CHAPTER II

Economic Design of np-Control Charts

2.1 In this chapter two expected cost models are developed for the optimum economic design of np-control charts. The basis for the construction of these models is discussed in section 2.2. The cost model developed in section 2.3 treats the case where a single assignable cause of known effect occurs randomly. However, a single assignable cause model is inappropriate when the production process is affected by the several assignable causes. In section 2.4 we propose an expected cost model which is designed in such a way that it incorporates the occurrences of the several assignable causes.

2.2 The Basis for the Construction of the Expected Cost Models

2.2.1 The Main Features of the Existing Models

Duncan (1956) proposed an economic model for controlling a production process where a single assignable cause of known effect occurs randomly. His paper was the first to deal with a fully economic model of a Shewhart-type control chart and to incorporate the optimization methodology to determine the control chart parameters. Duncan's paper was the stimulus for much of the subsequent research work done in this area.

Later on two distinctly different economic models have been developed for controlling a production process subject to multiple assignable causes-one by Duncan (1971) himself and the other by Knappenberger and Grandage (1969). The Duncan's (1971) multiple assignable cause model is a generalisation of his (1956) single assignable cause model. The Knappenberger and Grandage's (1969) model differs considerably from the Duncan's (1971) model.

The Duncan's (1971) model and the Knappenberger and Grandage's (1969) model have different objective functions. The Duncan's model minimizes the expected cost per unit of time during a production cycle. A production cycle is the average length of the time spent between two successive in-control states after detection and the elimination of the assignable cause. The Knappenberger and Grandage's model minimizes the expected cost per unit produced between two successive samples. It is more realistic to minimize the expected cost of controlling the process between two successive in-control states because an entry of the process into an in-control state is a regeneration of controlling the process. Hence the Duncan's model seems to have more realistic cost structure than the Knappenberger and Grandage's model from this point of view.

The Duncan's model assumes that once the process shifts to an out-of-control state, it remains in that state without further quality deterioration, until the shift is detected by the control chart. The Knappenberger and Grandage's model allows continuous deterioration of quality beyond the initial shift, which is a more realistic feature of the behavior of the production process than the Duncan's model.

The above description gives the major drawbacks of both the models and the good points of one over the other. These and the other drawbacks of both the models are listed systematically in the next few lines.

The drawbacks and the unrealistic assumptions of the Knappenberger and Grandage's (1969) model are as follows :-

(1) The model uses the time independent, steady state probabilities in place of the true process state probabilities.

(2) The model assumes the same cost of producing a nonconforming unit whether it is detected during sampling or it goes undetected to the customer.

(3) The model assumes the same cost of searching for a false alarm and of searching for a true alarm and repairing the process.

(4) The cost structure of the model is such that the expected cost of controlling the process between two successive samples is minimized.

The drawbacks and the unrealistic assumptions involved in the Duncan's (1956) model are as follows :-

(5) The model assumes that the production continues during the search for an assignable cause.

(6) The model does not include the time and the cost of repairing the process if it is found to be out of control.

The Duncan's (1971) multiple assignable cause model involves two more unrealistic assumptions in addition to (5) and (6) which are listed as (7) and (8) below.

(7) Once the process is out of control no further quality deterioration can occur until the shift in the process is detected by the chart.

(8) The assignable causes are assumed to occur independently.

2.2.2 The Main Features of the Models developed by us

The economic models developed in this chapter do not involve any of the unrealistic assumptions and drawbacks as listed (1) through (8). Also they take care of the good points of both the models mentioned above. Hence the models under our study are likely to be more realistic and hence more applicable. The improvements in our models are listed as (1') through (8') and they have one to one correspondence with the drawbacks as listed (1) through (8).

(1') The model uses the true process state probabilities and not the steady state probabilities.

(2') A higher cost is attached to a nonconforming unit that goes undetected to the customer than to the one which is detected during sampling.

(3') The cost of searching for a false alarm and the cost of searching for a true alarm and repairing the process are different.

(4') The model computes the expected cost per unit produced between two successive in-control states.

(5') The production may or may not be continued during the search for an assignable cause.

(6') The cost of repairing the process is taken into consideration.

(7') The model allows that once the process is out of control it may further deteriorate before the shift in the process is detected by the chart.

(8') The model does not assume that the assignable causes occur independently.

It may be mentioned that Chiu (1975) developed the economic model for the np-control chart using the Duncan's (1956) single assignable cause model for $\bar{X}$-control chart. He, furthermore, developed in (1976) the economic model for np-control chart using the Duncan's (1971) multiple assignable cause model for $\bar{X}$-control chart. Chiu has taken care of the unrealistic assumptions (5) and (6) mentioned above and has improved his models accordingly. However, Montgomery, Heikes and Mance (1975), while developing the economic model for np-control chart using the Knappenberger and Grandage's (1969) model for $\bar{X}$-chart, have not done any improvement in the Knappenberger and Grandage's model.

In the rest of the chapter, we develop our models for np-control chart incorporating the improvements (1') through (8') listed above. Section 2.3 is devoted to single assignable cause whereas in Section 2.4 it is assumed that the production process is affected by several assignable causes.

2.3 The Single Assignable Cause Model

2.3.1 The Production Process and the Sampling Scheme

The production process starts in the in-control state in which it produces a known acceptable proportion p_0 of the nonconforming units. There exists a single assignable cause of variation which has the effect of increasing the proportion of nonconforming units to p_1. The assignable cause occurs at a rate λ per hour of operating time and the operating time until its occurrence is assumed to be an exponential random variable. Hence the production process remains in the in-control state for an exponential duration with mean $1/\lambda$ hours of operating time. At the end of this exponential duration it moves to an out-of-control state in which it produces a higher proportion p_1 of nonconforming units. Thus, there are only two states p_0 and p_1 of the production process.

The sampling scheme is as follows. After every h hours of operating time, n units are sampled and examined. Let R be the number of units produced per hour of operating time. Hence the number of units produced between two successive samples is hR(=k say). Let d be the number of nonconforming units detected in the sample. If $d < m$ the process is declared to be in the in-control state p_0 and the production continues. If $d \geq m$ the process is declared to be in the out-of-control state p_1. The production at this stage may or may not be stopped and a search for the assignable cause is undertaken. If the assignable cause exists, the process is repaired and restored to the in-control state p_0.

We want to find the optimal values of the design variables n, m, k which minimize the expected cost per unit of the product during the production cycle.

2.3.2 The Expected Number of Samples taken during the Production Cycle

A production cycle is defined to be the time period from the beginning of the production process to the detection and elimination of the assignable cause. We find an expression for the expected number of samples taken during the production cycle.

Let Z be the number of samples required to detect the shift in the process. Then Z is a random variable taking the values 1, 2, 3, ...

The probability of detecting the shift on the basis of zth sample is

$$p(z) = \sum_{i=1}^{z} p_i(z) \qquad \ldots(2.3.1)$$

where $$p_i(z) = \theta^{i-1}(1-\theta)\ (1-q_1)^{z-i}q_1, \qquad \ldots(2.3.2)$$

$$\theta = \exp(-\lambda k/R), \qquad \ldots(2.3.3)$$

$$q_1 = \sum_{d=m}^{n} \binom{n}{d} p_1^d(1-p_1)^{n-d}. \qquad \ldots(2.3.4)$$

The expression on the R.H.S. of (2.3.2) is the product of the following three probabilities :-

(i) The probability that the shift occurs during the production of (i-1)k to ik units

$$= \int_{(i-1)k/R}^{ik/R} \lambda \exp(-\lambda t)dt$$

$$= \theta^{i-1}(1-\theta),$$

(ii) The probability that the shift is not detected on the basis of ith, (i+1)th, ..., (z-1)th samples

$= (1-q_1)^{z-i}$,

(iii) The probability that the shift is detected on the basis of zth sample

$= q_1$.

It can be verified that

$$\sum_{z=1}^{\infty} p(z) = 1 \qquad \text{...(2.3.5)}$$

and that

$$E(Z) = \frac{\theta}{1-\theta} + \frac{1}{q_1} \qquad \text{...(2.3.6)}$$

Furthermore, it can be shown that the above break-up of the expectation is more meaningful by showing that it is the sum of expectations of two random variables X and Y where

X = the number of samples taken when the process is in the in-control state p_0, and

Y = the number of samples required to detect the shift given that the shift has occured.

Here Z = X + Y and X takes the values 0,1,2, ..., and Y takes the values 1,2,3, ... We derive the expressions for E(X) and E(Y) in the next few lines.

Now, P(X=r) = P(r samples are taken in the in-control state)

= P(the shift occurs between rth sample and r+1th sample)

= P(the shift occurs during the production of rk/R units to (r+1)k/R units)

$$= \int_{rk/R}^{(r+1)k/R} \lambda e^{-\lambda t}\, dt$$

$$= \theta^r (1-\theta)$$

where expression for θ is given by (2.3.3).

Hence $E(X) = \sum_{r=0}^{\infty} rP(X=r) = \theta/(1-\theta)$.

