CHAPTER - 10

DEVELOPMENT OF ALGORITHM, FLOW CHART AND COMPUTER PROGRAMME (USING C LANGUAGE) TO PREDICT THE OPTIMUM DESIGN OF FEED MECHANISM FOR FINE BORING MACHINE

10.1 INTRODUCTION TO ALGORITHM DEVELOPMENT

In this chapter an algorithm and a computer programme are developed based on cost analysis (Ref. Chapter 7), analysis by theory of value (Ref. Chapter 8) and quality analysis (Ref. Chapter 9) of three feed mechanisms of fine boring machines. A flow chart and computer programme using C language are developed. This algorithm and programme helps to predict the optimum design of feed mechanism for fine boring machine. The method to predict optimum design using cost analysis have received a lot of attention among researchers. Hence development of this algorithm and computer programme helps researchers to get immediate results for different input parameters.

For predicting the optimum design of feed mechanisms for fine boring machine taken for our research based on cost analysis, input parameters are diameter of bore to be machined, feed rate, bore length, rapid travel, rapid rate, cutting speed, cost of equipment, life in hours, working hours per annum, scrap value, maintenance hours, cost of repair and maintenance, cost of overhead per annum, cost of wages per annum, feed motor power and electricity charges.
The output parameters are cycle time, machine hour rate, cost of component, revenue earned per year, return on investment, profitability index, profits, contribution, profit volume ratio, break even point and margin of safety. A programme in "C: is written to find out the output parameters, based on the algorithm developed. The output parameters depend on input parameters. Programme is written also to predict the optimum feed mechanism based on the value of output parameters either maximum value or minimum value. Hence by giving different input parameters for different feed mechanisms, the corresponding output parameters can be obtained. After that programme is written to compare the output parameters and optimum feed mechanism is predicted based on comparison by the computer.

In quality analysis the data from samples of fine bored component was taken as input parameter. The part cast iron cylinder whose bore was machined using fine boring machine was inspected and results were tabulated as shown in Table 9.1. Specification of bore is 50 ± 0.05. Various sets of readings were obtained for components made out of 3 designs namely electromechanical design, hydraulic design and CNC design. Algorithm and computer programme is written to find out the output parameters and hence the optimum design. The output parameters for quality analysis are arithmetic mean, range, standard deviation, variance process capability $C_p$, process capability index $C_{pk}$. 
Input parameters based on feed mechanism is given in the computer. The output parameters can be obtained and programme is written to predict the optimum design based on value of output parameters.

10.2 ALGORITHM TO FIND THE OPTIMUM FEED MECHANISM BY COST ANALYSIS

10.2.1 Cycle Time

Input parameters

- N: Speed of spindle - rpm
- D: Diameter of bore - mm
- F.R: Feed rate - mm/rev.
- B.L: Bore length - mm
- R.T: Rapid travel - mm
- R.R: Rapid rate - min/min.
- C.S: Cutting speed - m/min.
- M.T: Machining time - sec.
- C.T: Cycle time - sec.

(Rapid travel for forward and backward are assumed to be equal)

Step 1

To find the speed of tool - (N)

\[
\text{Cutting speed} = \frac{\pi DN}{1000}
\]

\[
\text{C.S} = \frac{\pi DN}{1000}
\]
Step 2
To find feed / min.
Feed / min = N x F.R.

Step 3
To find machining time
M.T. = \( \frac{B.L}{\text{Feed / min.}} \) x 60

Step 4
To find time for rapid travel
T.R.T. = \( \frac{R.T}{R.R} \) x 60

Step 5
To find cycle time
C.T = Time for home to right + Time for right to left + Time for left to home + Right side machine time + Left side machine time
C.T = T.R.T (1+2+1) + M.T (1+1)

\[ C.T = 4(T.R.T) + 2(M.T) \]

Step 6
To find the cycle time for different feed mechanism.

Step 7
To compare cycle time for different feed mechanism.

Step 8
Select optimum being whichever is least.
10.2.2 Machine hour rate (M.H.R)

Input Parameters

- C.E: Cost of equipment in rupees
- L.H: Life in hours
- W.H/Annum: Working hours per annum
- S.V: Scrap value in rupees
- M.H: Maintenance hours
- C.R.M/Annum: Cost of repair and maintenance in rupees
- C.O.H: Cost of overhead per annum
- C.W: Cost of wages per annum
- P: Motor power in H.P.
- E.C: Electricity charges per KW hour

Step 1

To find the actual working hours per annum.

\[ A.W.H = W.H - M.H. \]

Step 2

To find fixed expenses per hour

\[ F.E / \text{hour} = \frac{C.O.H + C.W.}{A.W.H.} \]

Step 3

To find depreciation charges per hour.

\[ D.C. / \text{hour} = \frac{C.E - S.V.}{L.H.} \]
Step 4

To find cost of repair and maintenance per hour

\[
\text{C.R.M. / hour} = \frac{\text{C.R.M. / Annum}}{\text{A.W.H. / Annum}}
\]

Step 5

To find cost of power / cycle

\[
\text{C.P.} = P \times 0.736 \times \frac{\text{E.C.} \times \text{C.T.}}{3600}
\]

Step 6

To find machine hour rate

\[
\text{M.H.R.} = \frac{\text{F.E. / hour}}{3600} + \frac{\text{D.C. / hour}}{3600} + \frac{\text{C.R.M. / hour}}{3600} + \frac{\text{C.P.}}{3600}
\]

10.2.3 Cost of component (C.C)

S.T. - Setup time in second
C.T. - Cycle time in second

Step 1

To find total time

\[
\text{T.T.} = \text{C.T.} + \text{S.T.}
\]

