OPTIMAL POLICY FOR PREVENTIVE MAINTENANCE SCHEDULING OF THERMAL POWER PLANT USING FUZZY LOGIC

A DISSERTATION

submitted in partial fulfillment of the
requirements for the award of the degree
of
MASTER OF ENGINEERING
in
WATER RESOURCES DEVELOPMENT

By
HARIS ZULKARNAIN

WATER RESOURCES DEVELOPMENT TRAINING CENTRE
UNIVERSITY OF ROORKEE
ROORKEE-247 667 (INDIA)
December 2000

CANDIDATES DECLARATION

I hereby declare that the Dissertation "**Optimal Policy For Preventive Maintenance Scheduling of Thermal Power Plant Using Fuzzy Logic**" being in partial fulfillment of the requirements for the award of the degree of Master of Engineering in **Hydro Electric System Engineering and Management** at Water Resources Development Training Center, University of Roorkee, is an authentic record of my own work carried out during period July 17, 2000 to December 12, 2000 under the supervision of Prof. Devadutta Das, Director WRDTC, and DR. N.P. Padhi, Department of Electrical Engineering University of Roorkee, Roorkee, India.

The matter embodied in this Dissertation has not been submitted by me for the award of any other degree diploma

Date : 12/12/2000
Place : Roorkee

Haris Zulkarnain

It is certified that above statement made by the candidate is correct to the best of my knowledge.

12/12/2K

DR. N.P. Padhy
Department of Electrical Engineering
University of Roorkee
Roorkee 247667
I N D I A

Prof. Devadutta. Das
Director WRDTC
University Of Roorkee
Roorkee 247667
I N D I A

ACKNOWLEDGEMENT

I wish to express my very deep of gratitude and very sincere acknowledgment to DR. N.P. Padhi, Department of Electrical and Prof. Devadutta Das, Director WRDTC University of Roorkee for his continued inspiration, valuable guidance, advice and encouragement during the preparation of this dissertation.

I am very much grateful to the faculty members and staff of Water Resources Development Training Center for their inspiration kind cooperation and facilities provided for this work.

I wish to express my whole hearted gratitude to the Ministry of Public Work, Directorate General of Water Resources Development which gave me an opportunity to study M.E. degree at Water Resources Development Training Center, University of Roorkee.

Finally, I am highly indebted to my parent and brother, sister and my families for their moral support and all the help during my absence from Indonesia.

December 12, 2000

Haris Zulkarnain

SYNOPSIS

Thermal power plant may be base load or peaking power plants. It is necessary to meet the load demand at each and every interval, independent of type of power plant. Any loss of load due to break down of any or all the unit of plant do not only result in loss of revenue but also may resulting in threat to security of the system.

Hence an optimal of preventive maintenance policy is proposed so that break down and loss of load can be minimised.

CONTENT

LIST OF TABLE

LIST OF FIGURES

CHAPTER – 1

INTRODUCTION

In developing countries electricity generation capacity is much less than the demand. It is therefore required to use the installed capacity in the best possible way. Thermal power plants shares the major portion of the total generation capacity. It is the responsibility of the power plant managers to evaluate a preventive maintenance policy so that the plant availability can be maximized. Since power plant equipment are all complex and the system as a whole is fairly complex, the problem of finding optimal preventive maintenance schedule for its equipments (sub-systems) is of theoretical interest and practical value.

The main objective of this thesis is to develop practical model for preventive maintenance scheduling for thermal power plants so that the plant availability can be maximized. Special emphasis has given in this study to deal with the situations when any past record of failure and repair time of critical power plant equipments are not available. The study can be broadly divided into two parts.

(i) Formulation o appropriate process to evaluate critical equipments outage/repair distribution from the expert originated information based on Fuzzy set theory.

(ii) Evaluating a proper simulation model to plan preventive maintenance policy or optimizing the availability o thermal power plants.

The chapterwise content are given below :

Chapter – 2 Literature review

Chapter – 3 present the necessity of preventive maintenance choice of model, power plant critical equipment configuration and required assumptions.

Chapter – 4 deals with the problem how to develop relevant data from the expert opinion when no past record is available.

Chapter – 6 gives the logic and flow chart of the computer program developed and discusses and analysis the result.

CHAPTER – II

LITERATURE REVIEW

2.1 Preventive Maintenance

With cost and difficulty of constructing new generating units increasing on improving, utilities are being forced to focus more efforts on improving the productivity of existing units. In simplest terms, the utilities problem is to identify those areas where limited funds are most likely to produce the highest return. Return is usually measured as a reduction in replacement power costs resulting from improved productivity from existing units. One way to optimize the use of betterment funds is,- assessment of options to improve power plant availability [4]. The required availability-goal can be achieved by reliability and maintainability analyses. Chang [5] has given a co-hesive and comprehensive approach to improve power plant availability. Since the object of any betterment program is to reduce the impact of system and unit productivity, mathematical models are needed which predict the productivity impact of various combination of system configuration and component performance.

The purpose of this study is to develop a suitable model of fossil fuel based thermal power plant which takes into account the operating and maintenance policies. This model will help us to understand the impact of preventive maintenance on unit productivity, and hence take the might decision. There were very few studies to find the optimal preventive maintenance schedule for critical equipment of a power plant based on their failure and repair characteristics such that the plant generation is maximized. Das and Acharya [11] has made an attempt to given a simple model for evaluating availability of thermal power plants.