Obviously $E(Y) = 1/q_1$, since $P(Y=r) = (1-q_1)^{r-1}q_1$.

Sum of these expectations establishes the fact stated.

2.3.3 The Expected Cost Model

We compute the total expected cost per unit for controlling the process during the production cycle. The total cost C consists of three components C_1, C_2, C_3

where

C_1 = the cost of sampling and inspection,

C_2 = the cost of finding the assignable cause and repairing the process when the sampling policy generates an out of control signal,

C_3 = the cost of producing nonconforming units.

It may be noted that the cost components C_1, C_2, C_3 are similar to those described by Knappenberger and Grandage (1969). Furthermore, in order to calculate E(C) we derive in turn the expressions for $E(C_1)$, $E(C_2)$ and $E(C_3)$.

(a) The expected cost of sampling and inspection is

$$E(C_1) = (a_1+a_2n)N \qquad \ldots(2.3.7)$$

where

a_1 = the fixed cost of sampling,

a_2 = the variable cost per unit of sampling,

n = the sample size,

N = the expected number of samples taken during the production cycle.

Here N represents E(Z) whose expression is given by (2.3.6). This is also useful in computing the denominator in the total expected cost per unit.

(b) The expected number of samples taken during the in-control state is denoted by N(0) and as derived in Section 2.3.2, its expression is

$$N(0) = \theta/(1-\theta). \qquad \ldots(2.3.8)$$

Hence the expected number of false alarms during the production cycle is

$$B_o = q_o\ \theta/(1-\theta) \qquad \ldots(2.3.9)$$

where

$$q_o = \sum_{d=m}^{n} \binom{n}{d} p_o^d (1-p_o)^{n-d} \qquad \ldots(2.3.10)$$

Thus the expected cost of finding the assignable cause and repairing the process is

$$E(C_2) = a_{3,1}B_o + a_{3,2}1 \qquad \ldots(2.3.11)$$

where

$a_{3,1}$ = the cost of searching for a false alarm,

$a_{3,2}$ = the cost of searching for a true alarm and repairing the process.

(c) Let $e_i(z)$ be the event that the shift occurs during the production of $(i-1)k$ to ik units, not detected on the basis of ith, (i+1)th, ..., (z-1)th samples, detected on the basis of zth sample ($i = 1,2,...z$ and $z = 1,2,3,....$). The probability of the event $e_i(z)$ is given by $p_i(z)$. The expression for $p_i(z)$ is given by (2.3.2).

Under the event $e_i(z)$, first (i-1) samples are taken in the in-control state p_0 and the remaining $z-(i-1)$ samples are taken in the out-of-control state p_1.

Hence the number of nonconforming units produced during the situation $e_i(z)$ is

$$D_i(z) = \{ (i-1)k + \Delta k \}p_0 + \{ (z-i+1)k - \Delta k \}p_1$$

$$(i = 1, 2, ... ,z \text{ and } z = 1, 2, ...) \qquad ...(2.3.12)$$

where Δ is the average fraction of the time the process remains in the in-control state before shifting to the out-of-control state, given that the shift occurs between two successive samples. Duncan (1956) has shown that,

$$\Delta = \frac{1 - (1 + \lambda k/R)\Theta}{(1-\Theta)\lambda k/R} \qquad ...(2.3.13)$$

Multiplying $D_i(z)$ by their respective probabilities $p_i(z)$ the expected number of nonconforming units produced during the production cycle is

$$D = \sum_{z=1}^{\infty} \sum_{i=1}^{z} D_i(z)\ p_i(z) \qquad ...(2.3.14)$$

$$= \left[\frac{k\theta}{(1-\theta)} + \Delta k \right] p_o + \left[\frac{k}{q_1} - \Delta k \right] p_1 \qquad \ldots(2.3.15)$$

Similarly the number of nonconforming units detected during sampling under the same event $e_i(z)$ is given by

$$S_i(z) = (i-1)np_o + (z-i+1)np_1 \qquad \ldots(2.3.16)$$

$$(i = 1, 2, \ldots z \text{ and } z = 1, 2, \ldots)$$

Multiplying $S_i(z)$ by their respective probabilities $p_i(z)$, the expected number of nonconforming units detected during sampling is given by

$$S = \sum_{z=1}^{\infty} \sum_{i=1}^{z} S_i(z)\ p_i(z) \qquad \ldots(2.3.17)$$

$$= \frac{np_o\theta}{1-\theta} + \frac{np_1}{q_1} \qquad \ldots(2.3.18)$$

Hence the expected cost of producing nonconforming units during the production cycle is

$$E(C_3) = a_{4,1}S + a_{4,2}(D - S) \qquad \ldots(2.3.19)$$

where

$a_{4,1}$ = the cost per nonconforming unit which is detected during sampling,

$a_{4,2}$ = the cost per nonconforming unit which goes undetected to the customer.

In practice, it may be the fact that $a_{4,2} > a_{4,1}$.

Combining the expressions (2.3.7), (2.3.11) and (2.3.19) one can now calculate the expected total cost incurred during the production cycle as

$$E(C) = E(C_1) + E(C_2) + E(C_3) \quad \ldots(2.3.20)$$

and the expected total cost per unit of the product during the production cycle as

$$ECPU = \frac{E(C)}{Nk} \quad \ldots(2.3.21)$$

where Nk is the expected number of units produced during the production cycle which is k times the expression given in (2.3.6).

The optimal design variables n, m, k of this model are then obtained by minimizing the expression (2.3.21).

2.3.4 Direct Search Method

Hooke-Jeeves' (1961) search procedure given in a book by Kuester and Mize (1973) is used to find the optimal values of the design variables which minimize a given objective function.

The original function minimized by Hooke-Jeeves' procedure is a function of continuous variables. The details of the method of Hooke-Jeeves' procedure are reproduced at the end of this chapter. However, in this section we mention the essential appropriate changes.

All the three design variables of the objective function developed in this chapter are discrete. Hence Hooke-Jeeves' procedure is to be used with appropriate care. Some limitation are required on the step size (EPS) and on the factor (β) that reduces the initial step size. This is done by giving initially some integer values (n_o, m_o, k_o) and by choosing EPS=1 and β=1. Thereby Hooke-Jeeves' procedure calculates the objective function at various discrete values of the design variables and ultimately finds the optimal solution in the discrete form. To avoid the possibility of the local minimization, one has to apply Hooke-Jeeves' method repeatedly giving different initial values, and find that solution which is the best among all the optimal solutions given by Hooke-Jeeves' method.

2.3.5 A Numerical Example

Let a_1 = \$10, a_2 = \$1, $a_{3,1}$ = \$100, $a_{3,2}$ = \$100, $a_{4,1}$ = \$10, $a_{4,2}$ = \$15.

Let λ = 1, R = 1000, p_o = 0.01, p_1 = 0.10.

For this combination of the cost coefficients and the systems parameters, the search technique described in the previous section is used.

We have developed a computer program on FORTRAN to calculate the expected cost per unit of the product given by the expression (2.3.21) for the given values (n_o, m_o, k_o). This program uses a subroutine for the calculation of the cumulative binomial probabilities. This program is then linked to Hooke-Jeeves' procedure to derive the optimal solution and to calculate the minimum ECPU.

The listing associated with all the programs developed are given at the end of this chapter.

The search technique yielded the following optimal values.

n = 37, m = 2, k = 350 with minimum ECPU = \$ 0.5457.

The values of some intermediate terms calculated for this numerical example at the optimal stage are given below.

$N(0) = 3$, $N = 4$.

$q_0 = 0.0528$, $q_1 = 0.8964$.

$D = 33$, $S = 5$.

$E(C_1) = \$ 187.99$, $E(C_2) = \$ 112.62$, $E(C_3) = \$ 463.41$,

$E(C) = \$ 764.02$

Interpretation :- From the above numerical output for the stated configuration of constants of the system, one can have the following interpretation for the optimal economic design of the np-control chart.

After the production of every 350 units, take a sample of first 37 units for inspection and hunt for the assignable cause if 2 or more nonconforming units are obsreved. The process may be continued or may be discontinued when the search for the assignable cause is undertaken. When 0 or 1 nonconforming units are observed in the inspection of 37 units the process is be continued. The resulting minimum expected total cost per unit during the production cycle is \$ 0.5457. The probability of decting the shift (from 1% nonconforming units to 10% nonconforming units) is 0.8964 and the probability of the false alarm is 0.0528. The expected number of samples taken during the production cycle is 4. Among these 4 samples 3 samples are expected to be taken in the in-control state and 1 sample is expected to be taken during the out-of-control state. The expected cost of sampling and inspection is \$ 187.99 (0.1343 per unit) and the expected cost of finding the assignable cause and repairing the process is \$ 112.62 (0.0804 per unit) which includes the cost of false alarms \$ 12.62. Lastly the cost of producing nonconforming units is \$ 463.41 (0.3310 per unit).