Step 2

To find cost of component

\[
\text{Cost of component} = \frac{\text{M.H.R.} \times \text{T.T.}}{3600} + \text{Profit}
\]
10.2.4. **Revenue Earned / year**

R.E. / years

**Step 1**

To find number of pieces machined per day

\[
\text{N.P.M. / day} = \frac{2 \times \text{working time / day in hour}}{\text{T.T.}}
\]

Working time per day is assumed to be 14 hrs (2 shifts)

\[
= \frac{2 \times \text{W.T. / day in hours} \times 3600}{\text{T.T.}}
\]

**Step 2**

To find revenue earned / year

\[
\text{R.E. / year} = \text{C.C.} \times (\text{N.P.M. / day}) \times \text{(working days per year)}
\]

**Step 3**

To find R.E./year for different feed mechanisms.

**Step 4**

To compare revenue per year for different feed mechanisms.

**Step 5**

Select the optimum being whichever is higher.
10.2.5 Return on investment (ROI)

10.2.5.1 Average rate of return method

**Step 1**

To find average rate of return $A.R.R. = \frac{R.E. \text{ / year}}{C.E.}$

**Step 2**

To find $A.R.R.$ for 3 feed units.

**Step 3**

To compare $A.R.R.$ for different feed mechanisms.

**Step 4**

Select the optimum being whichever is higher.

10.2.5.2 Payback period method

**Step 1**

To find payback period

$P.B.P. = \frac{1}{A.R.R.}$

**Step 2**

To find $P.B.P.$ for 3 feed units.

**Step 3**

To compare $P.B.P.$ for different feed units.

**Step 4**

Select the optimum being whichever is least.
10.2.5.3 Internal rate of return method

\[ A_t \] - Cash flow for period \( t \)
\[ r \] - Internal rate of return
\[ n \] - last period where cash flow expected

**Step 1**
To find \( r \) from equation

\[
\sum_{t=0}^{n} \left[ \frac{A_t}{(1+r)^t} \right] = 0
\]

**Step 2**
\[ A_t = A_0 = \text{C.E.} \]
\[ A_1 = A_2 = A_3 = A_4 = A_5 = \text{R.E. / year} \]

**Step 3**
Find \( r \) using

\[
\text{C.E.} = \frac{\text{R.E. / year}}{(1+r)} + \frac{\text{R.E. / year}}{(1+r)^2} + \frac{\text{R.E. / year}}{(1+r)^3} + \frac{\text{R.E. / year}}{(1+r)^4} + \frac{\text{R.E. / year}}{(1+r)^5}
\]

**Step 4**
To find internal rate of return for 3 feed units.

**Step 5**
To compare I.R.R. for 3 feed units.
Step 6

Select the optimum being whichever is higher.

10.2.5.4 Net present value method

K - rate of return

Step 1

Find N.P.V.

\[ \text{N.P.V.} = \sum_{t=0}^{n} \left[ \frac{A_t}{(1+k)^t} \right] = 0 \]

Step 2

\[ \text{N.P.V.} = - \text{C.E.} + \frac{\text{R.E. / year}}{(1+k)^1} + \frac{\text{R.E. / year}}{(1+k)^2} + \frac{\text{R.E. / year}}{(1+k)^3} + \frac{\text{R.E. / year}}{(1+k)^4} + \frac{\text{R.E. / year}}{(1+k)^5} \]

Step 3

To find N.P.V. for 3 feed units.

Step 4

To compare N.P.V. for 3 feed units.

Step 5

Check whether it is positive, if positive feasible.

Step 6

Find the optimum being which is maximum.
10.2.5.5 Profitability index

Step 1

To find profitability index.

\[
P.I. = \frac{\sum_{t=1}^{n} \frac{A_t}{(1+k)^t}}{A_0}
\]

Step 2

To find P.I. for 3 feed units.

Step 3

To compare P.I. for 3 feed units.

Step 4

Find the optimum being whichever is higher.

10.2.6 Break even point

Step 1

To find variable cost per hour.

\[
V.C. / \text{hour} = D.C. / \text{hour} + C.P. + C.R. \text{ M/hour}
\]

Step 2

To find variable cost per component.

\[
V.C. / \text{component} = \frac{V.C. / \text{hour} \times \text{T.T.}}{3600}
\]
Step 3

To find annual sales.

A.S. = N.M. / day x Working days in a year

Step 4

To find total cost

T.C. = V.C. / component x A.S. + (C.O.H. + C.W.)

Step 5

(a) To find profit

Profit = R.E. / year - T.C.

(b) To find profit for 3 feed units.

(c) To compare profit for 3 feed units.

(d) Find the optimum being whichever is higher

Step 6

To find contribution

Con = C.C. - V.C. / component

Step 7

To find profit volume ratio

P/V ratio = \[
\frac{\text{Con}}{\text{C.C.}} \times 100
\]
Step 8
(a) To find break even point
\[ \text{B.E.P.} = \frac{\text{C.O.H.} + \text{C.W.}}{\text{Con}} \]
(b) To find break even point for 3 feed units
(c) Compare B.E.P. for 3 feed units.
(d) To select the optimum being whichever is least.

Step 9
(a) To find break even point in rupees
\[ \text{B.E.P. (Rup)} = \frac{\text{C.O.H.} + \text{C.W.}}{\text{P/V ratio}} \]
(b) To find B.E.P. (Rup) for 3 feed units.
(c) Compare B.E.P. (Rup) for 3 feed units.
(d) Select the optimum being whichever is higher.

