CHAPTER V

Use of More Powerful Decision Rule on Economic Design of $\bar{x}$-Control Charts

5.1 Woodall (1986) raised some controversy to the economic approach to design the control charts. He mentioned that the economic model balances the cost of poor quality against the cost of sampling and the cost of finding the assignable causes and repairing the process. The statistical performance of the economically designed control chart is ignored. He, furthermore, mentioned that in many cases the calculated cost savings may be misleading and the optimal economic control charts may have poor statistical performance.

However, we feel that there is no point in debating like this. One should decide in advance whether one wants to design the control chart

(i) to have the minimum cost

or (ii) to have the better statistical performance

or (iii) to have both simultaneously.

If one decides to have the first case one should accept whatsoever the design that comes up. If one decides to have the second case then it may be noted that he is indifferent about the cost aspect. However, the third case is a sort of compromize. If one decides to have the third case one can optimize the cost subject to achievement of better statistical performance. But in this case it is likely that one has to pay more cost, in the sense that the minimum cost under the third case may be more than the minimum cost under the first case.

In this chapter we study the effects of using a more powerful decision rule on the total expected cost of Knappenberger and Grandage's (1969) cost model. It is observed that the total expected cost of controlling the process increases when the more powerful decision rule is used.

5.2 Knappenberger and Grandage's (1969) Cost Model.

Knappenberger and Gandage (1969) developed an economic model for $\bar{x}$-chart for a process subject to multiplicity of assignable causes. Montgomery et al. (1975) used this model in the development of np-chart. The model is well explained in chapter IV while studying the model under different sampling schemes. In the Knappenberger and Grandage's (1969) model for $\bar{x}$-chart, it is assumed that the process mean μ is a continuous variable which can satisfactorily be approximated by a discrete variable taking the values μ_0, μ_1, ... μ_s. The value μ_0 is associated with the in-control state and the remaining values μ_1, μ_2, ..., μ_s are associated with the out-of-control states.

The total expected cost per unit under the Knappenberger and Grandage's (1969) model (recalling (4.2.6) and making suitable changes for complete sampling) is

$$E(C) = \frac{a_1+a_2n}{k} + \frac{a_3\underline{\alpha}'\underline{q}}{k} + a_4\underline{r}'\underline{f} \qquad (5.2.1)$$

Here, the first term of the above expression corresponds to the expected cost per unit of sampling and inspection and is denoted by $E(C_1)$, the second (or the middle) term corresponds to the expected cost per unit of finding the assignable causes and repairing the process and is denoted by $E(C_2)$, the third (or the last) term corresponds to the expected cost per unit of producing nonconfoming units and is denoted by $E(C_3)$.

The cost coefficients $a_i (i = 1,2,3,4)$ involved in (5.2.1) are known and are independent of the design variables.

The explanation of the vector $\underline{f}$ is as follows. The term f_i $(i = 0,1, \ldots, s)$ is the pobability of getting a nonconforming unit when the process is in the state μ_i $(i = 0,1, \ldots, s)$. In the Knappenberger and Grandage's (1969) model it is assumed that a unit is nonconforming if its measurement falls outside the limits $\mu_0 \pm 3\sigma$. Using this assumption and specifying $\mu_i = \mu_0 + i\sigma$ $(i = 1,2, \ldots, s)$ as done by Knappenberger and Grandage (1969) the pobabilities f_i $(i = 0,1, \ldots, s)$ can be easily obtained from the standard normal tables.

The probability vectors $\underline{q}$, $\underline{\alpha}$, $\underline{r}$ are functionally related to the design variables. The vector $\underline{q}$ is explained in the next section. For the given vector $\underline{q}$, the procedures for evaluating $\underline{\alpha}$ and $\underline{r}$ are well explained in Section 4.2.4 of chapter IV. The expression for vector $\underline{\alpha}$ is given by (4.2.15) and the expressions for r_0 and r_i $(i = 1,2, \ldots, s)$ are given by (4.2.16) and (4.2.18) respectively.

5.3 Use of More Powerful Decision Rule

The decision rule given by Knappenberger and Grandage (1969) on the basis of inspection of n units after the production of every k units is as follows :

Reject the hypothesis $H_o : \mu = \mu_o$, if the sample point falls outside the control limits $\mu_o - L\sigma/\sqrt{n}$ and $\mu_o + L\sigma/\sqrt{n}$. According to this decision rule the probability of rejecting $H_o : \mu = \mu_o$ under $H_1 : \mu = \mu_i$ is

$$
\begin{aligned}
q_i &= P(\bar{x} > \mu_o + L\sigma/\sqrt{n} \mid \mu = \mu_i) + P(\bar{x} < \mu_o - L\sigma/\sqrt{n} \mid \mu = \mu_i) \\
&= P(\bar{x} > \mu_o + L\sigma/\sqrt{n} \mid \mu = \mu_o + i\sigma) + P(\bar{x} < \mu_o - L\sigma/\sqrt{n} \mid \mu = \mu_o + i\sigma) \\
&= P(Z > L - i\sqrt{n}) + P(Z < -L - i\sqrt{n}) \qquad \text{...(5.3.1)}
\end{aligned}
$$

where Z is a standard normal variable.

It may be noted that q_i represents the power of the decision rule when $\mu = \mu_i$ (i = 0, 1, ..., s).

The proposed more powerful decision rule is as given below.

Reject the hypothesis $H_o : \mu = \mu_o$ not only when a sample point falls outside the control limits but also when seven successive sample points fall between the center line and the upper control limit or between the lower control limit and the center line.