2.2. Indices of Plant Productivity

The productivity of a thermal power plant can be measure by reliability of individual utilities. Measurement of actual reliability provides feedback to planners on the actual performance of the executed plans, and to operations personal on the reliability effect for operating and maintenance practices. Several fundamental indices are proposed in a report by IEEE group on measurement indices [8]. Any one of them can be adopted based on individual utility's approach, requirement, as well as the flexibility to expand on essentially the same maintenance schedule for each individual utility [2].

2.3. System Modeling

Many techniques are available for calculating system unavailability using component failure rates and repair time. These techniques are invariably based on certain assumptions which are not strictly applicable to power plant system. It has been found that for many situation a straight forward simulation of the system (Monte Carlo Analysis) has several advantage [4] among these are :

- Many power plant systems are non Markovian. The Monte Carlo simulation will take account of such things as different failure rates for components in operation and in standby and different repair urgencies depending on whether or not a spare is available.
- The Monte Carlo simulation will handle any distribution of Time Between Failure (TBF) and Time to Restore (TTR) rather than assuming exponential distribution.

Monte Carlo simulation model provides the basic for the development of outage histories that include both random events and the consequences of human decision affecting outage [1].

EPRI (Electric Power Research Institute, Palo Alto) project RP 1534-1,2 reveals that most detailed and accurate modeling of complex operating consideration presently requires the use of Monte Carlo Simulation [6].

ENEL (Italian National Electricity Authority) has been for many years using a Monte Carlo – based program (SICRET) for system planning. This is due to several advantage of the sampling simulation techniques such as high flexibility and detail in the simulation of complex system operation and configuration [7].

Of course the simulation may not always be the most efficient approach and there are always questions of accuracy. However most power plant system are simple enough that can "overkill" the problem while keeping computer costs reasonable. The Monte Carlo Simulation can be used to give a point estimate of system availability or probability distribution.

2.4. Outage Data

There were suggestion to apply probability for evaluation of reliability since 1933. The interest to use probability method took real shape after publication of forced outage data by AIEE subcommittee on "Application of Probability Methods" in 1949. This first publication was followed by two additional report on outage experience in 1954 and in 1957 [2]. Power system equipment outage data are well collected in Western Countries.

A bibliography of equipment outage data is available in [12]. No such endeavor has been taken up in countries like India and Indonesia to collect the equipment outage data and analyze them. It not realistic to use the data competed by organizations of western countries, as the manufacturing and maintenance practice are quite different along with the operating conditions.

The question arise how to evaluate reliability in our context and formulate future planning. Will we start initiating the process of collecting data and wait for the result which may take long time.

One possible solution can be using information originating from expert in the field of reliability, engineers and manufactures. To develop a outage distribution of equipments we can adopt possibilis framework. The main reason for adopting such a framework is tl possibility theory offers a simple theory of uncertainty that explicitly ta into account the lack of precision of expert knowledge. The whole ta can be divided into three parts, collection information from exper assessment of experts and clustering of data supplied by expert. Expe can be evaluated in term of calibration and level of precision, respectiv measured by membership grades and fuzzy cardinality [9]. The cluster of data can be done by possibility theory such that objects within same cluster have a high degree of similarity with expert to precision a accuracy [10].

2.5. Objective

It is revealed from the review of literature, among many models evaluate reliability of thermal power plant, Monte Carlo Simulat Method is the most preferred one, for its flexibility and adaptability in situations.

Further it is more realistic to gather, access and cluster hum originated information in possibility framework which account for inevitable uncertainty of the information.

CHAPTER – III

PREVENTIVE MAINTENANCE OF THERMAL POWER

3.1 PREVENTIVE MAINTENANCE

Preventive maintenance policies consist of some action based upon either the operating age of certain components in the system or the state of system degradation. In the first case, a preventive maintenance policy usually consists of some program for the planned replacement of certain critical components after they have accumulated a given number of operating hours. In the second case, the preventive maintenance policies are designed to minimize the time the system will spend in degraded states.

Under certain preventive maintenance policies it may be possible either to increase an equipment's availability or reliability (probability of survival) or to minimize the total cost of replacements. When components exhibit a constant failure rate preventive maintenance policies cannot be justified because it is equally as likely that the component will fail in the next interval of time whether or not it is replaced with a new one.

Basically, a planned replacement policy involves the choice of when to replace the components assuming they have not failed. The choice of a schedule depends primarily upon the measure of reliability effectiveness chosen. Preventive maintenance policy can be adopted on the basis of one or more of the following reasons.

(i) *Probability of Survival* : A preventive maintenance policy is justified when the component exhibits an increasing failure rate. Since the measure is concerned with the probability of failure free operation over a given time interval, replacing a component that has an operating age x with a new one returns the failure rate to the initial value (at time zero). In effect a preventive maintenance policy changes the failure law of the component.

(ii) *Availability* : A preventive maintenance policy is justified when the component exhibits an increasing, failure rate with time and when the replacement time of components that have not failed is less than the replacement time of failed components. The reason for the last qualification is that each maintenance action-preventive or corrective – induces downtime. Thus availability may be enhanced by substituting preventive maintenance time for corrective maintenance time. In this case, preventive maintenance reduces the number of failures by reducing the operating time of each component.