Step 10
(a) To find margin of safety.
\[ \text{M.O.S.} = \text{R.E./year} - \text{B.E. (rup)} \]
(b) To find M.O.S. for 3 feed units.
(c) To compare M.O.S. for 3 feed units.
(d) Find the optimum being whichever is higher.
10.3 ALGORITHM TO FIND THE OPTIMUM BY THEORY OF VALUE

Step 1

To find independent design variables.

Step 2

To find design characteristics.

Step 3

To find value categories.

Step 4

To find value categories for 3 feed units.

Step 5

To compare value categories for 3 feed units.

Step 6

Select the optimum being whichever is higher.

10.4 ALGORITHM TO FIND THE OPTIMUM BY QUALITY ANALYSIS

Input data

Sample data. (Refer Table 9.1)

Step 1

(a) To find arithmetic mean using \( \bar{x} = \frac{\sum_{i=1}^{n} f_i x_i}{n} \)

(b) To find arithmetic mean for 3 feed units.
Step 2
(a) To find range
Range = Maximum limit - Minimum limit
(b) To find range for 3 feed units.

Step 3
(a) To find standard deviation
\[ \sigma = \sqrt{\frac{\sum fx^2 - n\bar{x}^2}{n}} \]
(b) To find standard deviation for 3 feed units.

Step 4
(a) To find variance \( \sigma^2 \).
(b) To find variance for 3 feed units.

Step 5
(a) To find process capability
\[ Cp = 6\sigma \]
(b) To find process capability of machine used by 3 designs.

Step 6
To find process capability index \( C_{pk} \).
\[ C_{pk} = \frac{USL - LSL}{6\sigma} \]
Upper specification limit - lower specification limit
\hline Process capability
(b) To find \( C_{pk} \) of machines used by 3 feed units.
(c) Check \( C_{pk} \)
if \( C_{pk} < 1 \) process not capable
if \( C_{pk} = 1 \) process just capable
if \( C_{pk} > 1 \) process highly capable
10.5 FLOW CHART TO PREDICT OPTIMUM FEED MECHANISM OF FINE BORING MACHINE

Int i, nwt
Float pi = (22/7)
Char type [3] [20]:

Strcpy (type [0], "Hydraulic")
Strcpy (type [1], "Electro Mechanical")
Strcpy (type [2], "CNC")

Which (n ≠ 8)

Switch (n)

Case 1 Prog 1(); Break;
Case 2 Prog 2(); Break;
Case 3 Prog 3(); Break;
Case 4 Prog 4(); Break;
Case 5 Prog 5(); Break;
Case 6 Prog 6(); Break;
Case 7 Prog 7(); Break;

Return to the case

Stop
int i, b
float (d, cs, n, tr, bl, rt, rr[3], mt, trt[3])
Feed, opt:

input
cycle time, bore length
feed length, feed rate
rapid travel

For i = 0; i < 3

input rapid rate

i++

Output machine tools program

\[ n = \frac{(1000 \times c6) \times (Pl \times d)}{Fr} \]
Feed = \( n \times Fr \);
mt = (bt \times 60) / Feed

For i = 0; i < 3

trt[i] = rt \times 60 / rr[i]
ct[i] = 4 \times trt[i] + 2 \times mt

i++

opt = ct[0]
b = 0

For i = 0; i < 3

if dpt > ct[i]
    Yes
    opt = ct[i]
b = i

No

i++
For $i = 0 \leq i < 3$

Output

Cutting speed, feed per minute

Machining time, time for rapid travel cycle time

Output optimum cycle time

For $i = 0 \leq i < 3$

Log = 1

While $i = 1$

Go to sub prog 2 (1)

Output enter 1 to edit previous data

Enter 0 to proceed

Input Log

Return to case while

$i++$

For $i = 0 \leq i < 3$

$awh = wh - mh$

$Fe[i] = ((coh[i] + cw[i]) / awh)$

$(crmh[i] = crma[i]) / awh$
\[ \text{ac} \{ i \} = (\text{ce} \{ i \} \times \text{sv} \{ i \}) / \text{awh} \]
\[ \text{cp} \{ i \} = (c[i] \times 0.736 \times 10^\text{ct} \{ i \}) / 360 \]
\[ \text{mhrh} \{ i \} = \text{Fe} \{ i \} + \text{dc} \{ i \} + \text{crmh} \{ i \} + \text{cp} \{ i \} \]

For \( i = 0 \leq 3 \) 
- Go to Subdisp 2 \( (i) \)
- \( i++ \)

**Input**
- Cost of equipment, \( \text{ce} \{ i \} \)
- Life in hours, \( \text{lh} \)
- Working hours per annum, \( \text{wh} \)
- Scrap value, \( \text{sv} \{ j \} \)
- Maintenance hours, \( \text{mh} \)
- Cost of repair and maintenance \( \text{crma} \{ i \} \)
- Cost of overhead \( \text{coh} \{ i \} \)
- Wages per annum \( \text{cw} \{ j \} \)
- Motor power in HP; \( p \{ i \} \)
- Electricity charges per unit \( \text{ec} \)

**Output**
- Actual working hrs
- Fixed working hours, cost of repair
- Depreciation charges
- Cost of power
- Machine hour rate
int i, float st, pr[3]