Some Details of Outcome of Hook-Jeeves Technique for the Present Example :-

As explained in the Section 2.3.4 one has to apply Hooke-Jeeves procedure repeatedly in order to avoid the possibility of local optimization. We have applied Hooke-Jeeves procedure repeatedly giving 16 different triplets (n_0, m_0, k_0) of initial values. The 16 triplets of the initial values and the resulting optimal values with minimum ECPU are given in the following Table 2.1 as columns (1), (2), (3) respectively. It can be seen from the Table 2.1 that ECPU decreases as both n and k increase upto a certain stage and then it shows an increasing trend as we further increase both n and k.

As seen from the column (2) the optimal solution is (40, 2, 346) with ECPU = \$ 0.5476. Again we tried (40, 2, 346) as an initial value and the resulting exact optimal solution is (37, 2, 350) with minimum ECPU = \$ 0.5457. This triplet was used as a triplet of initial values and it was seen that the resulting optimal triplet is again the same as the initial triplet. Hence finally we could conclude that the optimal solution is (37,2,350) with minimum ECPU as \$ 0.5457.

Table 2.1

Initial Values (n_o, m_o, k_o)	The resulting optimal values	ECPU for the optimal values
(1)	(2)	(3)
(10, 2, 30)	(13, 2, 57)	$ 0.7698
(15, 2, 60)	(13, 2, 61)	$ 0.7448
(15, 2, 70)	(15, 2, 72)	$ 0.7138
(20, 2, 80)	(15, 2, 84)	$ 0.6829
(13, 2, 85)	(14, 2, 85)	$ 0.6791
(20, 2, 90)	(17, 2, 90)	$ 0.6522
(25, 2,100)	(18, 2,109)	$ 0.6363
(30, 2,110)	(20, 2,119)	$ 0.6260
(35, 2,120)	(21, 2,134)	$ 0.6104
(40, 2,130)	(23, 2,153)	$ 0.5938
(50, 2,150)	(25, 2,178)	$ 0.5790
(30, 2,200)	(31, 2,209)	$ 0.5689
(40, 2,300)	(40, 2,346)	$ 0.5476*
(50, 2,500)	(48, 2,522)	$ 0.5529
(45, 2,400)	(48, 2,521)	$ 0.5534
(55, 2,600)	(63, 2,1097)	$ 0.6184

2.4 Multiple Assignable Cause Model

2.4.1 The Production Process and the Sampling Scheme

The production process starts in the in-control state E_0 in which it produces a known acceptable proportion p_0 of the nonconforming units. There are assignable causes which have the effect of shifting the process from the in-control state E_0 to any of the s out-of-control states E_1, E_2, ..., E_s. When the process is in state E_j ($j = 0, 1, \ldots, s$) the proportion of nonconforming units produced is p_j ($j = 0, 1, \ldots, s$). We assume that $p_0 < p_1 < \ldots < p_s$. When the process is in state E_j the assignable causes occur according to the Poison process at a rate λ_j per hour of operating time. This means that the operating time until the process remains in the state E_j ($j = 0, 1, \ldots, s$) is an exponential random variable with mean $1/\lambda_j$ hours of operating time. At the end of this exponential duration, it moves to a state E_k ($k > j$) with probability a_{jk} such that $\sum_{k=j+1}^{s} a_{jk} = 1$. Thus the transition of the process from state E_j to E_k is possible provided the direction of the movement is towards quality deterioration. Once the process reaches to state E_s no further quality deterioration is possible and the process will stay in E_s until the shift is detected.

The sampling and the inspection procedure is as follows. A sample of size n is taken after every production of k units. The n units so chosen are inspected. Let d be the number of nonconforming units detected in a sample of size n. If $d < m$ the process is declared to be in the in-control state and the production continues. If d $\geq$ m the process is declared to be in the out-of-control state and a search for the assignable causes is undertaken. The production may or may not be stopped at this stage. Incidentally, m is known as the rejection number. Thus the sampling and inspection proceudre remains the same as that explained when the production process is affected by single assignable cause.

We want to find the optimal values of the design variables (n, m, k) which minimize the expected cost per unit of controlling the production process during the production cycle.

2.4.2 Description of Certain Terms required to develop the Expected Cost Model

n = the number of units examined in a sample.

m = the rejection number.

k = the number of units produced between two successive samples.

R = the number of units produced per hour of operating time.

k/R = the time interval between two successive samples.

$1/\lambda_i$ = the average time spent by the process in the state E_i ($i = 0, 1, \ldots, s-1$) before shifting to state E_j ($j > i$).

Γ_i = the average time spent by the process in the state E_i before shifting to the state E_j ($j > i$) given that the shift occurs between two successive samples.

As shown by Duncan (1956),

$$\Gamma_i = \frac{1 - (1 + \lambda_i k/R)\exp(-\lambda_i k/R)}{\lambda_i[1 - \exp(-\lambda_i k/R)]}$$

$$i = 0, 1, \ldots, s \qquad \ldots(2.4.1)$$

Δ_i = the average fraction of time spent by the process in the state E_i before shifting to the state E_j ($j > i$) given that the shift occurs between two successive samples

$\Delta_i = \Gamma_i R/k$ ...(2.4.2)

N(i) = the expected number of samples taken in the state E_i ($i = 0, 1, \ldots, s$).

$$N(i) = \frac{\exp(-\lambda_i k/R)}{1-\exp(-\lambda_i k/R)} \qquad i = 0, 1, \ldots, s-1$$

$$= 1/\beta_s \qquad i = s \qquad \ldots(2.4.3)$$

We derive the expression for N(i) in the following manner.

Let X_i be the number of samples taken in the state E_i ($i = 0, 1, \ldots, s-1$). Then X_i is a random variable taking the values 0, 1, 2, ...

$P(X_i = r)$ = the probability that r samples are taken in the state E_i

= the probability that the shift occurs between rth and (r+1)th samples

$$= \int_{rk/R}^{(r+1)k/R} \lambda_i \exp(-\lambda_i t)dt \qquad \ldots(2.4.4)$$

$$= \exp(-\lambda_i kr/R)[1 - \exp(-\lambda_i k/R)] \qquad \ldots(2.4.5)$$

Furthermore, noting that the expected number of samples taken when the process is in state E_i being the following summmation

$$\sum_{r=0}^{\infty} r\, P(X_i = r) \qquad \ldots(2.4.6)$$

one gets the expression (2.4.3) for N(i) for $i = 0, 1, \ldots s-1$ after substitution of (2.4.5) in (2.4.6) and evaluation of the sum. The expression for N(s) is obvious from the expectation of the geometric distribution with parameter β_s where the meaning of β_s can be had from the definition of β_i which follow next.

β_i = the probability of concluding (on the basis of a single sample) that the process is out of control when it is in state E_i $(i = 0, 1, \ldots, s)$

$$= \sum_{d=m}^{n} \binom{n}{d} p_i^d (1-p_i)^{n-d} \qquad i = 0, 1, \ldots, s \qquad \ldots(2.4.7)$$

$\beta(i,N(i))$ = the probability of concluding on the basis of N(i) samples that the process is out of control when it is in state E_i

$$= \beta_i + (1-\beta_i)\beta_i + (1-\beta_i)^2\beta_i + \ldots + (1-\beta_i)^{N(i)-1}\beta_i$$

$$= 1 - (1-\beta_i)^{N(i)}$$

$$i = 0, 1, \ldots, s-1 \qquad \ldots(2.4.8)$$

$$\beta(s,N(s)) = \beta_s + (1-\beta_s)\beta_s + (1-\beta_s)^2\beta_s + \ldots$$

$$= 1 \qquad \ldots(2.4.9)$$

M(i) = the expected number of samples required to detect the shift given that the shift is detected when the process is in state E_i

$$= 1\,\beta_i + 2\,(1-\beta_i)\beta_i + 3\,(1-\beta_i)^2\beta_i + \ldots$$

$$\ldots + N(i)\,(1-\beta_i)^{N(i)-1}\beta_i$$

$$i = 0, 1, 2, \ldots, s-1 \qquad \ldots(2.4.10)$$

$$M(s) = 1\ \beta_s + 2\ (1 - \beta_s)\beta_s + 3\ (1 - \beta_s)^2\beta_s + \ldots$$
$$= 1/\beta_s \qquad \ldots(2.4.11)$$

2.4.3 The Number of Transitions required to detect the Shift

Let T_i be the event that the shift in the process is detected after the ith transition in the process ($i = 1,2,\ldots,s$).

The probability of detecting the shift in the process after the first transition is

$$P(T_1) = \sum_{i=1}^{s} a_{oi}\ \beta(i,\ N(i)) \qquad \ldots(2.4.12)$$

The probability of detecting the shift in the process after the second transition is

$$P(T_2) = \sum_{i=o}^{s-1} a_{oi}\ \{1 - \beta(i,N(i))\}\ \{\sum_{j=i+1}^{s} a_{ij}\ \beta(j,N(j))\}$$
$$\ldots(2.4.13)$$

The present model assumes that $P(T_j)$ $j = 3,\ 4,\ \ldots$ are negligible. That is the shift in the process is detected by the chart before the third transition occurs in the process. The justification in adopting this assumption is given in the next few lines.