(iii) *Total Cost of Replacements* : A preventive maintenance policy is justified when the component exhibits an increasing failure rate and when the cost of replacement of a component that has not failed is less than the cost of replacing a failed component. The reason for this last qualification is that the total cost of replacement is made up of the cost of replacing a "good" component plus the cost of replacing the failed components. Since a preventive maintenance policy reduces the number of component failures by reducing their operating time, it also reduces the total cost of failure replacement. In this case the best policy is evaluated as a matter of economics. A balance must be struck between the expense due to planned replacement and the expense due to failures, so that the total cost is minimized.

3.2 PREVENTIVE MAINTENANCE OF THERMAL POWER PLANT

The main objective of adopting a preventive maintenance policy for a thermal power plant is to increase the availability of power plant. Though it can also be adopted on the basis of economics, as mentioned earlier but that in our study we will look upon the problem only from the availability point of view and do not consider economics.

The following opportunistic maintenance policy is followed in our study. A subsystem/unit X is repaired on failure. Further, preventive maintenance is done for X; if it is in continuous operation for at least T_1 periods, during repair of another subsystem/unit Y. In addition to above, preventive maintenance is also done for units of a sub-system having standby redundant units, after they are in continuous operation for T_2 periods.

Thus, T_1 is the lower limit and T_2 is the upper limit of the age at preventive maintenance for the units of a sub-system having standby redundant units. The sub-system which do not have redundant units (boiler control, turbine bearing etc.). are preventively maintained (at an age ≥ T) only when another subsystem/unit is under repair.

After making the above assumptions the objective is to evaluate a suitable process to determine the lower limit T_1 and upper limit T_2 of the age of each equipment as described above so that we can achieve maximum reliability or in other words plant availability is maximized.

3.3 RELIABILITY EVALUATION METHOD

There are many analytical methods of reliability evaluation. In these methods the life process of a component or a system is described by a mathematical model and the required reliability indices are provided by the solution of this model. Power plant equipments (e.g. boiler, pulverizes, feed pumps, and fans, etc) are all complex and the failure, repair etc, times of these equipments are not exponential. The non-exponential failure, repair times of the dependent equipments leads to mathematical complexity of the model that prevent its solution. Even if the analytical part is manageable the shear size of the computations and of the computer time involved can be prohibitive.

Another method of system reliability evaluation is Monte Carlo method. In the Monte Carlo approach an actual realization of the process is simulated on the computer and, after having observed the simulated process for some time, estimates are made of the desired reliability indices.

Advantages of the Monte Carlo simulation include the following :

- There are no restrictions on the failure and other time distributions in the system.
- Dependent relations between the failure, repair, etc. events can be easily accounted for.
- The analytical work involved is simple.
- Short-term solutions can be easily obtained.
- System additions can be easily incorporated in the study.

The difficulty to apply Monte Carlo method lies in the prohibitive computing time whenever a very rare event has to be shown.

This only difficulty of Monte Carlo method is out weighed by its advantages and we adopt this method in our study.

3.4 MONTE CARLO SIMULATION

This simulation is treated as a series of real experiments. During its course, events are made to occur at times determined by random processes obeying predetermined probability distributions.

One of the central problems in the Monte Carlo method is timing of the various events in the simulated process, in accordance with these distributions. The simplest way to do this for a given event is randomly selecting a number from a large set of numbers possessing the appropriate distribution and making the event 'occur' at the moment indicated by the number chosen. This method would require the generations and storage of several sets of numbers with distributions corresponding to all the time distributions in the process. Matters can be simplified by using a single set, where the numbers are uniformly distributed between the values 0 and 1. The random selection of a number from this set can be simply converted into the selection of a number from a set with an arbitrary distribution, using the CDF (commutative distribution function) of the latter. This is explained in the following.

Consider a random variable T with the CDF $F_T(t)$. with each value t that T can assume let a value u be associated such that $u = F_T(t)$. This set of u values then defines a random variable u which depends on T as shown in the Fig. 3.1.

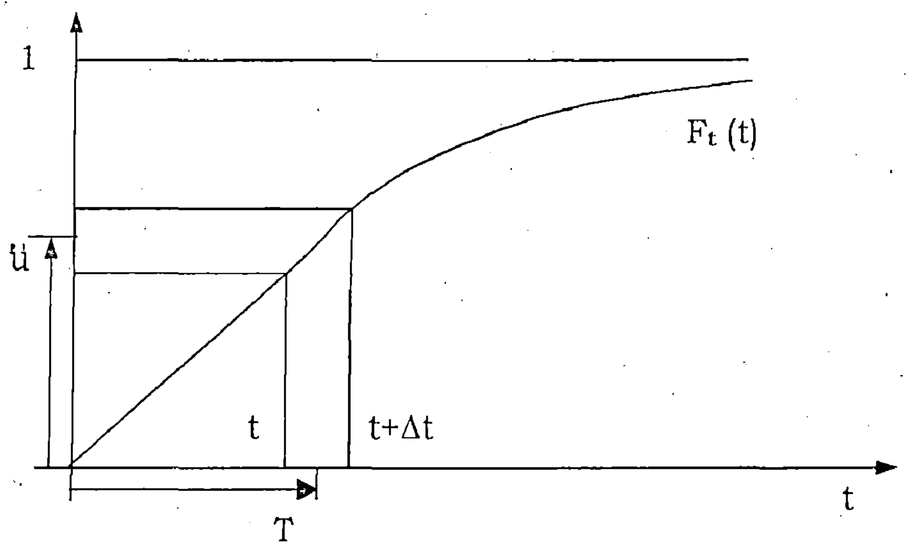


Fig. 3.1. The Random Variable T and the Associated Random Variable u with Uniform Distribution.

