- Compare result from previous technique
- Discuss about the result.

- Recommendation for this project.

APPENDIX C

GANTT CHART PSM 1

Project activities	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16
INTRODUCTION																
Project briefing	√	√														
Objective and scope	√	√														
Problem statement			√													
Flowchart				√												
LITERATURE REVIEW																
Internets and books				√	√	√	√									
Previous method				√	√	√	√									
Parameters used				√	√	√	√									
Data of problem							√	√	√	√						
Optimize turning formula							√	√	√	√	√					
ACO										√	√	√				
METHODOLOGY																
ACO step & rule									√	√	√	√				
Objective function											√	√	√	√		
SUBMIT AND REWRITE																
Submit report PSM 1													√	√		
Rewrite report PSM 1													√	√		
FINAL 1																
Pre-presentation															√	√
PSM 1 presentation															√	√

GANTT CHART PSM 2

Project activities	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16
RESULT AND DISCUSSION																
Initialization	√	√														
Evaluation		√	√													
Reproduction			√	√	√											
Local search					√	√										
Update trail						√	√									
Termination								√	√							
Compare result										√						
Discussion											√					
CONCLUTION																
Recommendation												√				
Conclude project													√			
SUBMIT AND REWRITETEN																
Submit final report														√		
Rewritten final report															√	
FINAL																
Final presentation																√