Chiu (1976) developed the economic design for np-control chart using the Duncan's (1971) multiple assignable cause model for $\bar{X}$-control chart. Chiu assumes that once the process shifts to an out-of-control state it is free from further deterioration until the shift is detected by the chart. This means that the shift in the processes is detected by the chart before the second transition occurs. Chiu (1976) himself has mentioned that this assumption is somewhat debatable. But he adopts it for the mathematical simplicity and takes the support from Duncan's (1971) observation in this respect.

However, for the model under our study it is observed for many numerical examples that the probability of detecting the shift after only one transition, $P(T_1)$, is not sufficiently large. Also it is observed that the sum $P(T_1)+P(T_2)$ is sufficiently large. Therefore the model under this study assumes that the shift in the process is detected before the third transition occurs in the process. This assumption makes the mathematical model complicated but we adopt it because it is more realistic. Because of this assumption we use $P(T_1)+P(T_2) \geq 0.90$ as a desirable side condition while finding the optimal design variables of the np-control chart.

2.4.4 The Expected Cost Model

We develop an expected cost model to compute the expected cost per unit of the product during the production cycle.

The total cost C consists of three components C_1, C_2, C_3 where

C_1 = the cost of sampling and inspection,

C_2 = the cost of finding the assignable causes and repairing the process,

C_3 = the cost of producing nonconforming units.

We develop the expected cost model in the following four steps.

Step I

To derive the expressions for $E_i(C_1)$ (i=1,2), $E_i(C_2)$ (i=1,2), $E_i(C_3)$ (i=1,2) under both the situations T_1 and T_2.

Step II

To derive the expressions for the expected number of units produced under both the situations.

Step III

To find the expected total cost per unit $ECPU_i$ (i = 1,2) under both the situations T_1 and T_2.

Step IV

To find the expected total cost per unit ECPU using $P(T_1)$ and $P(T_2)$.

Step I

Under the situation T_1, the shift in the process is detected after one transition in the process. Hence N(o) samples are taken in the in-control state and M(i) (i = 1,2, ... s) samples are taken in state E_i (i = 1,2, ... s). Hence the expected number of samples taken under the situation T_1 is given by

$$L_1 = N(o) + \sum_{i=1}^{s} a_{oi} \; M(i) \qquad \text{...(2.4.14)}$$

Under the situation T_2, the shift in the process is detected after second transition. Hence the expected number of samples taken under the situation T_2 is given by

$$L_2 = N(o) + \sum_{i=1}^{s-1} a_{oi} \left[N(i) + \sum_{j=i+1}^{s} a_{ij} \; M(j) \right] \qquad \text{...(2.4.15)}$$

Hence the expected costs of sampling and inspection under the situations T_1 and T_2 are given by

$$E_1(C_1) = (a_1+a_2n)L_1 \qquad \ldots(2.4.16)$$

$$E_2(C_1) = (a_1+a_2n)L_2 \qquad \ldots(2.4.17)$$

where

a_1 = the fixed cost of sampling,

a_2 = the variable cost per unit of sampling.

The expected number of false alarms, B_o, remains the same under both the situations T_1 and T_2 and is given as

$$B_o = N(o)\ \beta_o$$

Hence the expected cost of finding the assignable causes and repairing the process remains the same under T_1 and T_2. It is given by

$$E_1(C_2) = E_2(C_2) = a_{3,1}B_o + a_{3,2}1 \qquad \ldots(2.4.18)$$

where

$a_{3,1}$ = the cost of searching for a false alarm,

$a_{3,2}$ = the cost of searching for a true alarm and repairing the process.

The expected number of nonconforming units produced under T_1 is given by

$$D_1 = \{\ N(o)k + \Delta_o\ k\ \}p_o + \sum_{i=1}^{s} a_{oi}\ \{\ M(i)k - \Delta_o\ k\ \}p_1 \qquad \ldots(2.4.19)$$

The expected number of the nonconforming units produced under T_2 is given by

$$D_2 = \{ N(o)k + \Delta_o\ k \}p_o + \sum_{i=1}^{s-1} a_{oi} \Big[\{ N(i)k - \Delta_o\ k + \Delta_i\ k \}p_i + \sum_{j=i+1}^{s} a_{ij} \{ M(j)k - \Delta_i\ k \}p_j \Big] \qquad ...(2.4.20)$$

The expected number of the nonconforming units detected during sampling under T_1 is given by

$$S_1 = N(o)np_o + \sum_{i=1}^{s} a_{oi}\ M(i)np_i \qquad ...(2.4.21)$$

The expected number of the nonconforming units detected during sampling under the situation T_2 is given by

$$S_2 = N(o)np_o + \sum_{i=1}^{s-1} a_{oi} \Big[N(i)np_i + \sum_{j=i+1}^{s} a_{ij}\ M(j)np_j \Big] \qquad ...(2.4.22)$$

Hence the expected cost of producing the nonconforming units under the situations T_1 and T_2 are given by

$$E_1(C_3) = a_{4,1}\ S_1 + a_{4,2}\ (D_1 - S_1) \qquad ...(2.4.23)$$

$$E_2(C_3) = a_{4,1}\ S_2 + a_{4,2}\ (D_2 - S_2) \qquad ...(2.4.24)$$

where

$a_{4,1}$ = the cost per nonconforming unit which is detected during sampling,

$a_{4,2}$ = the cost per nonconforming unit which goes undetected to the customer.

Here $a_{4,2} > a_{4,1}$ is likely to hold in many real life problems.

Step II

The expected number of units produced under the situation T_1 is given by

$$U_1 = N(o)k + \sum_{i=1}^{s} a_{oi}\ M(i)k \qquad \ldots(2.4.25)$$

The expected number of units produced under the situation T_2 is given by

$$U_2 = N(o)k + \sum_{i=1}^{s-1} a_{oi} \left[N(i)k + \sum_{j=i+1}^{s} a_{ij}\ M(j)k \right] \qquad \ldots(2.4.26)$$

Step III

The total expected cost under the situation T_1 is given by

$$E_1(C) = E_1(C_1) + E_1(C_2) + E_1(C_3)$$

The total expected cost under the situation T_2 is given by

$$E_2(C) = E_2(C_1) + E_2(C_2) + E_2(C_3)$$

Hence total expected cost per unit under the situation T_1 is given by

$$ECPU_1 = E_1(C)/U_1 \qquad \ldots(2.4.27)$$

The total expected cost per unit under the situation T_2 is given by

$$ECPU_2 = E_2(C)/U_2 \qquad \ldots(2.4.28)$$

Step IV

Multiplying $ECPU_1$ and $ECPU_2$ by their respective probabilities $P'(T_1)$ and $P'(T_2)$, the expected cost per unit of controlling the process during the production cycle is

$$ECPU = ECPU_1 \; P'(T_1) + ECPU_2 \; P'(T_2) \qquad \ldots(2.4.29)$$

where

$$P'(T_1) = \frac{P(T_1)}{P(T_1) + P(T_2)}$$

and

$$P'(T_2) = \frac{P(T_2)}{P(T_1) + P(T_2)} \qquad \ldots(2.4.30)$$

2.4.5 A Numerical Example

We define the transition matrix $((a_{ij}))$ in the following manner

$$a_{ij} = \frac{s - (j-1)}{1+2+ \ldots + (s-i)} \qquad \ldots(2.4.31)$$

$$i = 0, 1, \ldots, s-1, \quad j = 1, 2, \ldots, s.$$

These transition probabilities satisfy the following conditions

(i) $a_{i,i+1} > a_{i,i+2} > \ldots > a_{i,s}$

$$i = 0, 1, \ldots, s-1 \qquad \ldots(2.4.32)$$

(ii) $\sum_{j=i+1}^{s} a_{ij} = 1$

$$i = 0, 1, \ldots, s-1 \qquad \ldots(2.4.33).$$

We define λ_j $(j = 0,1, \ldots,s)$ in the following manner to have increasing trend which is expected to be so in several production processes.

$\lambda_0 = 1$

$\lambda_j = (j+1)\ \lambda_0 \quad j = 1, 2, \ldots, s$.

We have $\lambda_0 = 1$, $\lambda_1 = 2$, $\lambda_2 = 3$, ..., $\lambda_s = s+1$

so that $\lambda_0 < \lambda_1 < \lambda_2 < \ldots < \lambda_s$

Let a_1 = \$1.0, a_2 = \$0.1, $a_{3,1}$ = \$100.0, $a_{3,2}$ = \$100.0,

$a_{4,1}$ = \$10.0, $a_{4,2}$ = \$15.00.

Let R = 1000, S = 6.

Let $(p_0, p_1, \ldots, p_s)$ = (.01, .02, .04, .08, .16, .32, .64)

We have developed computer program on FORTRAN to calculate ECPU given by (2.4.29) for the given values (n_0, m_0, k_0). This program uses many subroutines. The listing associated with all the programs developed are given at the end of this chapter.