The decision rule proposed above is a well known statistical technique to detect the shift in the process average. It is based on the theory of runs. It is discussed, for instance, by E. L. Grant and R. S. Leavenworth (1980). Under this new decision rule, the probability of rejecting $H_o : \mu = \mu_o$ is given by

$$w_i = q_i + [\ P\ (\ \mu_o < \bar{x} < \mu_o + L\sigma/\sqrt{n}\ |\ \mu = \mu_i\ )]^7$$
$$+ [\ P\ (\ \mu_o - L\sigma/\sqrt{n} < \bar{x} < \mu_o\ |\ \mu = \mu_i\ )]^7 \quad \ldots(5.3.2)$$

Here w_i $(i = 0, 1, \ldots, s)$ represents the power of the new decision rule.

Since $w_i > q_i$ we can see that the power of the new decision rule is grater than that of the former.

The total expected cost per unit for the new decision rule is

$$E(C) = (a_1+a_2n)/k + a_3\underline{\alpha}'\underline{w}/k + a_4\underline{r}'\underline{f} \quad \ldots(5.3.3)$$

The breakup of the r.h.s. of (5.3.3) into $E(C_1)$, $E(C_2)$ and $E(C_3)$ is as obvious as done just after (5.2.1).

It appears as if the only difference between the expressions (5.2.1) and (5.3.3) is that the vector $\underline{q}$ is replaced by the vector $\underline{w}$. However, it is something more than this. It should be remembered that the vectors $\underline{\alpha}$ and $\underline{r}$ are to be evalued a fresh from the expressions (4.2.15), (4.2.16) and (4.2.18) after substitution of $\underline{w}$ in place of $\underline{q}$. The new values obtained for the vectors $\underline{\alpha}$ and $\underline{r}$ are to be substituted in the expression (5.3.3) to evaluate E(C) which gives ultimately the total expected cost per unit for the new rule.

5.4 Comparision of the two Decision Rules from the Cost Point of View

5.4.1 The Sample Example

To make the comparision of the two decision rules from the cost point of view we consider the following example given in

Knappenberger and Grandage (1969).

Let a_1 = \$ 10, a_2 = \$ 1, a_3 = \$ 100, a_4 = \$ 10.

Let λ = 1, R = 1000, π = 0.376.

For this combination of the cost coefficients and systems parameters, the optimal values of the design variables (n, k, L) are obtained by Knappenberger and Grandage. We could have utilized these values for our study. But we prefer to recalculate them independently for the following reasons.

(1) The mistakes are noted in their expressions for the matrix B^*. The corrected expressions are given in Section 8.2 of Chapter VIII. The numerical values of B^* obtained by them are also not correct.

(2) They give only the optimal values of the design variables (n, k, L) with minimum E(C). No intermediate terms such as $E(C_1)$, $E(C_2)$, $E(C_3)$ are given in their paper. So the information given by them is inadequate when we make a comparative study of the performance of the two decision rules from the cost point of view.

The corrected optimal values which minimize the expression (5.2.1) for the total expected cost per unit of the product (under the original rule) are given bellow in the tabular form. Using these values of (n, k, L) we calculate $E(C_1)$, $E(C_2)$, $E(C_3)$ and E(C) (under the original rule) given by the expression (5.2.1). Using the same values of (n, k, L) we calculate $E(C_1)$, $E(C_2)$, $E(C_3)$ and E(C) under the proposed new decision rule given by the expression (5.3.3).These expected costs are as given below

Design Variables Used	Expected Costs (Original Rule)	Expected Costs (New Rule)
$n = 3$	$E(C_1) = 0.2826$	$E(C_1) = 0.2826$
$k = 46$	$E(C_2) = 0.1104$	$E(C_2) = 0.1413$
$L = 2.70$	$E(C_3) = 0.3421$	$E(C_3) = 0.3303$
	$E(C) = 0.7351$	$E(C) = 0.7542$

We have the following results from this example.

(1) The expected cost per unit of sampling, $E(C_1)$, remains the same under both the decision rules and it should be the case as the same triplete is used for both the rules.

(2) The expected cost per unit of finding the assignable causes and repairing the process, $E(C_2)$, increases when a more powerful decision rule is used.

(3) The expected cost per unit of producing nonconforming items, $E(C_3)$, decreases when a more powerful decision rule is used.

(4) The increase in $E(C_2)$ is grater than the decrease in $E(C_3)$. Hence the total expected cost per unit of the product increases when a more powerful decision rule is used.

The probability vectors $\underline{q}$ and $\underline{w}$ found under the two decision rules are as follows. The optimal design variables $n = 3$, $k = 46$, $L = 2.70$ are used for the calculation of both $\underline{q}$and $\underline{w}$.

	μ_0	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6
Original Rule ($\underline{q}$)	.0069	.1665	.7776	.9937	.9999	1.00	1.00
New Rule ($\underline{w}$)	.0218	.3616	.7776	.9937	.9999	1.00	1.00
Difference	.0149	.1952	0	0	0	0	0
% difference	215.94	117.24	0	0	0	0	0

From the above figures it is ravealed that under the new decision rule the probability of detecting a small shift namely, the shift from the state μ_0 to state μ_1 is increased from 0.1665 to 0.3617. Thus there is an increase of 117.24 %. This shows that the smaller shifts in the process are detected with high probability if the new decision rule is used. Of course, it also has a consequence of increasing the probability of false alarms (from 0.0069 to 0.0218 i.e. 215.94 %). The effect of this increase has the natural reflection into the cost aspect by increasing $E(C_2)$ and hence $E(C)$.

5.4.2 Additional Examples

We now consider 18 combinations of the cost coefficients and the systems parameters given by Knappenberger and Grandage (1969). Using Hook-Jeeves direct search technique we obtain the optimal values of the design variables (n, k, L) under the original rule for all the 18 cases. These optimal values of (n, k, L) minimize the expected total cost per unit of the product

(under the original rule) whose expression is given by (5.2.1). Using these optimal values we calculate $E(C_1)$, $E(C_2)$, $E(C_3)$, $E(C)$ under the original rule.