Pr[0] = (10/100)
Pr[1] = (10/100)
Pr[2] = (15/100)

input setup time

For i = 0; i < 3

\[ t[i] = st[i] + (st \times 60) \]
\[ coc[i] = (mir[i] \times t[i] / 3000) \times (i+pr[i]) \]

i++

For i = 0; i < 3

output total time
cost of component

i++
(Prog 4) (revenue earned per year)

```
int i, b = 0
float opt, wt

input working time /day, wt
no of working days, nwd

For i = 0 ; i < 3 i++

npm[i] = 2 * wt * 60 * 60 / tt[i]
re[i] = cco[i] * npm[i] * nwd
i++

For i = 0 ; i < 3

output no of pieces
machine / day, nmp[i]
revenue earned per year, re[i]
i++

OPT = re[0]

For i = 0 ; i < 3

if opt < re[i] yes
    opt = re[i]
b = i

i++

output optimum
```
Float x, err, y, cere
arr [3], pbp [3], at [3]
opt, k, w
int b,q

For i = 0 ; i < 3
arr [i] = re [i]/ce [i]
pbp [i] = 1/arr [i]

i ++

For i = 0 ; i < 3
output overage rate
of return for % 5 is % F %, type [i]
arr [i] x 100

i ++

output "return o investment method"

opt= arr [0] ;

For i = 0 i < 3

If opt < are [i]

yes

opt = arr [i]
b = i

No

i ++

output optimised

opt = pbp [0]

Output n optimisation by pay back period method
Float x, err, y, cere
arr [3], ppb [3], at [3]
opt, k, w
int b, q

For i = 0; i < 3
arr [i] = re [i]/ce [i]
ppb [i] = 1/arr [i]
i++

For i = 0; i < 3
output overage rate
of return for % 5 is % F %, type [i]
arr [i] x 100
i++

output "return o investment method"

opt = arr [0];

For i = 0; i < 3
If opt < are [i]

Yes
opt = arr [i]
b = i

No
i++

output optimised
opt = ppb [0]

Output n optimisation by pay back period method
I

output optimised

For i = 0 to <= 3

opt > pbp[i]

No

i ++

output optimised

For i = 0 to <= 3

core = (1.0 x ce[i]) / re[i] x = 1

output value of internal rate of return

For q = 1 to <= 50

Y = (x - (x + pow(x, 2) + pow(x, 3) + pow(x, 4) + pow(x, 5) - core) / (1.0 + (2 x) + 3 x Pow(x, 2) + 4 Pow(x, 3) + 5 x pow(x, 4)
err = Fabs (y-x)
x = y

output q, error, y

if err < ee.0.7

No

q ++

(output internal rate of return)

i ++

I

Output optimised

if err < 0.7

b = i

For i = 0 to <= 3

core = (1.0 x ce[i]) / re[i] x = 1

output value of internal rate of return

For q = 1 to <= 50

Y = (x - (x + pow(x, 2) + pow(x, 3) + pow(x, 4) + pow(x, 5) - core) / (1.0 + (2 x) + 3 x Pow(x, 2) + 4 Pow(x, 3) + 5 x pow(x, 4)
err = Fabs (y-x)
x = y

output q, error, y

if err < ee.0.7

No

q ++

(output internal rate of return)

i ++
For $i = 0 \ i < 3$

if opt < cr[i]

opt = r[i]

b = i

No

$i++$

output the optimum feed

output net present value method

input the value of rate of return

\[ W = \text{pow} ((i, i + k, 1.0)) \]

For $i = 0 \ i < 3$

\[ \text{npv}[i] = -1.0 * \text{co}[i] + \text{re}[i] * w + \text{re}[i] * \text{pow}(w, 2) + \text{re}[i] * \text{pow}(w, 3) + \text{re}[i] * \text{pow}(w, 4) + \text{re}[i] * \text{pow}(w, 5) \]

output the net present value npv[i]

$i++$

For $i = 0 \ i < 3$

if opt < cr[i]

opt = r[i]

b = i

No

$i++$

output the optimiced feed
output profitability index method

For $i = 0 \to 3$

$\text{Pri}_{[i]} = (\text{npv}_{[i]} + \text{ce}_{[i]}) / \text{ce}_{[i]}$;

output profitability index $\text{pri}_{[i]}

++

For $i = 0; 1 < 3$

if $\text{opt} < \text{pri}_{[i]}$

$\text{opt} = \text{pri}_{[i]}

\text{b} = i$

else

++

output optimised feed

... (Prog 6) (Break even point)...

Float opt, vc[3], vcc[3], as[3], tc[3], prtt[3], con[3], pwr[3], bep[3], bepr[3], mos[3]

int b
For $i = 0$ to $3$

\[
\begin{align*}
vc[i] &= dc[i] + cp[i] + cmh[i] \\
vcc[i] &= (vc[i] \times tt[i])/3600 \\
\text{as}[i] &= nprm[i] \times nwd \\
tc[i] &= vcc[i] \times \text{as}[i] + coh[i] + lw[i] \\
prtt[i] &= re[i] - tc[i] \\
con[i] &= coc[i] - vcc[i] \\
pvr[i] &= (con[i] / coc[i] \times 100 \\
bep[i] &= (con[i] + w[i] / con[i] \\
bepr[i] &= (coh[i] + cw[i]) / pvr[i] \\
mos[i] &= re[i] - bep[i]
\end{align*}
\]

$i++$

For $i = 0$ to $3$

output break even point

output: variable cost per hour, variable cost per component, annual sales, total cost, profit, contribution margin, volume ratio, break even point, break even point in RS, margin of safety

$i++$

opt = prtt[0]

For $i = 0$ to $3$

if opt < prtt[i] then

output optimised profit volume of feed
opt = pvr[0]

For i = 0, i < 3

if opt < pvr[i]
    opt = pvr[i]
    b = i
else
    i++

output optimised profit volume ratio

opt = bep[0]