The distribution of U can be determined as follows. By the above definition.

$$P[u < U \leq u + \Delta u] = P[t < T << t + \Delta t] \quad (3.1)$$

where, since $F_T(t)$ is the CDF of T,

$$P[t < T \leq t + \Delta t] = F_T(t + \Delta t) - F(t) \quad (3.2)$$

By Fig. 3.1, however, the right hand side of (3.1) equals Δu and therefore, by combining (3.1) and (3.2} one obtains :

$$P[u < U \leq u + \Delta u] = \Delta u. \quad (3.3)$$

This result indicates that u has a uniform distribution between 0 and 1 (or more formally, $f_u(u) = 1, 0 < u \leq 1$). It follows that if one randomly selects a value u from among a set of numbers uniformly distributed in the range (0, 1) and computes t from.

$$t = F_T^{-1}(u) \quad (3.4)$$

where $F_T^{-1}(u)$ is the inverse function of $F_T(u)$, the t values will form a set with the CDF $F_T(t)$.

Returning to the simulated realization of the life history of a system, one proceeds by creating separate life histories for all the components, and examining all the system failure. Each time a system failure is encountered, its duration is registered, and the failure count is advanced by one.

3.5 SYSTEM CONFIGURATION AND ASSUMPTIONS

For the simulation models to find the optimal preventive maintenance intervals let us take a typical thermal power plant critical equipment configuration as shown in the Fig. 3.2

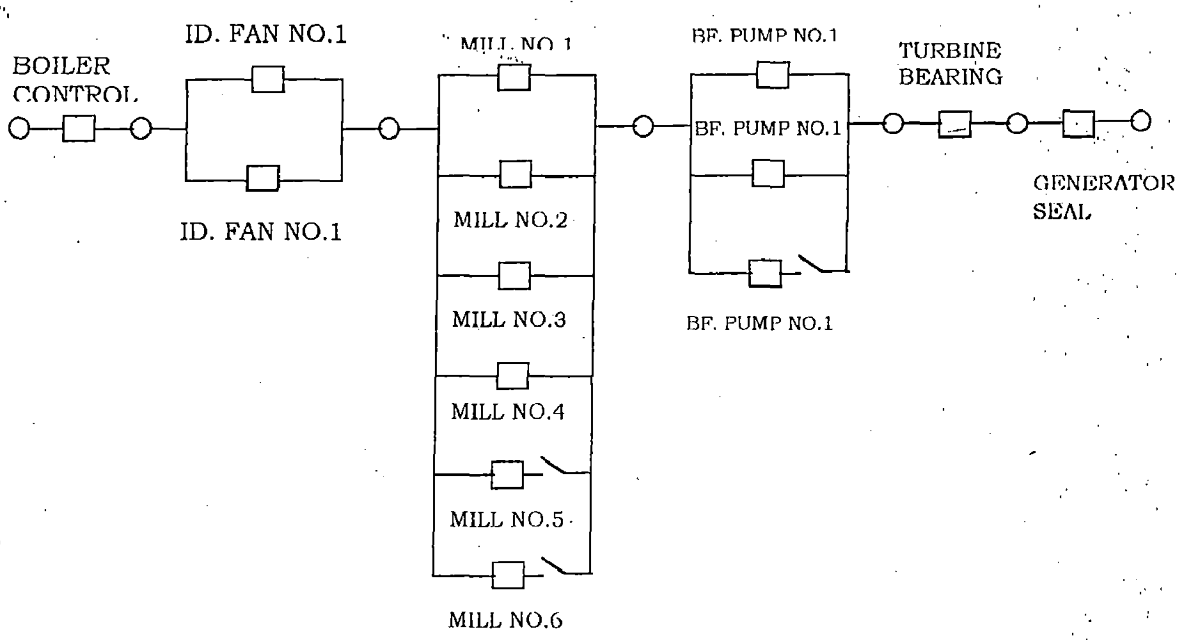


Fig. 3.2 : Block Diagram of Critical Equipment/Component of Thermal Power Plant.

For successful working of the plant as shown in Fig. (3.2), it is necessary to ensure satisfactory operation of the boiler control, the turbine bearings and the generator seals, two I.D. Fans, four out of six mills and two out of three boiler feed pumps. While the plant is shutdown in the events of boiler control failures, generator seal failures and turbine bearing failures, the plant is allowed to run at half load if an I.D. fan fails. To match with the reduced loads, two of the four running mills and one of the running feed pumps are also stopped. Similarly, in situations where either one feed pump or two mills are working satisfactorily, the plant is operated at half load. However, in these situations, both I.D. fans are kept running. This practice is often followed to maintain correct draft in the furnace.

In the models to follow, it is assumed that :

(i) A mill or a pump does not fail while it is on standby and its switching to operating mode is perfect.