Furthermore, using the same optimal values of (n, k, L) we calculate $E(C_1)$, $E(C_2)$, $E(C_3)$, $E(C)$ (under the new rule) whose expression is given by (5.3.3).

These values are given in Table 5.1. It is observed that in all the 18 cases the behavior of $E(C_1)$, $E(C_2)$, $E(C_3)$ and $E(C)$ are analogous to those observed in the sample example. i.e. $E(C_1)$ attains the same values under both the decision rules, $E(C_2)$ increases under the new rule whereas $E(C_3)$ decreases under the new rule. Furthermore, it is observed that the increase in $E(C_2)$ is grater than the decrease in $E(C_3)$, which ultimately leads to a higher expected total cost.

From these additional examples also we find that the use of a more powerful decision rule has an effect of increasing the total cost of controlling the production process.

5.5 More Exact Comparision

In section 5.4 the optimal values of the design variables (n, k, L) are obtained under the original decision rule. Using these optimal values of (n, k, L) we have calculated $E(C_1)$, $E(C_2)$, $E(C_3)$ and $E(C)$ for both the decision rules. Then comparisions are made between $E(C_1)$, $E(C_2)$, $E(C_3)$, $E(C)$ derived under the two decision rules. It seems to be customary to use the same optimal design variables even after making some modifications in the mathematical structure of the model such as using the new decision rule in place of the original rule.

For instance, in an example given in Montgomery's (1991) book on page number 420 he has explained the problem of minimizing the total expected cost. His cost model involves three design variables (n, h, k) where n is the sample size, h is the interval between two successive samples and k is the multiple of $\sigma/\sqrt{n}$. In the solution he obtains the optimal values of (n,h,k) as follows. n = 5, k = 2.99, h = 0.62. He finds that the optimal values of h = 0.62 is not convenient from the operation point of view. He proposes to take h = 0.75 (i.e. 45 minutes) with the same optimal values of n and k. In reality he should have considered the problem of minimizing the total expected cost which is now a function of only two design variables (n, k) with h = 0.75 as one of the constants in addition to the set of constants that he already has. The optimal values of (n, k) derived in this situation may not be the same as the original n = 5 and k = 2.99. In fact had he proposed to take h = 1, the optimal values for n and k would be n = 6 and k = 2.99.

Somewhat objectionable situation similar to the one as described in the above paragraph exists in the problem being studied here. The values of the design variables (n, k, L) which minimize E(C) given by (5.2.1) may not be the same as the values of (n, k, L) which minimize E(C) given by (5.3.3). This means that the optimal values of (n, k, L) under the original rule may not be the same as the optimal values of (n, k, L) under the new rule.

To make the comparision more realistic one should compute the values of (n, k, L) which minimize E(C) given by (5.3.3). These optimal values will minimize E(C) under the new rule. In this section we obtain the optimal values of (n, k, L) which minimize E(C) under the new rule for all the 18 cases studied in Section 5.4. The optimal design variables and the minimum E(C) under the new rule are given in columns (7) and (8) respectively of Table 5.2. The optimal design variables and the minimum E(C) under the original rule are also given in columns (5) and (6) of Table 5.2 for comparision.

For all the 18 cases it is observed from columns (6) and (8) of Table 5.2 that the minimum E(C) derived under the new rule are grater than the minimum E(C) derived under the original rule. This result is the same as the one observed in Section 5.4.

Thus we find that both the comparisons - the less sophisticated comparision as done in Section 5.4 as well as the more exact comparision done in this section lead to the same conclusion that the use of the more powerful decision rule increases the total expected cost of controlling the process. If one wants to use a control chart with more powerful decision rule one should be prepared to pay a higher cost. Of course, the advantage in using a more powerful decision rule is obvious that one would detect more often the smaller shift in the process when it exists.

Table 5.1

The optimal design variables of the original rule and $E(C_1)$, $E(C_2)$, $E(C_3)$, $E(C)$ of both the rules calculated for these design variables.

a_1 (1)	a_2 (2)	a_3 (3)	π (4)	Optimal design variables (Original Rule) (5)	Expected costs (Original Rule) (6)	Expected costs (New Rule) (7)
10	1	10	0.376	n = 2 k = 46 L = 1.70	$E(C_1)$ = 0.2608 $E(C_2)$ = 0.0280 $E(C_3)$ = 0.3207 E(C) = 0.6095	$E(C_1)$ = 0.2608 $E(C_2)$ = 0.0297 $E(C_3)$ = 0.3200 E(C) = 0.6105
10	1	100	0.376	n = 3 k = 46 L = 2.70	$E(C_1)$ = 0.2826 $E(C_2)$ = 0.1104 $E(C_3)$ = 0.3421 E(C) = 0.7351	$E(C_1)$ = 0.2826 $E(C_2)$ = 0.1413 $E(C_3)$ = 0.3303 E(C) = 0.7542
10	1	1000	0.376	n = 4 k = 50 L = 3.45	$E(C_1)$ = 0.2800 $E(C_2)$ = 0.9663 $E(C_3)$ = 0.3822 E(C) = 1.6285	$E(C_1)$ = 0.2800 $E(C_2)$ = 1.2671 $E(C_3)$ = 0.3503 E(C) = 1.8974