For i = 0, i < 3

if opt < bep[3]
    opt = bep[i]
    b = i
else
    i++

output the optimised break even point

opt = bepr[0]

For i = 0, i < 3

if opt < bepr[i]
    opt = bepr[i]
    b = i
else
    i++

output the optimised break even point in Rs
Optimise output the optimum feed

i = 0, i < 3

if opt > mos [i]

b = i

i++

output the optimum feed

declare sub prog 7

int n [3] 1, 2

float opt

For i = 0, i < 3

input number of sample feed data

i++

For i = 0, i < 3
goto sub prog 7

i++

opt = std dev

For i = 0, i < 3

if opt < std dev [i]
yes

opt = std dev [i]
z = i

No

i++
output the optimum feed mechanism

For i = 0 to 3

\[ pci[i] = (Fi[i][o] - Fi[i][n[i]]) / pc[i] \]

- if \( pci[i] < 1 \)
  - Yes: output process is not capable
  - No
- if \( pci[i] = 1 \)
  - Yes: output process is just capable
  - No
- if \( pci[i] > 1 \)
  - Yes: output process is very much capable
  - No

i++

J

int

Float F = 0, F = 0, F = 0, x = 0, y

For j = 0, j < 3

input value, Fi[a][i]

j++
For $j = 0, j < 3$

Input $x_i [a][j]$

$j++$

For $i = 0, i < b$

$X = F_i [a][j]$  
$Y = X_i [a][j]$  
$F_{x+} = x^*y$  
$F = x$  
$F_{xx+} = x^*y^*y$

$i++$

$X_{bar} (a) = (Fx/F)$  
$St\ dev [a] = Pow ((F_{xx-b} * Pow x_{bar} [a], 2)) / b, 0.5.)$  
$Var [a] = pow (stav [a], 2)$  
$Pc [a] = 6 * stdev [a]$
10.6 COMPUTER PROGRAMME (USING "C" LANGUAGE) TO PREDICT OPTIMUM FEED MECHANISM OF FINE BORING MACHINE

```c
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<cstring.h>
#include<graphics.h>
#include<process.h>
#include<dos.h>

/* global variable declaration */

int i,nwd;
float ce[3],lh,wh,mh,sv[3],crm[3],cog[3],cw[3],p[3],cc;
float ct[3],mhr[3],tt[3],coc[3],re[3];
float awh,fe[3],crmh[3],dc[3],cp[3],npm[3];
float fi[3][10],xi[3][10],xbar[3],stdev[3],var[3],pc[3],pr[3];
float r[3],npv[3],pri[3];
float pi=(22/7);
char type[3][201];

void screen(void);

void processing_txt()
{
  gotoxy(10,10);

  textcolor(9);
  cprintf("\n\rPROCESSING......\n\r");

  char msg[30];

  int q;
  for(int i=0;i<=5;i++)
  {
    cprintf("000");
    delay(20);
    q=wherex();
    gotoxy(34,wherey()-1);
    sprintf(msg,"%i% Complete",i*5);
    cprintf(msg);
    gotoxy(q,wherey()+1);
  }
  delay(300);
  cprintf("00000000");
  for(i=6;i<=10;i++)
  {
    cprintf("000");
    delay(300);
    q=wherex();
    gotoxy(34,wherey()-1);
    sprintf(msg,"%i% Complete",i*5);
    cprintf(msg);
    gotoxy(q,wherey()-1);
  }
  delay(50);
  cprintf("000000");
  for(i=10;i<=20;i++)
```

{cprintf("000");
  delay(25);
  q=wherex();
  gotoxy(34,wherey()-1);
  sprintf(msg,"%i Complete",i*5);
  cprintf(msg);
  gotoxy(q,wherey()+1);
}

void processing_grp()
{
  settextstyle(DEFAULT_FONT,0,1);
  setcolor(3);
  moveto(0,380);
  outtext("PROCESSING.....");
  int w=getx();
  int e=gety();
  char msg[30];
  moveto(0,400);
  int q,r;
  for(int i=0;i<=5;i++)
  {
    outtext("000");
    delay(200);
    q=getx();
    r=gety();
    moveto(w,e);
    setcolor(0);
    outtext("00000000000000000000000000000000");
    moveto(w,e);
    setcolor(3);
    sprintf(msg,"%i Complete",i*5);
    outtext(msg);
    moveto(q,r);
  }
  delay(300);
  outtext("0000000");

  for(i=6;i<=10;i++)
  {
    outtext("000");
    delay(300);
    q=getx();
    r=gety();
    moveto(w,e);
    setcolor(0);
    outtext("00000000000000000000000000000000");
    moveto(w,e);
    setcolor(3);
    sprintf(msg,"%i Complete",i*5);
    outtext(msg);
    moveto(q,r);
  }
  delay(500);
outtext("00000");

for(i=11;i<=20;i++)
{
    outtext("000");
    delay(500);
    q=getx();
    r=gety();
    moveto(w,e);
    setcolor(0);
    outtext("000000000000000000000000");
    moveto(w,e);
    setcolor(3);
    sprintf(msg, "%i Complete", i);
    outtext(msg);
    moveto(q,r);
}

void main()
{
    int gdriver = DETECT, gmode, errorcode;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    /* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: \n", grapherrormsg(errorcode));
        printf("Press any key to halt:"):
        getch();

        exit(1); /* terminate with an error code */
    }

    setcolor(2);
    settextstyle(10,0,1);
    outtextxy(180,50,"OPTIMISATION OF");
    outtextxy(153,100,"FINE BORING MACHINE");
    setcolor(4);
    settextstyle(8,0,1);
    outtextxy(300,220,"BY R.HENRY XAVIER");

    processing_grp();
    closegraph();

    textcolor(2);
    /* MAIN PROGRAM */

    void progl(void);
    void prog2(void);
    void prog3(void);
void prog4(void);
void prog5(void);
void prog6(void);
void prog7(void);
int n=0;
strcpy(type[0],"HYDRAULIC");
strcpy(type[1],"ELECTRO MECHANICAL");
strcpy(type[2],"CNC");