(ii) After a repair or preventive maintenance, the unit is brought back to as good as new state.

(iii) The failure process of one unit is independent of the other units. Plant output is unity except shutdown and partial shutdown.

3.6 REALIBILITY INDEX

The reliability of the thermal power plant for different preventive maintenance interval is evaluated by unit availability (Au).

The unit availability is given by the following equation (3.5)

$$A_u = 1 - \frac{TSDT + 0.5 \times PSDT}{MST} \qquad \text{.......................................(3.5)}$$

Where,

A_u = Availability of one unit;

TSDT = total shut down time

PSDT = Partial shutdown time

MST = Maximum simulation time.

3.7 RELEVANT DATA

The failure and repair time data can be obtained from the data bank of similar types of plant with similar configuration. But in reality systematic records/history of equipments are not maintained and hence it is necessary to generate failure and repair data of equipments from experts. The procedure is described in the next chapter.

CHAPTER – IV

MAINTENANCE SCHEDULING APPROACH USING FUZZY SET THEORY

4.1. DISTRIBUTION PARAMETERS FOR FAILURE AND REPAIR TIME

These can be obtained from the history of the equipments kept systematically for similar type of thermal plant. After collecting the unit equipment history for time to failure and for time to repair completion the statistical patterns and corresponding parameters can be found out. But following the distribution pattern of equipment's manufactured years back (that too may be manufactured by some other manufactures) and operating in different environment does not seem to be realistic approach. Moreover in most of the case the records are not available. Hence the use of information originating from human experts in the field of reliability and safety analysis of newly designed installations or regarding process on which no experimental observations are possible becomes more and more and more accepted by scientific community. The uncertainty model play a central role in the use of expert judgements, because no human being can be absolutely sure about his judgement or advice. It is therefore necessary to incorporate into any model the individual experts uncertainty about his advice, the decision makers uncertainty about the quality of the experts, and how these two kinds of uncertainty interact and impact on the credibility of the final results.

4.2. Fuzzy Sets

The Characteristic function of a crisp set assign a value of either 1 or 0 to each individual in the universal set, thereby discriminating between members and nonmembers of the crisp set under considerations. This function can be generalized such that the value assigned to the elements of the universal set fall within a specified range and indicate the membership grade of these elements in the set in question. Larger value denote higher degrees of set membership. Such a function is called a membership function, and the set defined by it a fuzzy set.

The most commonly used range of value of membership functions is the unit interval [0,1]. In this case, each membership function maps element of a given universal set X, which is always a crisp set, into real members in [0,1].

The membership function of a Fuzzy set a is denoted by π_A : that is,

$$\pi_A : X \longrightarrow [0,1]$$

Each Fuzzy set is completely and uniquely defined by one particular membership function ; consequently, symbols of membership functions may also be used as labels of the associated Fuzzy sets.

Fuzzy also allow us to represent vague concepts expressed in natural language. The representation depends not only on the concept, but also on the context in which it is used. For example, applying the concept of high temperature in one context to weather an in another context to a nuclear reactor would necessarily be represented by different Fuzzy Sets. That would also be case, although to a lesser degree if the concept were applied to weather in different seasons, at least in some climates.

Several Fuzzy sets representing linguistic concept such as low, medium, high and so on are often employed to define state of a variable. Such a variable is called Fuzzy Variable. The significance of Fuzzy variables is that they facilitate gradual transitions between state and, consequently, possess a natural capability to expressed and deal with observation and measurement uncertainties. Traditional variables, which we may refer to as crisp variables, do not have this capability.

Since Fuzzy variable capture measurement uncertainties as part of experimental data, they are more attuned to really than crisp variables. It is an interesting paradox that data base on Fuzzy variables provides us, in fact, with more accurate evidence about real phenomenon that data based upon crisp variables. This important point can hardly be expressed better than by the following statement made by Albert

Einstein in 1921 : *So far as laws of mathematics refer to reality, they are not certain, and so far as they are certain, they do not refer to reality.*

4.3. POSSIBILITY THEORY, THE BASIC FRAMEWORKS

The uncertainty can be modeled using the classical and Bayesian approaches but possibility theory offers a simple theory of uncertainty that explicitly takes into account the lack of precision of the expert knowledge which is the main reason for adopting such a framework. A probability distribution never accounts for a lack of precision in the data, and so the possibilistic model is more faithful to the available data supplied by experts.

To get useful information from the experts, several problems must be solved. The first one is a proper modeling of expert knowledge about numerical parameters in the frameworks of possibility theory, which is more natural than a pure probabilistic model.

The second task to be solved is the assessment of the quality of the expert, namely his calibration and the precision of his response. This assessment evaluation is carried out in terms of calibration and level of precision, respectively, measured by membership grades and fuzzy cardinality indexes. Last when several expert responses are available, they may be combined so as to yield a unique, hopefully better response. The probabilistic framework looks somewhat restrictive to express the variety of possible pooling modes. Hence various pooling modes with their formal model under various assumptions concerning the experts based on possibility theory.