Hook-Jeeves' search technique described in Section 2.3.4 is used to find the optimal values of n, m and k which minimize the expected cost per unit of the product, ECPU, given by (2.4.29). The optimal procedure yielded is n = 12, m = 2, k = 31 with the minimum ECPU = \$0.3773. For the optimal procedure we give the values of some intermediate terms required in the calculation of ECPU.

$P(T_1)$ = 0.6691, $P(T_2)$ = 0.2883

L_1 = 32, L_2 = 41

U_1 = 1014, U_2 = 1280

D_1 = 12, D_2 = 30

S_1 = 5, S_2 = 13

$ECPU_1$ = \$0.3398, $ECPU_2$ = \$0.4643.

It is seen from the above numerical example that the probability of detecting the shift after one transition, namely $P(T_1)$, is 0.6691 which is not sufficiently large. However it is observed that $P(T_1) + P(T_2) = 0.9574$ which is sufficiently large. This justifies the assumption that the shift in the process is detected before the third transition occurs. This is in accordance with the explanation given in Section 2.4.3 of this chapter, while justifying why we go upto two transitions and not one as done by Chiu (1976).

The interpretation of the numerical output of the rest of the terms given above is not difficult, if one recalls the meaning of the various symbols. Instead, we give the behaviour of the objective function ECPU being minimized in a tabular form in Table 2.2. It may be noted that as mentioned in the Hook-Jeeves' procedure it is applied repeatedly to avoid the local minimization, and what is given in Table 2.2 is merely an extract of several calculations. It is revealed from column (5) of the table that the minimum cost (0.3773) occurs at the triplet n=12, m=2, k=31 satisfying the condition $P(T_1)+P(T_2) > 0.90$.