/* program selection menu */

while(n!=8)
{
    textcolor(6);
cprintf("\n\r\n\rSEL.ECTIC)N\n\r\rMENCI");
textcolor(2);
cprintf("\n\r\n\r1 - CYCLE TIME\r\r 2 - MACHINING HOUR RATE\r\r 3 - COST OF COMPONENT\r\r 4 - REVENUE EARNED PER YEAR\r\r 5 - RETURN ON INVESTMENT\r\r 6 - BREAK EVEN POINT\r\r 7 - OPTIMIZATION BY QUALITY ANALYSIS\r\r 8 - EXIT PROGRAM\r\n\r\rENTER THE CHOICE : ");
scanf("%d",&n);
clrscr();
switch(n)
{
    case 1 : prog1();
        break;
    case 2 : prog2();
        break;
    case 3 : prog3();
        break;
    case 4 : prog4();
        break;
    case 5 : prog5();
        break;
    case 6 : prog6();
        break;
    case 7 : prog7();
        break;
}
}

/* program to find and display the cycle time */

void prog1()
{
    int i,b;
    float d,cs,n,fr,bl,rt,rr[3],mtr,tru[3],feed,opt;
    textcolor(6):
cprintf("CYCLE TIME");
textcolor(2);
cprintf("Enter the Diameter of the Bore");
scanf("%f", &d);
cprintf("Enter the Bore Length");
scanf("%f", &bl);
cprintf("Enter the Feed Rate");
scanf("%f", &fr);
cprintf("Enter the Rapid Travel");
scanf("%f", &rt);
for (i = 0; i < 3; i++)
{
cprintf("Enter the Rapid Rate for %s", type[i]);
scanf("%f", &rr[i]);
}
cprintf("Enter the cutting speed");
scanf("%f", &cs);
cprintf("Machining Tool Program");
clrscr();

/* CYCLE TIME */

n = (1000 * cs) / (pi * d);
feed = n * fr;
mt = (bl * 60) / feed;

for (i = 0; i < 3; i++)
{
    try[i] = rr[i] * 60 / rr[i];
    ri[i] = 4 * try[i] + 2 * mt;
}

/* TO FIND THE OPTIMUM CYCLE TIME */

opt = ct[0];
b = 0;
for (i = 0; i < 3; i++)
    if (opt > ct[i])
    {
        opt = ct[i];
        b = i;
    }

/* TO DISPLAY THE CALCULATED RESULTS */

cprintf("The list of the values obtained are");
for (i = 0; i < 3; i++)
{
    cprintf("%s", type[i]);
    cprintf("Cutting Speed = %0.4f", cs);
    cprintf("Feed per minute = %0.4f", feed);
    cprintf("Machining time = %0.4f", mt);
    cprintf("Time for Rapid Travel = %0.4f", try[i]);
}

cprintf(" CYCLE TIME = \%0.4f",ct[i])

screen();
)

cprintf(" THE OPTIMUM CYCLE TIME IS THAT OF %s ",type[b]);

screen();
)

void prog2()
{
 int log;
 /* sub program to calculate the machining hour rate */
 clrscr();

void subprog2(int);
void subdisp2(int);

/* program for entering the data */
for(i=0;i<3;i++)
{
 log=1;
 while(log==1)
 {
 subprog2(i);
 screen();
 cprintf(" Enter 'l' to Edit previous input data");
 cprintf(" Enter '0' to Proceed ");
 cprintf(" Enter your choice : ");
 scanf("%d",&log),
 clrscr();
 }
}

/* program for calculating the results */
for(i=0;i<3;i++)
{
 awh=wh-mh;
 fe[i]=((coh[i]+cw[i])/ahw);
 crmh[i]=(crma[i]/ahw);
 dc[i]=(ce[i]-sv[i])/lh;
 cp[i]=(p[i]*0.736*ec*ct[i])/3600;
 mhr[i]=fe[i]+dc[i]+crmh[i]+cp[i];
}

/* program for displaying the results */
clrscr();

for(i=0;i<3;i++)
{
 subdisp2(i);
 Screen();
}
void subprog2(int j)
{
    cprintf("\n\r Input Form of Machine\n\r Enter The Cost of the Equipment:");
    scanf("%f", &ce[j]);
    cprintf("\n\r Enter the Life in Hours:");
    scanf("%f", &lh);
    cprintf("\n\r Enter the Working Hours Per Annum:");
    scanf("%f", &wh);
    cprintf("\n\r Enter the Scrap Value:");
    scanf("%f", &sv[j]);
    cprintf("\n\r Enter the Maintenance hours:");
    scanf("%f", &mhr[j]);
    cprintf("\n\r Enter the Maintenance hours:");
    scanf("%f", &mhr[j]);
    cprintf("\n\r Enter the Electrical Charges Per Unit:");
    scanf("%f", &ec);
}