Here a_4 = \$ 10, λ = 1, R = 1000, s = 6

Table 5.1 (Continued)

a_1 (1)	a_2 (2)	a_3 (3)	π (4)	Optimal design variables (Original Rule) (5)	Expected costs (Original Rule) (6)	Expected costs (New Rule) (7)
10	1	10	0.597	n = 2 k = 34 L = 2.20	$E(C_1)$ = 0.3529 $E(C_2)$ = 0.0169 $E(C_3)$ = 0.3945 --- E(C) = 0.7643	$E(C_1)$ = 0.3529 $E(C_2)$ = 0.0213 $E(C_3)$ = 0.3936 --- E(C) = 0.7679
10	1	100	0.597	n = 2 k = 33 L = 2.95	$E(C_1)$ = 0.3636 $E(C_2)$ = 0.1071 $E(C_3)$ = 0.4028 --- E(C) = 0.8736	$E(C_1)$ = 0.3636 $E(C_2)$ = 0.1518 $E(C_3)$ = 0.3987 --- E(C) = 0.9142
10	1	1000	0.597	n = 3 k = 36 L = 3.65	$E(C_1)$ = 0.3611 $E(C_2)$ = 0.9839 $E(C_3)$ = 0.4340 --- E(C) = 1.7790	$E(C_1)$ = 0.3611 $E(C_2)$ = 1.4011 $E(C_3)$ = 0.4265 --- E(C) = 2.1887

Table 5.1 (Continued)

a_1 (1)	a_2 (2)	a_3 (3)	π (4)	Optimal design variables (Original Rule) (5)	Expected costs (Original Rule) (6)	Expected costs (New Rule) (7)
10	1	10	0.800	n = 1 k = 29 L = 2.35	$E(C_1) = 0.3793$ $E(C_2) = 0.0161$ $E(C_3) = 0.4296$ $E(C) = 0.8250$	$E(C_1) = 0.3793$ $E(C_2) = 0.0207$ $E(C_3) = 0.4293$ $E(C) = 0.8294$
10	1	100	0.800	n = 1 k = 28 L = 2.95	$E(C_1) = 0.3928$ $E(C_2) = 0.1093$ $E(C_3) = 0.4304$ $E(C) = 0.9326$	$E(C_1) = 0.3928$ $E(C_2) = 0.1623$ $E(C_3) = 0.4293$ $E(C) = 0.9844$
10	1	1000	0.800	n = 2 k = 30 L = 3.90	$E(C_1) = 0.4000$ $E(C_2) = 0.9866$ $E(C_3) = 0.4455$ $E(C) = 1.8321$	$E(C_1) = 0.4000$ $E(C_2) = 1.4911$ $E(C_3) = 0.4438$ $E(C) = 2.3349$

Table 5.1 (Continued)

a_1 (1)	a_2 (2)	a_3 (3)	π (4)	Optimal design variables (Original Rule) (5)	Expected costs (Original Rule) (6)	Expected costs (New Rule) (7)
100	1	10	0.376	n = 4 k = 151 L = 1.00	$E(C_1)$ = 0.6887 $E(C_2)$ = 0.0272 $E(C_3)$ = 0.8247 E(C) = 1.5407	$E(C_1)$ = 0.6887 $E(C_2)$ = 0.0272 $E(C_3)$ = 0.8247 E(C) = 1.5407
100	1	100	0.376	n = 8 k = 152 L = 2.20	$E(C_1)$ = 0.7105 $E(C_2)$ = 0.1075 $E(C_3)$ = 0.8417 E(C) = 1.6597	$E(C_1)$ = 0.7105 $E(C_2)$ = 0.1147 $E(C_3)$ = 0.8417 E(C) = 1.6669
100	1	1000	0.376	n = 10 k = 162 L = 3.05	$E(C_1)$ = 0.6790 $E(C_2)$ = 0.9193 $E(C_3)$ = 0.9136 E(C) = 2.5120	$E(C_1)$ = 0.6790 $E(C_2)$ = 0.9988 $E(C_3)$ = 0.9130 E(C) = 2.5909

Table 5.1 (Continued)

a_1 (1)	a_2 (2)	a_3 (3)	π (4)	Optimal design variables (Original Rule) (5)	Expected costs (Original Rule) (6)	Expected costs (New Rule) (7)
100	1	10	0.597	n = 3 k = 107 L = 1.65	$E(C_1)$ = 0.9626 $E(C_2)$ = 0.0177 $E(C_3)$ = 1.0843 E(C) = 2.0647	$E(C_1)$ = 0.9626 $E(C_2)$ = 0.0184 $E(C_3)$ = 1.0843 E(C) = 2.0653
100	1	100	0.597	n = 4 k = 107 L = 2.65	$E(C_1)$ = 0.9719 $E(C_2)$ = 0.1011 $E(C_3)$ = 1.0965 E(C) = 2.1696	$E(C_1)$ = 0.9719 $E(C_2)$ = 0.1135 $E(C_3)$ = 1.0939 E(C) = 2.1794
100	1	1000	0.597	n = 5 k = 110 L = 3.35	$E(C_1)$ = 0.9545 $E(C_2)$ = 0.9482 $E(C_3)$ = 1.1315 E(C) = 3.0343	$E(C_1)$ = 0.9545 $E(C_2)$ = 1.0755 $E(C_3)$ = 1.1227 E(C) = 3.1528

Table 5.1 (Continued)

a_1 (1)	a_2 (2)	a_3 (3)	π (4)	Optimal design variables (Original Rule) (5)	Expected costs (Original Rule) (6)	Expected costs (New Rule) (7)
100	1	10	0.800	n = 2 k = 92 L = 2.20	$E(C_1)$ = 1.1087 $E(C_2)$ = 0.0123 $E(C_3)$ = 1.2129 E(C) = 2.3339	$E(C_1)$ = 1.1087 $E(C_2)$ = 0.0135 $E(C_3)$ = 1.2128 E(C) = 2.3351
100	1	100	0.800	n = 2 k = 92 L = 2.85	$E(C_1)$ = 1.1087 $E(C_2)$ = 0.0997 $E(C_3)$ = 1.2198 E(C) = 2.4283	$E(C_1)$ = 1.1087 $E(C_2)$ = 0.1010 $E(C_3)$ = 1.2195 E(C) = 2.4292
100	1	1000	0.800	n = 3 k = 95 L = 3.65	$E(C_1)$ = 1.0842 $E(C_2)$ = 0.9551 $E(C_3)$ = 1.2551 E(C) = 3.2944	$E(C_1)$ = 1.0842 $E(C_2)$ = 1.1043 $E(C_3)$ = 1.2542 E(C) = 3.4427