Table 2.2

(n, m, k)	$P(T_1)$	$P(T_2)$	$P(T_1)+P(T_2)$	ECPU
(1)	(2)	(3)	(4)	(5)
(10,2,30)	0.6022	0.3297	0.9319	0.3863
(11,3,30)	0.3442	0.4249	0.7691	0.4103
(11,1,30)	0.9888	0.0111	0.9999	0.6479
(13,2,31)	0.7007	0.2663	0.9670	0.3792
(13,3,31)	0.3913	0.4197	0.8110	0.3880
(13,1,31)	0.9931	0.0068	0.9999	0.6968
(16,2,32)	0.7710	0.2118	0.9828	0.3917
(15,3,32)	0.4388	0.4112	0.8500	0.3873
(15,1,32)	0.9958	0.0042	1.0000	0.7349
(12,2,31)	0.6991	0.2883	0.9874	0.3773*
(12,3,31)	0.3638	0.4221	0.7859	0.3917
(12,1,31)	0.9905	0.0049	0.9954	0.6638
(14,2,32)	0.7203	0.2516	0.9719	0.3899
(15,2,33)	0.7406	0.2366	0.9772	0.3902
(15,2,31)	0.7558	0.2245	0.9803	0.3862

```
C LISTING OF CHAPTER II
      SUBROUTINE OBJ2(AKE,NSTAGE,SUMN,A1,A2,A3,A4,A5,RATE,ALEMDA,
     1 PNOT,PONE)
      DIMENSION  AKE(10)
C  PROGRAM FOR FINDING ECPU OF np-CHART FOR SINGLE ASSIGNABLE
C  CAUSE MODEL
C FILE NAME IS NANDI1
 1    FORMAT(1X,5F10.4)
      WRITE(*,1)A1,A2,A3,A4,A5
 3    FORMAT(1X,2F10.4)
      WRITE(*,3)ALEMDA,RATE
      WRITE(*,3)PNOT,PONE
      SNOT=AKE(1)
      SRNOT=AKE(2)
      REJNOT=AKE(3)
      WRITE(*,5)SNOT,SRNOT,REJNOT
 5    FORMAT(1X,3F10.4)
      POWER=ALEMDA*SRNOT/RATE
      PPOWER=-POWER
      THEETA=EXP(PPOWER)
      WRITE(*,7)THEETA
 7    FORMAT(1X,F10.6)
      MM=REJNOT
      NT=SNOT-REJNOT+1
      CALL BIN(PONE,MM,NT,CPR,CPL,PI)
      QONE=CPR
      WRITE(*,8)QONE
 8    FORMAT(1X,F10.6)
      MM=REJNOT
      NT=SNOT-REJNOT+1
      CALL BIN(PNOT,MM,NT,CPR,CPL,PI)
      QNOT=CPR
      WRITE(*,8)QNOT
      R=1/QONE+THEETA/(1-THEETA)
C  IR IS EXPECTED NO OF SAMPLES REQUIED TO DETECT SHIFT
      IR=R+0.5
      WRITE(*,9)IR
 9    FORMAT(1X,I3)
C   COMPUTATION  OF EXPECTED COSTS
      EC1=(A1+A2*SNOT)*IR/(IR*SRNOT)
      EC2=A3*(QNOT*THEETA/(1-THEETA)+1)/(IR*SRNOT)
      ADALTA=(1-(1+POWER)*THEETA)/(POWER*(1-THEETA))
      WRITE(*,35)ADALTA
 35   FORMAT(1X,'ADALTA=', F10.6)
      D=THEETA*SRNOT*PNOT/(1-THEETA)+ADALTA*SRNOT*PNOT+SRNOT*PONE/
     1 QONE-ADALTA*SRNOT*PONE
      DS=THEETA*SNOT*PNOT/(1-THEETA)+SNOT*PONE/QONE
      WRITE(*,15)D,DS
 15   FORMAT(1X,2F10.4)
C   DS GIVES  EXPECTED  NO OF  DEFECTIVES  DETECTED  IN  SAMPLING
C   D  GIVES   EXPECTED  NO  OF  DEFECTIVES PRODUCED
      EC3=(A4*DS+A5*(D-DS))/(IR*SRNOT)
      TC=EC1+EC2+EC3
      SUMN=TC
      WRITE (*,30)TC,EC1,EC2,EC3
 30   FORMAT(1X,'TC=',F10.6,'EC1=',F10.6,'EC2=',F10.6,'EC3=',F10.6)
      RETURN
      END
```

```
C FILE NAME NSH1.FOR
      SUBROUTINE BIN(X, MM, NT, P,PP, PIND)
C PROGRAM FOR  CALCU INDIVIDUAL AND CUMULATIVE
C PROBABILITY OF  BINOMIAL DISTRIBUTION
34    DIMENSION AA(301)
      DOUBLE PRECISION  AA, RN, AANOT, RK
       NN=NT+MM-1
      RN=NN
      AANOT=(1.-X)**RN
      AA(1)=(RN*X*AANOT)/(1.-X)
        DO 25 K=2,NN
        RK=K
25      AA(K)=(X*(RN-RK+1.)*AA(K-1))/(RK*(1.-X))
        P=0
        DO 4 I=MM,NN
4       P=P+AA(I)
        PP=1.-P
        M=MM-1
        IF (M.EQ.0) GO TO 6
        PIND=AA(M)
        GO TO 7
6       PIND=AANOT
7       CONTINUE
C P  GIVES  PROB  FROM  M+1 TO N
        RETURN
        END


        SUBROUTINE OBJ1(AKE,NSTAGE,SUM,A1,A2,A3,A4,A5,NSTAT,RATE,
     1  ALEMDA,PIN)
C NAME OF THE FILE  SNN7.FOR
C  PROGRAM FOR  ECPU  OF np-CHART FOR MULTIPLE ASSIGNABLE CAUSE
C  MODEL
        DIMENSION ALEMDA(10), PIN(10),  TAW(10), ADALTA(10),  BEETA
     1  (10), PRDS(10), NTOS(10),S(10), R(10), V(10), W(10),
     1  X(10), M(10),A(10,10),AKE(5)
 1      FORMAT(1X,5F10.4)
 2      FORMAT(1X,I5,F10.4)
 3      FORMAT(1X,7F10.4)
 4      FORMAT(1X,3F10.5)
        WRITE(*,1) A1,A2,A3,A4,A5
        WRITE(*,2) NSTAT,RATE
        WRITE(*,3) (ALEMDA(I),I=1,NSTAT)
        WRITE(*,3) (PIN(I),I=1,NSTAT)
        SNOT=AKE(1)
        SRNOT=AKE(2)
        REJNOT=AKE(3)
        WRITE(*,4) SNOT,SRNOT,REJNOT
        NSTATE=NSTAT-1
        CALL   AVTIME(SRNOT,RATE,ALEMDA,NSTAT,TAW,ADALTA)
        WRITE(*,5) (TAW(I),I=1,NSTATE)
 5      FORMAT(1X,6F10.5)
        WRITE(*,5) (ADALTA(I),I=1,NSTATE)
        CALL PROBR(SNOT,REJNOT,PIN,NSTAT,BEETA)
        WRITE(*,3) (BEETA(I),I=1,NSTAT)
        CALL ENOS(SRNOT,RATE,ALEMDA,NSTAT,NTOS)
```

```
      WRITE(*,6) (NTOS(I),I=1,NSTATE)
 6    FORMAT(1X,7I7)
      CALL PRODS(BEETA,NTOS,NSTAT,PRDS)
      WRITE(*,3) (PRDS(I),I=1,NSTAT)
      CALL ESDS(NSTAT,BEETA,NTOS,M)
      WRITE(*,6) (M(I),I=1,NSTAT)
      TER=0
      DO 10 I=1,NSTATE
      DO 15 J=1,NSTAT
      IF(J.GT.I) GO TO 12
      GO TO 13
 12   A(I,J)=(FLOAT(NSTAT-J+1)*2.0)/FLOAT((NSTAT-I)*(NSTAT-I+1))
      GO TO 15
 13   A(I,J)=0
 15   CONTINUE
 10   CONTINUE
      WRITE(*,11) ((A(I,J),J=1,NSTAT),I=1,NSTATE)
 11   FORMAT(1X,7F10.6)
C     COMPUTATION  OF  PT1,PT2
      TERM=0
      DO 21 J=2,NSTAT
 21   TERM=TERM+A(1,J)*PRDS(J)
      PT1=TERM
      DO 25 I=2,NSTATE
      TERM2=0
      I1=I+1
      DO 26 J=I1,NSTAT
 26   TERM2=TERM2+A(I,J)*PRDS(J)
 25   S(I)=A(1,I)*(1-PRDS(I))*TERM2
      TERM3=0
      DO 27 I=2,NSTATE
 27   TERM3=TERM3+S(I)
      PT2=TERM3
      WRITE(*,28) PT1,PT2
 28   FORMAT(1X,'PT1=',F10.6,'PT2=',F10.6)
C  COMPUTATION OF EC1 UNDER T1 T2
      TERM4=0
      DO 29 I=2,NSTAT
 29   TERM4=TERM4+A(1,I)*M(I)
      L1=NTOS(1)+TERM4
      DO 30 I=2,NSTATE
      TERM5=0
      I1=I+1
      DO 31 J=I1,NSTAT
 31   TERM5=TERM5+A(I,J)*M(J)
 30   R(I)=A(1,I)*(NTOS(I)+TERM5)
      TERM6=0
      DO 33  I=2,NSTATE
 33   TERM6=TERM6+R(I)
      L2=NTOS(1)+TERM6
      WRITE(*,35) L1,L2
 35   FORMAT(1X,'L1=',I5,'L2=',I5)
      E1C1=(A1+A2*SNOT)*L1
      E2C1=(A1+A2*SNOT)*L2
      WRITE(*,80) E1C1,E2C1
 80   FORMAT(1X,'E1C1=',F10.3,'E2C1=',F10.3)
```

```
C  COMPUTATION OF EC2 UNDER T1 T2
      E1C2=A3*(NTOS(1)*BEETA(1)+1)
      E2C2=E1C2
      WRITE(*,81) E1C2,E2C2
 81   FORMAT(1X,'E1C2=',F10.3,'E2C2=',F10.3)
C   COMPUTATION OF EC3 UNDER  T1  T2
      TERM7=0
      DO 37 I=2,NSTAT
 37   TERM7=TERM7+A(1,I)*(M(I)*SRNOT-ADALTA(I)*SRNOT)*PIN(I)
      ND1=(NTOS(1)*SRNOT+ADALTA(1)*SRNOT)*PIN(1)+TERM7
      TERM9=0
      DO 40 I=2,NSTATE
      I1=I+1
      TERM8=0
      DO 45 J=I1,NSTAT
 45   TERM8=TERM8+A(I,J)*(M(J)*SRNOT-ADALTA(I)*SRNOT)*PIN(J)
      V(I)=TERM8
 40   TERM9=TERM9+A(1,I)*((NTOS(I)*SRNOT-ADALTA(1)*SRNOT+
     1 ADALTA(I)*SRNOT)*PIN(I)+V(I))
      ND2=(NTOS(1)*SRNOT+ADALTA(1)*SRNOT)*PIN(1)+TERM9
      WRITE(*,38)ND1,ND2
 38   FORMAT(1X,'ND1=',I9,'ND2=',I9)
      TERM10=0
      DO 50 I=2,NSTAT
 50   TERM10=TERM10+A(1,I)*M(I)*PIN(I)*SNOT
      NS1=NTOS(1)*SNOT*PIN(1)+TERM10
      TERM12=0
      DO 55 I=2,NSTATE
      I1=I+1
      TERM11=0
      DO 56 J=I1,NSTAT
 56   TERM11=TERM11+A(I,J)*M(J)*SNOT*PIN(J)
      W(I)=TERM11
 55   TERM12=TERM12+A(1,I)*(NTOS(I)*SNOT*PIN(I)+W(I))
      NS2=NTOS(1)*SNOT*PIN(1)+TERM12
      WRITE(*,57) NS1,NS2
 57   FORMAT(1X,'NS1=',I7,'NS2=',I7)
      E1C3=A4*NS1+A5*(ND1-NS1)
      E2C3=A4*NS2+A5*(ND2-NS2)
      WRITE(*,85) E1C3,E2C3
 85   FORMAT(1X,'E1C3=',F10.3,'E2C3=',F10.3)
C   COMPUTATION OF EXPECTED NO OF UNITS UNDER TI  T2
      TERM 13=0
      DO 60 I=2,NSTAT
 60   TERM 13=TERM13+A(1,I)*M(I)*SRNOT
      NU1=NTOS(1)*SRNOT+TERM13
      TERM15=0
      DO 65 I=2,NSTATE
      TERM14=0
      I1=I+1
      DO 66 J=I1,NSTAT
 66   TERM14=TERM14+A(I,J)*M(J)*SRNOT
      X(I)=TERM14
 65   TERM15=TERM15+A(1,I)*(NTOS(I)*SRNOT+X(I))
      NU2=NTOS(1)*SRNOT+TERM15
      WRITE(*,68) NU1,NU2
 68   FORMAT(1X,'NU1=',I8,'NU2=',I8)
```

```
C  COMPUTATION OF EC UNDER  T1  T2
      E1C=(E1C1+E1C2+E1C3)/NU1
      E2C=(E2C1+E2C2+E2C3)/NU2
      WRITE(*,87) E1C,E2C
 87   FORMAT(1X,'E1C=',F10.6,'E2C=',F10.6)
C  COMPUTATION OF  EC
      TRPT1=PT1/(PT1+PT2)
      TRPT2=PT2/(PT1+PT2)
      EC=E1C*TRPT1+E2C*TRPT2
      WRITE(*,72) EC
 72   FORMAT(1X,'EC=',F10.6)
      SUM=EC
      RETURN
      END


      SUBROUTINE  ENOS(SRNOT,RATE,ALEMDA,NSTAT,NTOS)
C SUBROUTINE FOR TOTAL NO OF SAMPLES IN STATE I N(I)
C  FILE NAME IS SNN2
      DIMENSION ALEMDA(10),POWER(10),PPOWER(10),NTOS(10)
      NSTATE=NSTAT-1
      DO 1 I=1,NSTATE
      POWER(I)=ALEMDA(I)*SRNOT/RATE
      PPOWER(I)=-POWER(I)
      NTOS(I)=EXP(PPOWER(I))/(1-EXP(PPOWER(I)))
 1    CONTINUE
      RETURN
      END


      SUBROUTINE AVTIME(SRNOT,RATE,ALEMDA,NSTAT,TAW,ADALTA)
C  SUBROUTINE FOR TAW(I) AND ADALTA(I)
C  FILE NAME IS SNN3
      DIMENSION ALEMDA(10),POWER(10),PPOWER(10),TAW(10),ADALTA(10)
      NSTATE=NSTAT-1
      DO 1 I=1,NSTATE
      POWER(I)=ALEMDA(I)*SRNOT/RATE
      PPOWER(I)=-POWER(I)
      TAW(I)=  (1-(1+POWER(I))*EXP(PPOWER(I)))/(ALEMDA(I)-
     1ALEMDA(I)*EXP(PPOWER(I)))
      ADALTA(I)=TAW(I)*RATE/SRNOT
 1    CONTINUE
      RETURN
      END


      SUBROUTINE PROBR(SNOT,REJNOT,PIN,NSTAT,BEETA)
C  SUBROUTINE FOR PROB OF DETECTING SHIFT ON THE BASIS OF ITH SAMPLE
C FILE NAME IS SNN4
      DIMENSION PIN(10),BEETA(10)
      DO 1 I=1,NSTAT
      PROB=PIN(I)
      MM=REJNOT
      NN=SNOT
```

HOOKE AND JEEVES (HOOKE ALGORITHM)*

(Listing reproduced from Kuester and Mize (1973))

A. Purpose

This program finds the minimum of a multivariable, unconstrained, nonlinear function:

Minimize $F(X_1, X_2, \ldots, X_N)$

B. Method

The procedure is based on the direct search method proposed by Hooke and Jeeves (30). No derivatives are required. The procedure assumes a unimodal function; therefore, if more than one minimum exists or the shape of the surface is unknown, several sets of starting values are recommended. The algorithm proceeds as follows:

1) A base point is picked and the objective function evaluated.
2) Local searches are made in each direction by stepping X_i a distance S_i to each side and evaluating the objective function to see if a lower function value is obtained.
3) If there is no function decrease, the step size is reduced and searches are made from the previous best point.
4) If the value of the objective function has decreased, a "temporary head", $X_{i,o}^{(k+1)}$, is located using the two previous base points $X_i^{(k+1)}$ and $X_i^{(k)}$:

$$X_{i,o}^{(k+1)} = X_i^{(k+1)} + \alpha(X_i^{(k+1)} - X_i^{(k)})$$

where i is the variable index = 1, 2, 3, ..., N
o denotes the temporary head
k is stage index (a stage is the end of N searches)
α is an acceleration factor, $\alpha \geq 1$.