void subdisp2(int j)
{
    cprintf("\n\r Actual Working Hours:");
    scanf("%f", &awd[j]);
    cprintf("\n\r Fixed Expenses Per Hour:");
    scanf("%f", &fe[j]);
    cprintf("\n\r Cost of Repair and Maintenance:");
    scanf("%f", &crma[j]);
    cprintf("\n\r Cost of Overhead:");
    scanf("%f", &coh[j]);
    cprintf("\n\r Enter Motor Power in HP:");
    scanf("%f", &p[j]);
    cprintf("\n\r Enter the Electricity Charges Per Unit:");
    scanf("%f", &ec);
}

void prog3()
{
    int i;
    float st,pr[3];
    pr[0]=(10.0/100);
    pr[1]=(10.0/100);
    pr[2]=(15.0/100);
    cprintf("\n\r Setup Time in minutes:");
    scanf("%f", &ct);
    for(i=0;i<3;i++)
    {
        tt[i]=ct[i]+(st*60);
```c
float ppm[3];

void prog4()
{
    clrscr();
    int i, b=0;
    float opt, wt;
    textcolor(6);
    cprintf("REVENUE EARNED PER YEAR *: ");
    textcolor(2);
    cprintf("NO. OF PIECES MACHINED/DAY = %0.4f ", npm[i]);
    cprintf("REVENUE EARNED PER YEAR = %0.4f ", re[i]);
}

/* TO FIND THE OPTIMUM FEED MECHANISM */
opt=re[0];
for(i=0; i<3; i++)
    if(opt<re[i])
    { opt=re[i];
        b=i;
    }

screen();
```
void prog5() {
    float x, err, y, cere;
    float arr[3], pbp[3];
    float opt, k, w;
    int b, q;

    for (i = 0; i < 3; i++) {
        arr[i] = (re[i] / ce[i]);
        pbp[i] = (1 / arr[i]);
    }
    screen();

    for (i = 0; i < 3; i++)
        cprintf("Please enter the type of feed: ");

    cprintf("AVERAGE RATE OF RETURN FOR \$n IS \$f \n", type[i], arr[i] * 100);
    screen();

    /* to optimize the results */
    /* optimization by RETURN OF INVESTMENT */

    opt = arr[0];
    for (i = 0; i < 3; i++)
        if (opt < arr[i])
            { opt = arr[i];
              b = i;
            }
    cprintf("The optimized feed is \$s ", typelb);

    /* optimization by PAY BACK PERIOD METHOD */

    opt = pbp[0];
    for (i = 0; i < 3; i++)
        if (opt > pbp[i])
            { opt = pbp[i];
              b = i;
            }
    cprintf("The optimized feed is \$s ", typelb);
    screen();

    /* INTERNAL RATE OF RETURN METHOD */
for(i=0;i<3;i++)
{
    cere=(1.0*ce[i])/re[i];
x=1;

    /* finding the unknown variable by NEWTON-RAPHSONS METHOD */
    processing_txt();
clearscr();

cprintf("\n\r The value of Internal Rate of Return for \$", type[i]);
cprintf("\n\r Internal Rate of Return for \$", type[i]);
    for(q=1;q<=50;q++)
    {
        y=x-(x+pow(x,2)+pow(x,3)+pow(x,4)+pow(x,5)+pow(x,6));
cere)/(1.0+2*x+3*pow(x,2)+4*pow(x,3)+5*pow(x,4));
        err=fabs(y-x);
        x=y;
        cprintf("\n\r $d Error= %12.5e Value= %d$, err,y);
        if(err<1.0e-07) break;
    }
r[i]=(1.0-x)/x;

cprintf("\n\r INTERNAL RATE OF RETURN FOR \$ is \$0.6f \%
", type[i], r[i]*100);
    screen();
}
/* TO FIND THE OPTIMUM INTERNAL RATE OF RETURN */
for(i=0;i<3;i++)
    if(opt<r[i])
    {
        opc=r[i];
        b=i;
    }
cprintf("\n\r The Optimized Feed is \$ \$", type[b]);

/* NET PRESET VALUE METHOD */

screen();
textcolor(6);
cprintf("\n\r NET PRESET VALUE METHOD ");
textcolor(2);
cprintf("\n\r Enter the value of Rate of Return ");
    scanf("%f",&k);
    screen();
    w=pow((1+k),-1.0);
    for(i=0;i<3;i++)
    {
        npv[i]=
        1.0*ce[i]+re[i]*w+re[i]*pow(w,2)+re[i]*pow(w,3)+re[i]*pow(w,4)+re[i]*pow(w,5);
        cprintf("\n\r THE NET PRESET VALUE FOR \$ is \$0.6f", type[i], npv[i]);
    }
screen();
/** TO FIND THE OPTIMUM NET PRESET VALUE */

for(i=0;i<3;i++)
    if(opt<r[i])
        { opt=r[i];
        b=i;
    }
cprintf("\n|\n|\n|\n|\nThe Optimized Feed is %s ",type[b]);
screen();

/** PROFITABILITY INDEX */

cprintf("\n|\n|\n|\n|\nThe PROFITABILITY INDEX for %s is %0.4f ",type[i],pri[i]);
}

for(i=0;i<3;i++)
    if(opt<pri[i])
        { opt=pri[i];
        b=i;
    }
cprintf("\n|\n|\n|\n|\nThe Optimized Feed is %s ",type[b]);
screen();
}