Table 5.2

The optimal design variables and $E(C_1)$, $E(C_2)$, $E(C_3)$, $E(C)$ under both the rules

a_1 (1)	a_2 (2)	a_3 (3)	π (4)	Original Rule: Optimal design variables (5)	Original Rule: Expected costs (6)	New Rule: Optimal Design variables (7)	New Rule: Expected costs (8)
10	1	10	0.376	n = 2 k = 46 L = 1.70	$E(C_1)$ = 0.2608 $E(C_2)$ = 0.0280 $E(C_3)$ = 0.3207 E(C) = 0.6095	n = 2 k = 46 L = 1.70	$E(C_1)$ = 0.2608 $E(C_2)$ = 0.0297 $E(C_3)$ = 0.3200 E(C) = 0.6105
10	1	100	0.376	n = 3 k = 46 L = 2.70	$E(C_1)$ = 0.2826 $E(C_2)$ = 0.1104 $E(C_3)$ = 0.3421 E(C) = 0.7351	n = 3 k = 49 L = 2.90	$E(C_1)$ = 0.2653 $E(C_2)$ = 0.1325 $E(C_3)$ = 0.3533 E(C) = 0.7511
10	1	1000	0.376	n = 4 k = 50 L = 3.45	$E(C_1)$ = 0.2800 $E(C_2)$ = 0.9663 $E(C_3)$ = 0.3822 E(C) = 1.6285	n = 4 k = 76 L = 3.50	$E(C_1)$ = 0.1842 $E(C_2)$ = 1.1379 $E(C_3)$ = 0.5083 E(C) = 1.8304

Here $a_4 = 10$, $\lambda = 1$, $R = 1000$, $s = 6$

Table 5.2 (Continued)

a_1 (1)	a_2 (2)	a_3 (3)	π (4)	Original Rule: Optimal design variables (5)	Original Rule: Expected costs (6)	New Rule: Optimal Design variables (7)	New Rule: Expected costs (8)
10	1	10	0.597	n = 2 k = 34 L = 2.20	$E(C_1)$ = 0.3529 $E(C_2)$ = 0.0169 $E(C_3)$ = 0.3945 ——— E(C) = 0.7643	n = 2 k = 34 L = 2.25	$E(C_1)$ = 0.3529 $E(C_2)$ = 0.0204 $E(C_3)$ = 0.3945 ——— E(C) = 0.7679
10	1	100	0.597	n = 2 k = 33 L = 2.95	$E(C_1)$ = 0.3636 $E(C_2)$ = 0.1071 $E(C_3)$ = 0.4028 ——— E(C) = 0.8736	n = 2 k = 36 L = 2.90	$E(C_1)$ = 0.3333 $E(C_2)$ = 0.1483 $E(C_3)$ = 0.4301 ——— E(C) = 0.9117
10	1	1000	0.597	n = 3 k = 36 L = 3.65	$E(C_1)$ = 0.3611 $E(C_2)$ = 0.9839 $E(C_3)$ = 0.4340 ——— E(C) = 1.7790	n = 3 k = 54 L = 3.55	$E(C_1)$ = 0.2407 $E(C_2)$ = 1.2461 $E(C_3)$ = 0.6144 ——— E(C) = 2.1013

Table 5.2 (Continued)

a_1 (1)	a_2 (2)	a_3 (3)	π (4)	Original Rule: Optimal design variables (5)	Original Rule: Expected costs (6)	New Rule: Optimal Design variables (7)	New Rule: Expected costs (8)
10	1	10	0.800	n = 1 k = 29 L = 2.35	$E(C_1)$ = 0.3793 $E(C_2)$ = 0.0161 $E(C_3)$ = 0.4296 E(C) = 0.8250	n = 1 k = 29 L = 2.35	$E(C_1)$ = 0.3793 $E(C_2)$ = 0.0207 $E(C_3)$ = 0.4293 E(C) = 0.8293
10	1	100	0.800	n = 1 k = 28 L = 2.95	$E(C_1)$ = 0.3928 $E(C_2)$ = 0.1093 $E(C_3)$ = 0.4304 E(C) = 0.9326	n = 1 k = 30 L = 2.95	$E(C_1)$ = 0.3666 $E(C_2)$ = 0.1577 $E(C_3)$ = 0.4573 E(C) = 0.9817
10	1	1000	0.800	n = 2 k = 30 L = 3.90	$E(C_1)$ = 0.4000 $E(C_2)$ = 0.9866 $E(C_3)$ = 0.4455 E(C) = 1.8321	n = 2 k = 46 L = 3.70	$E(C_1)$ = 0.2608 $E(C_2)$ = 1.3033 $E(C_3)$ = 0.6544 E(C) = 2.2186

Table 5.2 (Continued)

a_1 (1)	a_2 (2)	a_3 (3)	π (4)	Original Rule: Optimal design variables (5)	Original Rule: Expected costs (6)	New Rule: Optimal Design variables (7)	New Rule: Expected costs (8)
100	1	10	0.376	n = 4 k = 151 L = 1.00	$E(C_1)$ = 0.6887 $E(C_2)$ = 0.0272 $E(C_3)$ = 0.8247 E(C) = 1.5407	n = 4 k = 151 L = 1.00	$E(C_1)$ = 0.6887 $E(C_2)$ = 0.0272 $E(C_3)$ = 0.8247 E(C) = 1.5407
100	1	100	0.376	n = 8 k = 152 L = 2.20	$E(C_1)$ = 0.7105 $E(C_2)$ = 0.1075 $E(C_3)$ = 0.8417 E(C) = 1.6597	n = 8 k = 155 L = 2.15	$E(C_1)$ = 0.6967 $E(C_2)$ = 0.1159 $E(C_3)$ = 0.8540 E(C) = 1.6668
100	1	1000	0.376	n = 10 k = 162 L = 3.05	$E(C_1)$ = 0.6790 $E(C_2)$ = 0.9193 $E(C_3)$ = 0.9136 E(C) = 2.5120	n = 6 k = 167 L = 3.25	$E(C_1)$ = 0.6287 $E(C_2)$ = 0.9761 $E(C_3)$ = 0.9767 E(C) = 2.5815