5) If the temporary head results in a lower function value, a new local search is performed about the temporary head, a new head is located and the value of F checked. This expansion continues as long as F decreases.

---

*Computer code developed by A. I. Johnson, University of Western Ontario, Canada. Used by permission.

6) If the temporary head does not result in a lower function value, a search is made from the previous best point.

7) The procedure terminates when the convergence criterion is satisfied (see Description of Parameters).

A flow sheet illustrating the above procedure is given in Figure 9.II.

C. Program Description

1) Usage:

The program consists of a short main program, the main subroutine HOOKE and the user supplied functional evaluation subroutine OBJECT. Initial values of the independent variables, step sizes, and solution parameters are supplied through the main program. Subroutine HOOKE performs all searches and provides all printout.

2) Subroutine Required:

SUBROUTINE HOOKE (RK, EPS, NSTAGE, MAXK, NKAT, EPSY, ALPHA, BETA, QD, Q, QQ, W, IPRINT) called from main program, performs all searches.

SUBROUTINE OBJECT (SUMN, AKE, NSTAGE) function evaluation subroutine (user supplied).

3) Description of Parameters:

NSTAGE	Number of decision variables to be used
RK	Vector of initial guesses for decision variables
EPS	Vector of initial step size to be used for each of the variables
ITMAX	Maximum number of times the objective function is called (=MAXK)
NKAT	Maximum number of times the initial step size is to be reduced
EPSY	Error in objective function to be reached before program terminates (difference between current value and previous stage value)
ALPHA	Factor for extending the size of the initial steps, greater than or equal to 1.0
BETA	Factor for reducing the initial step size, $0.0 \leq BETA \leq 1.0$
QD	Optimum value of the function resulting from the search
AKE	Vector of independent variables in subroutine OBJECT
SUMN	Objective function to be minimized - define in OBJECT
IPRINT	Print control. IPRINT = 0 results in no intermediate output. IPRINT = 1 results in output on each iteration

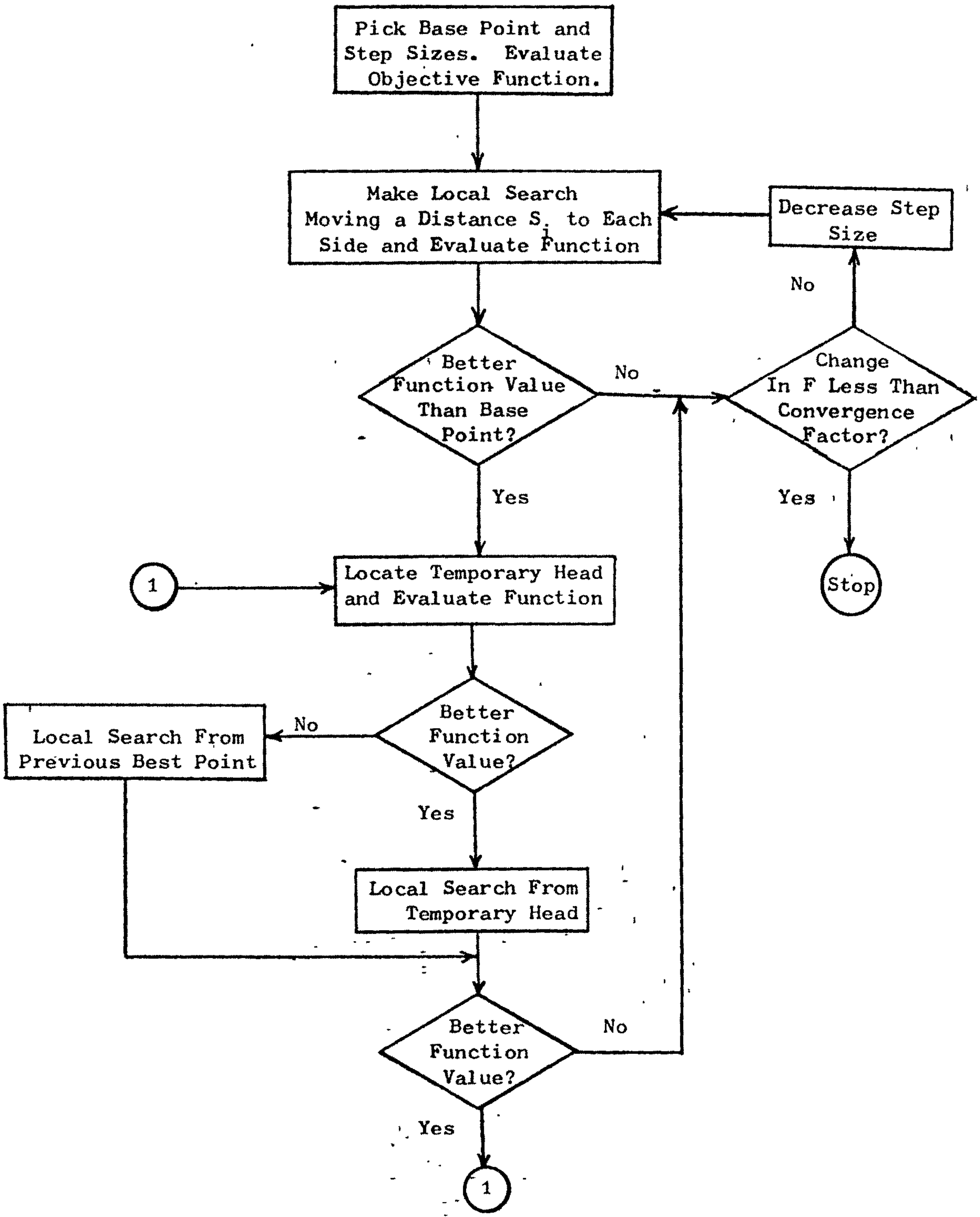

Figure 9.II. Hooke and Jeeves (HOOKE ALGORITHM) Logic Diagram

NI Card reader unit number

NO Printer unit number

4) DIMENSION Requirements:

The DIMENSION statement in main program should be modified accord to the requirements of the particular problem. The parameters included in the following DIMENSION statement conform to the Input Parameter definitions above:

DIMENSION EPS (NSTAGE), RK(NSTAGE), Q(NSTAGE), QQ(NSTAGE), W(NST

5) Input Formats:

CARD TYPE	FORMAT	CONTENTS
1	(8I10)	NSTAGE, IPRINT, ITMAX, NKAT
2	(8E10.4)	(RK(II), II = 1,NSTAGE)
	(If N > 8, additional CARD TYPE 2's will be requirec	
3	(8E10.4)	(EPS(JJ), JJ = 1,NSTAGE)
	(If N > 8, additional CARD TYPE 3's will be require	
4	(8E10.4)	ALPHA, BETA, EPSY

6) Output:

All output is from subroutine HOOKE. Initial parameter value printed. Intermediate results are printed, if the user speci IPRINT = 1 on Card Type 1. Final results are printed upon termination.

7) Summary of User Requirements:

a) Determine values for NSTAGE, IPRINT, ITMAX, NKAT, NI, an
b) Determine initial estimates of independent variables; er as (RK(II), II = 1, NSTAGE).
c) Specify initial step sizes; enter as (EPS(JJ), JJ = 1,N
d) Determine values for ALPHA, BETA, AND EPSY observing th rules stated in Description of Parameters section.
e) Adjust DIMENSION in main program.
f) Specify objective function by writing SUBROUTINE OBJEC
g) Adjust FORMAT statements as necessary.

D. Test Problem

The following test program was taken from the literature Calculations were performed on a CDC 6400 computer.

Function: $F = -3803.84 - 138.08X_1 - 232.92X_2 + 123.08X_1^2 + 203.64X_2^2 + 182.25X_1X_2$

Starting Point: $X_1 = 1.0$, $X_2 = 0.5$

Parameters: N = 2, ITMAX = 500, NKAT = 20,
EPSY = 0.00001, ALPHA = 1.0, BETA = 0.5

Initial Step Sizes: EPS(1) = 0.10, EPS(2) = 0.10

Algorithm Answers: F = - 3873.9

$X_1 = 0.20576$

$X_2 = 0.47979$

Number of Function Evaluations: 110

Central Processor Time: 3 seconds

The listing and output for this problem are contained in the following section.