4.4. ELICITATION OF EXPERT KNOWLEDGE

The simplest model of a family of probability distributions is offered by possibility theory. A possibility distribution π_v attached to parameter v can be viewed as the membership function of the fuzzy set of possible values of a variable v. The possible values as described by π_v are assumed to mutually exclusive, since v takes on only one value (its true value) from a set x taken here to be closed, bounded real interval $[x_l, x_u]$. Moreover, since one of the elements of x is the true value of v, $\pi_v(x) = 1$ for at least one value $x \in X$. Possibility distributions, can be rigorously related to probability distributions, in which case $\pi_v(x)$ is taken to be an upper probability bound.

The simplest form of a possibility distribution on x is the characteristic function of a subinterval $[s_l, s_u]$ of x , i.e., $\pi_v(x) = 1$ if $x \in [s_l, s_v]$, o otherwise. This type of possibility distribution results when experts claim that "v lies between s_l and s_v"(Not that $\pi_v(x) = 1$ has a weaker meaning than in probability theory, it only means that x is a completely possible value for v). This way of expressing knowledge is more natural than giving a point value, say x^*, for v right away, because it allows for some imprecision ; (the true value of v is more likely to lie between s_l and s_u than to be equal to x^*). clearly, allowing for imprecision reduces the uncertainty of the assessment. Indeed imprecise statements are always safer than precis ones.

The representation, however is not entirely satisfactory. Namely, claiming that $\pi_v(x) = 0$ for some x means that $v = x$ is impossible, a very strong statement. This is too strong for the expert who is then tempted to give wide, uninformative intervals (e.g., $s_l = x_L$, $s_v = x_v$). It is more satisfactory in this connection, to obtain from the expert several nested intervals with various levels of confidence and to admit that even the widest, saflest intervals contain some residual uncertainty, here denoted ε. These nested intervals will lead to membership functions of fuzzy intervals.

A fuzzy interval can be viewed as a finite set of nested (local) subsets $\{A_1, A_2, \ldots A_m\}$ as long as the set of possibility values $\{\pi_v(x) \mid x \in X\}$ is finite.In this case , there is a set of weights $p_1, p_2, \ldots p_m$ summing to one, such that

$$\forall x,\ \pi_v(x) = \sum_{x \in Ai} p_i \quad \ldots\ldots(4.1)$$

Namely it can be proved that if the set of possibility values is $\{\alpha_1=1, \alpha_2 \geq \alpha_3 \geq \ldots\ldots \geq \alpha_m\}$, and letting $\alpha_{m+1} = 0$ we have

$A_i = \{x \mid R_v(x) \geq \alpha_i\}$; ....(4.2)

$P_i = \alpha_i - \alpha_{i+1} \qquad 1 \leq i \leq m$

Knowing a possibility distribution, the likelihood of events can be described by means of two set- functions. The possibility measure (Π) and the necessity measure (N). When Π is the membership function of a crisp set A given as the evidence, an event B is said to be possible if and only if $A \cap B \# 0$, and certain if and only $A \subseteq B$; by definition we let Π (B) = 1 and N(B) = 1 in these respective situations. Letting Π_i and N_i be the {0, 1} – valued possibility and necessity measure induced by the set Ai, define it can be defined

$$\pi(B) = \sum_{i=1,m} p_i \Pi_i(B) = \sup_{x \in A} \pi_v(x) \quad \text{.................................. (4.3)}$$

$$N(B) = \sum_{i=1,m} p_i N_i(B) = \inf_{x \notin A} (1 - \pi_v(x)) \quad \text{.................................. (4.4)}$$

$$= 1 - \Pi(\overline{B})$$

where B is the complement of B with respect to X. This duality expresses the fact that B tends towards certainty as B tends towards impossibility.

The expert is supposed to be capable of supplying several intervals A_1, ..., A_m directly, corresponding to prescribed levels of confidence λ_1,......... λ_m. The level of confidence λ_1 .can be conveniently interpreted as the smallest probability that the true value of v hits A_i (e.g., from the point of view of experts, the proportion of cases where $v \in A_i$ from his experience). In practice, only three intervals have been kept : A_1 with $\lambda_1 = 0.05$, A_2 with $\lambda_2 = 0.5$ and A_3 with $\lambda_3 = 0.95$. A_1 corresponds to usual values of v, and $A_3 = [s_l, s_u]$ corresponds to the interval which leaves a 0.05 probability (=ε) that v misses A_3, i.e., the residual uncertainty of the conservative evaluation.

The links between λ_I's and the degrees of possibility are defined by $\lambda_i = 1 - \alpha_{i+1}$ for i=1, m, i.e., the degree of possibility α_{i+1} is related to the degree of certainty (λ_i) that x lies in A_i; this degree of certainty being interpreted as a lower bound on the probability $P(A_i)$. In the terminology of possibility theory, $\lambda_i = N(A_i)$ the degree of necessity of A_i. Finally, the focal subset $A_m = A_4$ is always x itself, due to the residual uncertainty. The following Table 4.1 summarizes the data supplied by one expert.