Table 5.2 (Continued)

				Original Rule		New Rule	
a_1 (1)	a_2 (2)	a_3 (3)	π (4)	Optimal design variables (5)	Expected costs (6)	Optimal Design variables (7)	Expected costs (8)
100	1	10	0.597	n = 3 k = 107 L = 1.65	$E(C_1) = 0.9626$ $E(C_2) = 0.0177$ $E(C_3) = 1.0843$ $E(C) = 2.0647$	n = 3 k = 107 L = 1.65	$E(C_1) = 0.9626$ $E(C_2) = 0.0184$ $E(C_3) = 1.0843$ $E(C) = 2.0653$
100	1	100	0.597	n = 4 k = 107 L = 2.65	$E(C_1) = 0.9719$ $E(C_2) = 0.1011$ $E(C_3) = 1.0965$ $E(C) = 2.1696$	n = 4 k = 108 L = 2.70	$E(C_1) = 0.9629$ $E(C_2) = 0.1124$ $E(C_3) = 1.1039$ $E(C) = 2.1793$
100	1	1000	0.597	n = 5 k = 110 L = 3.35	$E(C_1) = 0.9545$ $E(C_2) = 0.9482$ $E(C_3) = 1.1315$ $E(C) = 3.0343$	n = 3 k = 119 L = 3.50	$E(C_1) = 0.8823$ $E(C_2) = 1.0573$ $E(C_3) = 1.2064$ $E(C) = 3.1461$

Table 5.2 (Continued)