/** TO FIND THE BREAK EVEN POINT */

void prog6()
{
    float opt,vc[3],vcc[3],as[3],tc[3],prft[3];
    float con[3],pvr[3],bep[3],bepr[3],mos[3];
    int b;
    /* break even point */

    screen();
    for(i=0;i<3;i++)
    {
        vc[i]=dc[i]+cp[i]+crmh[i];
        vcc[i]=(vc[i]*tt[i])/3600.0;
        as[i]=npm[i]*nwd;
        tc[i]=(vcc[i]*as[i]*1.0)+coh[i]+cw[i];
        prft[i]=re[i]-tc[i];
        con[i]=coc[i]-vcc[i];
        pvr[i]=(con[i]/coc[i])*100.0;
        bep[i]=coh[i]+cw[i]/con[i];
        bepr[i]=(coh[i]+cw[i])/(pvr[i]/100.0);
        mos[i]=re[i]-bepr[i];
    }
    /* to display the various values obtained */
for(i=0;i<3;i++)
{
    clrscr();
textcolor(5);
cprintf("\n\r
\n\r Variable Cost per Hour  \$0.4f * \vc[i]);
cprintf("\n\r Variable Cost per Component  \$0.4f * \vcc[i]);
cprintf("\n\r Annual Sales  \$0.4f * \as[i]);
cprintf("\n\r Total Cost  \$0.4f * \tc[i]);
cprintf("\n\r Profit  \$0.4f * \prft[i]);
cprintf("\n\r Contribution  \$0.4f * \con[i]);
cprintf("\n\r Profit Volume Ratio  \$0.4f * \pvr[i]);
cprintf("\n\r Break Even Point  \$0.4f * \bep[i]);
cprintf("\n\r Break Even Point in Rupees  \$0.4f * \bepr[i]);
cprintf("\n\r Margin of Safety  \$0.4f * \mos[i]);
}

 /* OPTIMIZED PROFIT VALUE */

opt=prft[0];
for(i=0;i<3;i++)
    if(opt>prft[i])
    {
        opt=prft[i];
        b=i;
    }
textcolor(6);
cprintf("\n\r OPTIMIZED PROFIT VALUE ");
textcolor(2);
cprintf("\n\r\n\n\r The optimum feed type is \$s * .type[b]);
}

 /* OPTIMIZED BREAK EVEN POINT */

opt=bep[0];
for(i=0;i<3;i++)
    if(opt>bep[i])
    {
        opt=bep[i];
        b=i;
    }
textcolor(6);
cprintf("\n\r OPTIMIZED BREAK EVEN POINT ");
textcolor(2);
cprintf("\n\r\n\n\r The optimum feed type is \$s * .type[b]);
}

 /* OPTIMIZED BREAK EVEN POINT IN RUPEES */

opt=bepr[0];
for(i=0;i<3;i++)
    if(opt>bepr[i])
    {
        opt=bepr[i];
b=i;
}
textcolor(6);
cprintf("\n\r OPTIMIZED BREAK EVEN POINT IN RUPEES");
textcolor(2);
cprintf("\n\r\n\r The optimum feed type is %s ",type[b]);
screen();

/* OPTIMIZED MARGIN OF SAFETY */

opt=mos[0];
for(i=0;i<3;i++)
if(opt<mos[i])
{
    opt=mos[i];
    b=i;
}
textcolor(6);
cprintf("\n\r OPTIMIZED MARGIN OF SAFETY ");
textcolor(2);
cprintf("\n\r\n\r The optimum feed type is %s ",type[b]);
screen();

/* OPTIMIZATION BY QUALITY */

void prog7()
{
    clrscr();
    void subprog7(int,int);
    int n[3],i,z;
    float opt;

    for(i=0;i<3;i++)
    { cprintf("\n\r Number of Sample Feed Data for %s ",type[i]);
        scanf("%d","n[i]"));
    }
    for(i=0;i<3;i++)
    { subprog7(i,n[i]);
    }
    /* to find the optimum feed mechanism */
    opt=stdev[0];
    for(i=0;i<3;i++)
        if(opt>stdev[i])
            if(opt>stdev[i])
            { opt=stdev[i];
                z=i;
            }
textcolor(6);
cprintf("\n\r\n\r THE OPTIMUM FEED MECHANISM IS %s ",type[z]);
textcolor(2);
screen();

    /* to find the process capability index */
    for(i=0;i<3;i++)
    { pci[i]=(fi[i][0]-fi[i][n[i]])/pc[i];
    }
if(pci[i]<1)
cprintf("\r %s PROCESS IS NOT CAPABLE",type[i]);
else
  if(pci[i]==1)
cprintf("\r %s PROCESS IS JUST CAPABLE",type[i]);
  else
    if(pci[i]>1)
cprintf("\r %s PROCESS IS HIGHLY CAPABLE",type[i]);
}
screen();

/* OPTIMIZATION BY QUALITY ANALYSIS */

void subprog7(int a, int b)
{
iclrscre();
  int j;
  float fx=0, f=0, fxx=0, x=0, y=0;
  cprintf("\n\r %s ",type[a]);
  cprintf("\n\n\n\r
\r");
  for(j=0; j<b; j++)
  { cprintf("\n\r Enter the value f(%d) = ",j);
    scanf("%f",&fi[a][j]);
  }
  screen();
  cprintf("\n\r %s ",type[a]);
  for(j=0; j<b; j++)
  { cprintf("\n\r Enter the value of x(%d) = ",j);
    scanf("%f",&xi[a][j]);
  }
  for(j=0; j<b; j++)
  { x=fi[a][j];
    y=xi[a][j];
    fx+=x*y;
    f+=x;
    fxx+=x*y*y;
  }
  xbar[a]=(fx/f);
  stdev[a]=pow((fxx-b*pow(xbar[a],2))/b,0.5);
  var[a]=pow(stdev[a],2);
  pc[a]=6*stdev[a];
}

void screen(void)
{
  gotoxy(20,25);
cprintf("press any key to cont... ");
getch();
cilrscre();
}