Table 4.1 : Data Supplied by Exerts ($_1$, s_u, m_l, m_u, c_1, c_u)

(in the bold Faced Rectangle)

A_1	$[c_1, c_u]$	00.05	1	0.05
A_2	$[m_l, m_u]$	0.5	0.95	0.45
A_3	$[s_l, s_u]$	0.95	0.5	0.45
A_4	X	1	0.05	0.05

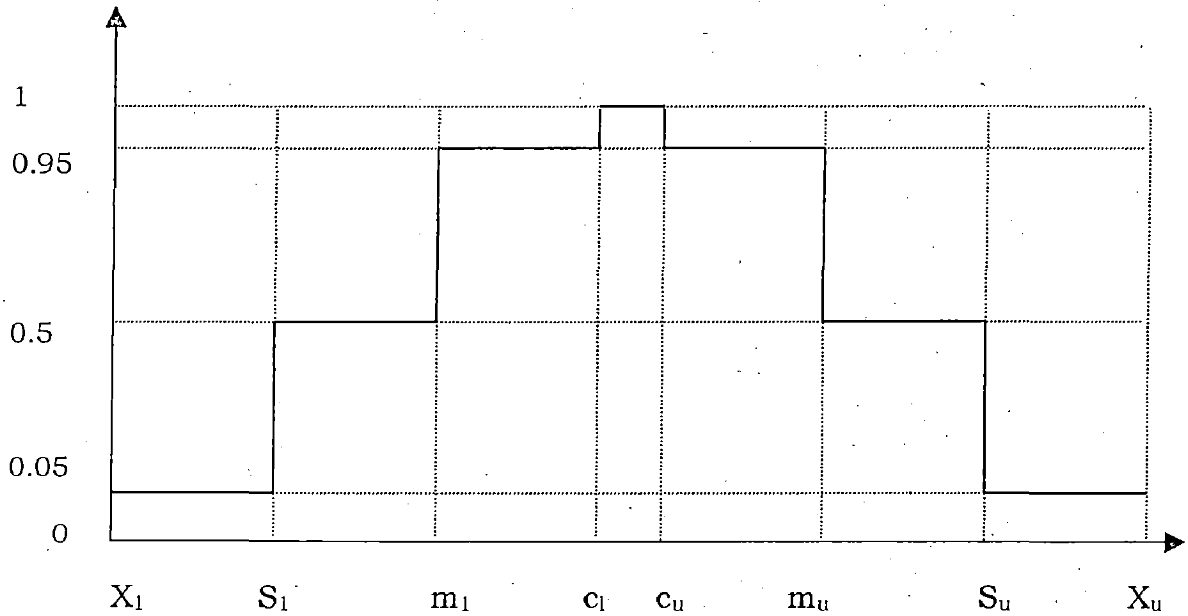


Fig. 4.1 : Expert – Originated Possibility Distribution

The first three lines of Table 4.1 correspond to specific question asked to experts. Although intervals $[c_l,c_u]$, $[m_l, m_u]$, $[s_l, s_u]$ are not used in the probabilistic approaches, these intervals can be interpreted in terms of quantities of a probability distribution, (e.g., $[s_l, s_u]$ corresponds to the range between the 2.5% and the 97.5% quintiles

The nestedness property of the supplied intervals presupposes that the expert, although having imprecise knowledge, give coherent answers to various questions.

4.5. ASSESSMENTS OF EXPERTS

Once the possibility distributions of the uncertain variables under consideration have been determined the next step is identifying the type of deficiencies experts may be prone to and then defining indexes that enables to build a meaningful rating system for the experts. Experts can be deficient with regard to three aspects:

- *Inaccuracy*: Value given by the expert are inconsistent with the real values of the parameters, for instance underestimated. The expert is then said to be miscalibrated.
- *Imprecision:* the expert through not miscalibrated is too cautious. So, the intervals he supplies are too large to informative. Such an expert is said to be underconfident.
- *Exaggerated Precession:* the value of the parameters is not precisely known but the expert supplies intervals that are too narrow (or even point values). Such an expert is said to be overconfident.

The deficiencies cited above can be treated in the both probabilistic and possibilistic framework. In probabilistic framework, the concept of an individual calibration measure for each variable does not exist. As a result, no individual quality, measure can be obtained. This lack of individual measures may lead to distortions and represents the major inconvenient of this method. As an example, it may happen that a source which given precise information only when it is inaccurate, and accurate information only when it is imprecise, is considered to be good.

4.6. THE POSSIBILLISITC APPROACH

To build scoring indexes that reflects these issues in the possibilistic frame works, let us first consider a seed variable v whose value x is precisely known, and let E be the fuzzy set supplied by an expert, e, to describe his knowledge about v. let μ_E be the membership function of E (so that $\mu_E = \mu_v$). In this situation over confidence cannot arise. It is easy to see that.

- The greater $\mu_E(x^*)$, the more accurate is the expert. Indeed if $\mu_E(x^*) = 0$, E totally misses X^* while if $\mu_E(x^*) = 1$, x^* is acknowledged as a usual value of v. Hence, a natural value of accuracy is given by

  $$A(e,v) = \mu_E(x^*) \quad \text{..(4.5)}$$

- If E is a crisp interval [a,b] the wider E, the more imprecise (hence under confidant) the expert. The width of E the then $|E| = b\text{-}a$. When E is fuzzy the width of E is generalized by

  $$|E| = \int_X \mu_E(v)dv. \quad \text{..(4.6)}$$

This is a generalized fuzzy cardinality, where E is a finite nested random set,

$$|E| = \sum_{i=1,m} |A_i| \, p_i \quad \text{..(4.7)}$$

This evaluation my re-scaled so as to account for the residual uncertainty E and so that it yields one when $s_l = s_u$ (precise response for which is $|E| = \varepsilon . |X|$) and when $s_l = x_u$, $s_u = x_u$ (empty response. A reasonable specificity index is then

$$S_p(e,v) = f(|E|) = \frac{|X| - |E|}{(1-\varepsilon)|X|} \quad \text{.........................(4.8)}$$

On the whole, the overall rating of the expert with respect to a single seed variable can be defined as

$$Q(e,v) = A(e,v).\ S_p(e,v) \quad \text{...(4.9)}$$

which requires him to be both accurate and informative to score high.