a_1 (1)	a_2 (2)	a_3 (3)	π (4)	Original Rule: Optimal design variables (5)	Original Rule: Expected costs (6)	New Rule: Optimal Design variables (7)	New Rule: Expected costs (8)
100	1	10	0.800	n = 2 k = 92 L = 2.20	$E(C_1) = 1.1087$ $E(C_2) = 0.0123$ $E(C_3) = 1.2129$ $E(C) = 2.3339$	n = 2 k = 92 L = 2.20	$E(C_1) = 1.1087$ $E(C_2) = 0.0135$ $E(C_3) = 1.2128$ $E(C) = 2.3351$
100	1	100	0.800	n = 2 k = 92 L = 2.85	$E(C_1) = 1.1087$ $E(C_2) = 0.0997$ $E(C_3) = 1.2198$ $E(C) = 2.4283$	n = 2 k = 92 L = 2.85	$E(C_1) = 1.1087$ $E(C_2) = 0.1010$ $E(C_3) = 1.2195$ $E(C) = 2.4292$
100	1	1000	0.800	n = 3 k = 95 L = 3.65	$E(C_1) = 1.0842$ $E(C_2) = 0.9551$ $E(C_3) = 1.2551$ $E(C) = 3.2944$	n = 3 k = 102 L = 3.60	$E(C_1) = 1.0098$ $E(C_2) = 1.0901$ $E(C_3) = 1.3371$ $E(C) = 3.4370$

```
C LISTING OF CHAPTER V
C   FILE NAME IS MAN1
C   PROGRAM FOR E(C) OF KNAPPENBER-GRANDAGE MPDÈL
        SUBROUTINE OBJ7 (AKE,NSTAGE,SUMN,A1,A2,A3,A4,ALEMDA,RATE,
     1  PIE,NSTAT,PIN)
        DIMENSION PIN(10),PZ(10),P(10,10),QR(10),ZP(10),BZ(10),ZB(10)
     1  ,B(10,10),BST(10,10),CZ(10),ZC(10),C(10,10),D(10,10),
     1  DST(10,10),ALPHA(10),GAMMA(10),A(10,10),BSTZ(10),T(10,10),ZBST
     1  (10),S(10,10),U(10,10),V(10,10),BB(10,10),AKE(5)
        WRITE(*,5)
 5       FORMAT(4X,'COST COEFFICIENTS')
        WRITE(*,1)A1,A2,A3,A4
 1      FORMAT(1X,4F10.4)
        WRITE(*,3)ALEMBDA,RATE,PIE,NSTAT
 3      FORMAT(1X,3F12.4,I3)
        WRITE(*,6)(PIN(I),I=1,NSTAT)
 6       FORMAT(1X,7F8.4)
        SNOT = AKE(1)
        SRNOT = AKE(2)
        REJNOT = AKE(3)
        WRITE(*,2)SNOT,SRNOT,REJNOT
 2       FORMAT(1X,'SAMPLESIZE =',F10.2,'INT SAM RANGE =',F10.2,'REJ
     1  NUM=',F10.2)
        POWER = ALEMDA*SRNOT/RATE
        PPOWER =-POWER
        PZZ=EXP(PPOWER)
        WRITE(*,7) PZZ
 7      FORMAT(1X,F10.4)
        NSTATE = NSTAT-1
        DENO=1.-(1.-PIE)**NSTATE
        MSNOT = SNOT
        DO 10 J=1,NSTATE
        M1 = J+1
        M2 = NSTATE-J
        CALL BIN(PIE,M1,M2,CPR,CPL,PI)
        WRITE(*,8) CPR,CPL,PI,J
 8      FORMAT(1X,'CPR=',F10.6,'CPL=',F10.6,'PI=',F10.6,'J=',I2)
 10     PZ(J) = PI*(1.-PZZ)/DENO
        DO 600 I = 1,NSTATE
  600    ZP(I)=0.
        DO 20 I=1,NSTATE
        DO 20 J=1,NSTATE
        IF(I-J)30,31,32
   30   P(I,J) = PZ(J)/(1.-PZZ)
        GO TO 20
   31   SPZ=0.
        DO 40 KK=1,I
   40   SPZ = SPZ+PZ(KK)
        P(I,J) = SPZ/(1.-PZZ)
        GO TO 20
   32   P(I,J)=0.
   20  CONTINUE
        T(1,1)=PZZ
        DO 12 I =2,NSTAT
        K=I-1
   12   T(1,I) = PZ(K)
        DO 13 J=2,NSTAT
        K=J-1
```

```
 13     T(J,1)=ZP(K)
        DO 14 I=2,NSTAT
        K=I-1
        DO 14 J=2,NSTAT
        K1 = J-1
 14     T(I,J) = P(K,K1)
        WRITE(*,15)
 15     FORMAT(1X,'TRANSITION MATRIX')
        WRITE(*,11)((T(I,J),J = 1,NSTAT),I=1,NSTAT)
 11     FORMAT(1X,7F10.6)
        S1 = SNOT
        S2 = SRNOT
        S3 = REJNOT
        CALL PROBR(S1,S3,NSTATE,NSTAT,QR)
        WRITE(*,301)(QR(I),I = 1,NSTAT)
301     FORMAT(1X,7F10.6)
        DO 60 I=2,NSTAT
        DO 60 J=1,NSTAT
        IF(I-J)61,62,63
 61       S(I,J)=QR(I)*T(1,J)+(1-QR(I))*T(I,J)
        GO TO 60
 62     S(I,J)=QR(I)*T(1,I)+(1-QR(I))*T(I,I)
        GO TO 60
 63     S(I,J)=QR(I)*T(1,J)
 60     CONTINUE
        DO 326 J=1,NSTAT
326     S(1,J)=T(1,J)
        WRITE(*,302)
302     FORMAT(1X,'MATRIX S(I,J)')
        WRITE(*,303) ((S(I,J),J=1,NSTAT),I=1,NSTAT)
303        FORMAT(1X,7F10.6)
        DO 330 I=1,NSTAT
        DO 330 J=1,NSTAT
        IF(I-J)331,332,331
331        U(I,J)=S(I,J)
        GO TO 330
332        U(I,J)=S(I,J)-1
330        CONTINUE
        WRITE(*,311)
311        FORMAT(1X,'MATRIX U(I,J)')
        WRITE(*,312)((U(I,J),J=1,NSTAT),I=1,NSTAT)
312      FORMAT(1X,7F10.6)
        DO 321 I=1,NSTAT
        DO 321 J=1,NSTATE
321       V(I,J)=U(I,J+1)
        DO 322 I=1,NSTAT
322       V(I,7)=1
        WRITE(*,323)
323       FORMAT(1X,'MATRIX V(I,J)')
        WRITE(*,324) ((V(I,J),J=1,NSTAT),I=1,NSTAT)
324     FORMAT(1X,7F10.6)
        DO 325 I=1,NSTAT
        DO 325 J=1,NSTAT
325        A(I,J)=V(I,J)
        N=NSTAT
        CALL INVRS(A,BB,N)
        WRITE(*,97)
```

```
 97      FORMAT(1X,'INVERSE MATRIX')
         WRITE(*,98) ((BB(I,J),J=1,NSTAT),I=1,NSTAT)
 98      FORMAT(1X,7F10.6)
C    BB(I,J) IS INVERSE OF A(I,J)
         DO 81 J=1,NSTAT
 81      ALPHA(J)=BB(NSTAT,J)
         WRITE(*,150)
 150       FORMAT(1X,'VECTOR ALPHA')
         WRITE(*,82) (ALPHA(J),J=1,NSTAT)
 82      FORMAT(1X,7F10.6)
C COMPUTATION OF GAMMA
         ADALTA=(1.-(1.+POWER)*PZZ)/(POWER*(1.-PZZ))