E. Program Listings and Example Output

```
C
C
C     MAIN LINE PROGRAM FOR SUBROUTINE HOOKE.
C
      DIMENSION EPS(2), RK(2), Q(2), QQ(2), W(2)
      COMMON NI,NO
C
      NI = 50
      NO = 66
C
      READ (NI,001) NSTAGE, IPRINT, ITMAX, NKAT
  001 FORMAT (8I10)
      READ (NI,002) (RK(II), II=1,NSTAGE)
  002 FORMAT (8E10.4)
      READ (NI,002) (EPS(JJ), JJ=1,NSTAGE)
C
      READ (NI,003) ALPHA, BETA, EPSY
  003 FORMAT (8E10.4)
      QD = 0.0
C
      CALL HOOKE (RK,EPS,NSTAGE,ITMAX,NKAT,EPSY,ALPHA,BETA,QD,Q,QQ,W,
     1 IPRINT)
C
      END
```

```
      SUBROUTINE HOOKE (RK,EPS,NSTAGE,MAXK,NKAT,EPSY,ALPHA,BETA,QD,
     1 Q,QQ,W,IPRINT)
      DIMENSION RK(NSTAGE), EPS(NSTAGE), Q(NSTAGE), QQ(NSTAGE),
     1 W(NSTAGE)
```

```
      COMMON NI,NO

      WRITE (NO,001)
  001 FORMAT (1H1,10X,37HHOOKE AND JEEVES OPTIMIZATION ROUTINE)
      WRITE (NO,002) ALPHA, BETA, MAXK, NKAT
  002 FORMAT (//,2X,10HPARAMETERS,/,2X,8HALPHA = ,F5.2,4X,
     1 7HBETA = ,F5.2,4X,8HITMAX = ,I4,4X,7HNKAT = ,I3)
      WRITE (NO,003) NSTAGE
  003 FORMAT (/,2X,22HNUMBER OF VARIABLES = ,I3)
      WRITE (NO,004)
  004 FORMAT (/,2X,18HINITIAL STEP SIZES)
      DO 6  I=1,NSTAGE
      WRITE (NO,005)  I, EPS(I)
  005 FORMAT (/,2X,4HEPS(,I2,4H) = ,E16.8)
    6 CONTINUE
      WRITE (NO,007) EPSY
  007 FORMAT (/,2X,43HERROR IN FUNCTION VALUES FOR CONVERGENCE = ,E16.8)
      KFLAG = 0
      DO 601 I=1,NSTAGE
      Q(I) =RK(I)
      W(I) = 0.0
  601 CONTINUE
      KAT =0.0
      KK1 =0
   70 KCOUNT =0
      WBEST = W(NSTAGE)
      CALL OBJECT (SUM,RK,NSTAGE)
      KK1= KK1+ 1
      BO =SUM
      IF (KK1.EQ. 1)  QD = SUM
      IF (KK1.EQ. 1)   GO TO 201
      IF(BO.GT.QD)   KFLAG = 1
      IF (BO.LT.QD)  QD = BO

         ESTABLISHING THE SEARCH PATTERN

  201 DO   55   I = 1,NSTAGE
      QQ(I)=RK(I)
      TSRK = RK(I)
      RK(I) = RK(I) + EPS(I)
      CALL OBJECT (SUM,RK,NSTAGE)
      KK1= KK1+ 1
      W(I) = SUM
      IF (W(I) .LT.QD)   GO TO 58
      RK(I) = RK(I) - 2.0*EPS(I)
      CALL OBJECT (SUM,RK,NSTAGE)
      KK1= KK1+ 1
      W(I) = SUM
      IF (W(I) .LT.QD)   GO TO 58
      RK(I) = TSRK
      IF (I.EQ. 1) GO TO 513
      W(I) =W(I-1)
```

```
      GO TO 613
  513 W(I) =DO
  613 CONTINUE
      KCOUNT =1+ KCOUNT
      GO TO 55
   58 QD= W(I)
      QQ(I) =RK(I)
   55 CONTINUE
      IF (IPRINT) 60, 65, 60
   60 WRITE (NO,100) KK1
C
C         RECORD RESPONSES AND LOCATION
C
      WRITE(NO,102)
      WRITE(NO,207) (RK(I), I=1,NSTAGE), QD
C
C         TEST TO DETERMINE TERMINATION OF PROGRAM
C
   65 IF (KK1.GT.MAXK) GO TO 94
      IF (KAT .GE. NKAT)   GO TO 94
      IF (ABS(W(NSTAGE)-WBEST).LE.EPSY) GO TO 94
C
C         IF ALL AXES FAIL REDUCE STEP SIZE
C
      IF (KCOUNT .GE. NSTAGE ) GO TO 28
      DO 26   I = 1,NSTAGE
      RK(I) =RK(I) + ALPHA*(RK(I) - Q(I))
   26 CONTINUE
      DO 25    I = 1,NSTAGE
      Q(I) =QQ(I)
   25 CONTINUE
      GO TO 70
C
C         REDUCE STEP SIZE
C
   28 KAT = KAT + 1
      IF (KFLAG .EQ. 1)   GO TO 202
      GO TO 204
  202 KFLAG = 0
      DO 203   I = 1,NSTAGE
      RK(I) = Q(I)
  203 CONTINUE
  204 DO 80    I=1,NSTAGE
      EPS(I) =EPS(I) *BETA
   80 CONTINUE
      IF (IPRINT) 85, 70, 85
   85 WRITE (NO,101) KAT
      GO TO 70
   94 WRITE (NO,460) (EPS(I), I=1,NSTAGE)
      WRITE (NO,461) (RK(I), I=1,NSTAGE)
      WRITE (NO,462) QD
      DO 104 I=1,NSTAGE
  104 WRITE (NO,103) I, RK(I)
```

```
      WRITE (NO,100) KK1
  100 FORMAT (//,2X,33HNUMBER OF FUNCTION EVALUATIONS = ,I8)
  101 FORMAT (/,2X,18HSTEP SIZE REDUCED ,I2,6H TIMES)
  102 FORMAT(1X,26HEND OF EACH PATTERN SEARCH/)
  103 FORMAT (//,2X,8HFINAL X(,I2,4H) = ,1PE16.8)
  207 FORMAT(1X,18HVARIABLES AND SUMN,3X,9E12.4//)
  465 FORMAT (10X,3HSUM,3X,E14.5)
  460 FORMAT(1X, 18H THE FINAL EPS ARE,   4F20.8/)
  461 FORMAT (1X, 18H THE FINAL RK ARE ,   5F20.8/)
  462 FORMAT (1X, 24H THE MINIMUM RESPONSE IS,    F20.8/)
      RETURN
      END




      SUBROUTINE OBJECT (SUMN,AKE,NSTAGE)

      DIMENSION AKE(NSTAGE)

      X1 = AKE(1)
      X2 = AKE(2)
      X12 = (X1**2)
      X22 = (X2**2)
      SUMN = 3803.84 + 138.08*X1 + 232.92*X2 - 123.08*X1**2 - 203.64
     1 *X2**2 - 182.25*X1*X2
      SUMN = - SUMN

      RETURN
      END
```

HOOKE AND JEEVES OPTIMIZATION ROUTINE

PARAMETERS
ALPHA = 1.00 BETA = 0.50 ITMAX = 500 NKAT = 20

NUMBER OF VARIABLES = 2

INITIAL STEP SIZES

EPS(1) = 0.10000000E+00

EPS(2) = 0.10000000E+00

ERROR IN FUNCTION VALUES FOR CONVERGENCE = 0.10000000E-04

NUMBER OF FUNCTION EVALUATIONS = 5
END OF EACH PATTERN SEARCH

VARIABLES AND SUMN 0.9000E+00 0.4000E+00 -.3823E+04

(9 intervening printouts are omitted.)

NUMBER OF FUNCTION EVALUATIONS = 50
END OF EACH PATTERN SEARCH

VARIABLES AND SUMN 0.2250E+00 0.4750E+00 -.3874E+04

STEP SIZE REDUCED 4 TIMES

(11 intervening printouts are omitted.)

```
 NUMBER OF FUNCTION EVALUATIONS =        106
END OF EACH PATTERN SEARCH

VARIABLES AND SUMN      0.2059E+00  0.4797E+00  -.3874E+04

 STEP SIZE REDUCED 10 TIMES

 NUMBER OF FUNCTION EVALUATIONS =        110
END OF EACH PATTERN SEARCH

VARIABLES AND SUMN      0.2058E+00  0.4798E+00  -.3874E+04
 THE FINAL EPS ARE           0.00009766              0.00009766
 THE FINAL RK ARE            0.20576172              0.47978516
 THE MINIMUM RESPONSE IS       -3873.92354660


 FINAL X( 1) =    2.0576172E-01


 FINAL X( 2) =    4.7978516E-01


 NUMBER OF FUNCTION EVALUATIONS =        110
```