         WRITE(*,160)
 160         FORMAT(1X,'ADALTA')
         WRITE(*,82) ADALTA
         GAMMAZ=ALPHA(1)*PZZ+ALPHA(1)*ADALTA*(1-PZZ)
         WRITE(*,170)
  170       FORMAT(1X,'GAMMAZ')
         WRITE(*,82) GAMMAZ
         DO 90 I=2,NSTAT
         I3=I-2
         TERM3=0
         TERM4=0
         I1=I-1
         I2=I+1
         IF(I1-1) 101,102,101
 101        DO 100 J=1,I3
         K=J+1
 100      TERM3=TERM3+ALPHA(K)*P(J,I1)
         IF(I1-6) 102,104,102
 102       DO 110 K=I,NSTATE
 110      TERM4=TERM4+P(I1,K)
 104       GAMMA(I1)=ALPHA(I)*P(I1,I1)+(1.-ADALTA)*ALPHA(1)*PZ(I1)+
     1   (1.-ADALTA)*TERM3+ALPHA(I)*TERM4*ADALTA
         WRITE(*,82)GAMMA(I1)
 90      CONTINUE
C    COMPUTATION OF EXPECTATIONS
         EC1=(A1+A2*SNOT)/SRNOT
         TERM5=0
         DO 120 I=1,NSTAT
 120      TERM5=TERM5+QR(I)*ALPHA(I)
         EC2=A3*TERM5/SRNOT
         TERM6=0
         TERM6=TERM6+PIN(1)*GAMMAZ
         DO 130 I=2,NSTAT
         J=I-1
 130      TERM6=TERM6+PIN(I)*GAMMA(J)
         EC3=A4*TERM6
         TC=EC1+EC2+EC3
         SUMN=TC
         WRITE(*,140)TC,EC1,EC2,EC3
 140      FORMAT(1X,'TOTAL COST=',E18.8,'EC1=',E18.8,'EC2=',E18.8,
     1   'EC3=',E18.8)
         RETURN
         END
```

```
C FILE NAME IS MAN2
C  PROGRAM FOR POWER OF ORIGINAL DECISION RULE
       SUBROUTINE PROBR(S1,S3,NSTATE,NSTAT,QR)
       DIMENSION X1(10),X2(10),P1(10),P2(10),Q1(10),Q2(10),Q(10),
     1 QR(10)
       DO 17 I=1,NSTATE
       X1(I)=-I*SQRT(S1)+S3
       X2(I)=-I*SQRT(S1)-S3
       CALL NDTR(X1(I),P1(I),D)
       Q1(I)=1-P1(I)
       CALL NDTR(X2(I),P2(I),D)
       Q2(I)=P2(I)
       Q(I)=Q1(I)+Q2(I)
   17  CONTINUE
       WRITE(*,10)  (Q(I),I=1,NSTATE)
   10  FORMAT(1X,6F10.6)
       CALL NDTR(S3,P,D)
       S=1-P
       QNOT=2.0*S
       WRITE(*,2)QNOT
    2  FORMAT(1X,2F10.6)
       QR(1)=QNOT
       DO 20 I=1,NSTATE
       QR(I+1)=Q(I)
   20  CONTINUE
       WRITE(*,3) (QR(I),I=1,NSTAT)
    3  FORMAT(1X,7F10.6)
       RETURN
       END


C FILE NAME IS NMAN2
C PROGRAM FOR MORE POWERFUL RULE BASE ON 7 POINT FORMULA
       SUBROUTINE PROBR(S1,S3,NSTATE,NSTAT,QR)
       DIMENSION X1(10),X2(10),P1(10),P2(10),Q1(10),Q2(10),Q(10),
     1  QR(10),X3(10),ADQ(10),QN(10),P3(10)
       DO 17 I=1,NSTATE
       X1(I)=-I*SQRT(S1)+S3
       X2(I)=-I*SQRT(S1)-S3
       CALL NDTR(X1(I),P1(I),D)
       Q1(I)=1-P1(I)
       CALL NDTR(X2(I),P2(I),D)
       Q2(I)=P2(I)
       Q(I)=Q1(I)+Q2(I)
   17  CONTINUE
       WRITE(*,10)  (Q(I),I=1,NSTATE)
   10  FORMAT(1X,6F10.6)
       DO 30 I=1,NSTATE
       X3(I)=-I*SQRT(S1)
       CALL NDTR(X3(I),P3(I),D)
       ADQ(I)=(P1(I)-P3(I))**7
       QN(I)=Q(I)+ADQ(I)
   30  CONTINUE
       WRITE(*,10) (QN(I),I=1,NSTATE)
       CALL NDTR(S3,P,D)
       S=1-P
```

```
      QNOT=2.0*S
      WRITE(*,2)QNOT
    2 FORMAT(1X,F10.6)
      QNOTN=QNOT+2.0*((P-0.5)**7)
      WRITE(*,2)QNOTN
      QR(1)=QNOTN
      DO 20 I=1,NSTATE
      QR(I+1)=QN(I)
   20 CONTINUE
      WRITE(*,3) (QR(I),I=1,NSTAT)
    3 FORMAT(1X,7F10.6)
      RETURN
      END


C FILE NAME IS MAN3
C  CALCUATION OF NORMAL INTEGRAL
      SUBROUTINE NDTR(X,P,D)
      AX=ABS(X)
      T=1.0/(1.0+.2316419*AX)
      D=0.3989423*EXP(-X*X/2.0)
      P=1.0-D*T*(((((1.330274*T-1.821256)*T+1.781478)*T-0.3565638)*T
     1 +0.3193815)
      IF(X) 1,2,2
    1 P=1.0-P
    2 RETURN
      END


      SUBROUTINE INVRS(A,B,N)
C     INVERSE OF A MATRIX UPTO 10*10 (GAUSS JORDAN ELIMINATI65 453864'
C     A-INPUT MATRIX (DESTROYED AFTER EXECUTION)
C     B-OUTPUT MATRIX - INVERSE OF A
C     N-ORDER OF A MATRIX
      DIMENSION A(10,10), B(10,10)
C     CHECK DIAGONAL ELEMENTS NON ZERO (NOT DONE)
C     B-IDENTITY MATRIX
      WRITE(*,7) ((A(I,J),J=1,NSTAT),I=1,NSTAT)
    7 FORMAT(1X,7F10.6)
      DO 10 I=1,N
      DO 10 J=1,N
      B(I,J)=0.
      B(I,I)=1.0
   10 CONTINUE
      DO 40 I=1,N
C     I REFERS FIRST N COLUMNS (I-PIVOT ROW)
C     CALCULATION OF ROWS EXCEPT PIVOT ROW

C     CHECK FOR PIVOT ROW
      IF (K.EQ.I) GO TO 20
      CONST = -A(K,I)/A(I,I)
C     CALCULATE ROW ELEMENTS
      DO 30 J=1,N
      A(K,J)=A(K,J)+CONST*A(I,J)
      B(K,J)=B(K,J) + CONST*B(I,J)
```

```
   30  CONTINUE
C      EQUATE A(K,I)=0 TO GET RID OF ROUNDING ERRORS
       A(K,I) = 0.
   20  CONTINUE
C      REFER PIVOT ROW
       CONST = A(I,I)
       DO 50 J=1,N
       A(I,J) = A(I,J)/CONST
       B(I,J) = B(I,J)/CONST
   50  CONTINUE
C      EQUATE A(I,I) = 1.0 TO GET RID OF ROUNDING ERRORS
       A(I,I) = 1.0
   40  CONTINUE
       WRITE(*,60) ((B(I,J),J=1,NSTAT),I=1,NSTAT)
   60  FORMAT(1X,7F10.6)
       RETURN
       END
```