When the seed variable is not precisely known, the index Q(e,v) can be extended as follows:

- If the actual value of seed variable value is described by a histogram leading to probability distribution P then

  $$Q(e,v) = P(E).\ S_p(e,v) \quad \text{.......................................(4.10)}$$

  where P(E) is the probability for the fuzzy event E i.e.

  $$P(E) = \int_X \mu_E(v)\ dP(v) \text{...(4.11)}$$

- If the actual value of a seed variable is described by a possibility distribution $\pi_\upsilon^* = \mu_F$ then

  $$Q(e,v) = \Pi^*(E).\ f(|E \Delta F|) \quad \text{.................................(4.12)}$$

where Π^* is the possibility measure attached to π^*_v and Δ is the symmetric difference of fuzzy sets.

Global measures of accuracy, precision and quality to an expert e can be obtained using the simple arithmetic mean over the individual scores. If m is the total number of seed variables, then

$$A(e) = \frac{1}{m} \sum_{j=1,m} A(e, v), \quad \text{...(4.13)}$$

$$S_p(e) = \frac{1}{m} \sum_{j=1,m} S_p(e, v), \quad \text{...(4.14)}$$

$$Q(e) = \frac{1}{m} \sum_{j=1,m} Q(e, v), \quad \text{...(4.15)}$$

It is important to note that generally

$$Q(e) \neq A(e).S_p(e) \quad \text{...(4.16)}$$

Thus an expert e is rated by the set {Q(e,v) 1_j = 1, m} of evaluations. Ranking of experts can be based on the average rating of each expert. The standard deviation is also useful to check the significance of the gaps between average rating of experts. Based on this evaluations a set K of experts can be divided into groups of equal reliability. Moreover, the fuzzy set R of reliable experts can be defied by the membership function.

$$\mu_r(ei) = Q(e_i),\ i = 1, \ldots\ldots, k \quad \text{.................................(4.17)}$$

If there are k experts the cardinality of R, say

$$|R| = \Sigma_{i=1,k}\, \mu_R(ei) \quad \text{...(4.18)}$$

gives a good idea of the number of reliable experts in the group.

Let us describe the assessment step with a simple example. There be 10 seed variables, and two experts who give estimations in the form of

5%, 50% and 95% quintiles of a subjective distribution, i.e., $q(e_i, v_j)$ = ($q_{5\%}$, $q_{50\%}$ and $q_{95\%}$). The true value of each seed variable is given in the form of a real number. The data of the simple example is summarized as follows :

Number of experts : n = 2

Number of test variables : m = 10

Variable domain : $[x_l, x_u] (v_j) = [0, 10], 1 \le j \le 10$

Real value of variables :

$x^* (v_1) = 2.5$

$x^* (v_2) = x^* (v_3) = x^* (v_4) = x^* (v_5) = 3.5$

$x^* (v_6) = x^* (v_7) = x^* (v_8) = x^* (v_9) = 4.5$

$x^* (v_{10}) = 7.5$

Input expert e_1 : $q(e_1, v_j) = (1, 4, 8), \quad 1 \le j \le 10$

Input expert e_2 : $q(e_2, v_j) = (3, 4, 7), \quad 1 \le j \le 10$

In Fig. 4.2. we illustrate in a condensed manner the estimations given by the experts e_1 and e_2 for variables, v_1 to v_{10}, as well as the variables true values.

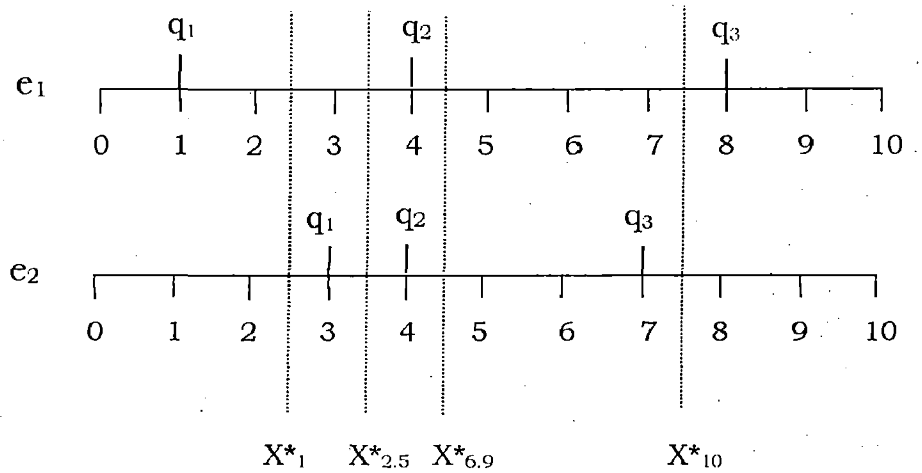


Fig. 4.2 Experts Estimations and True Values of Variables

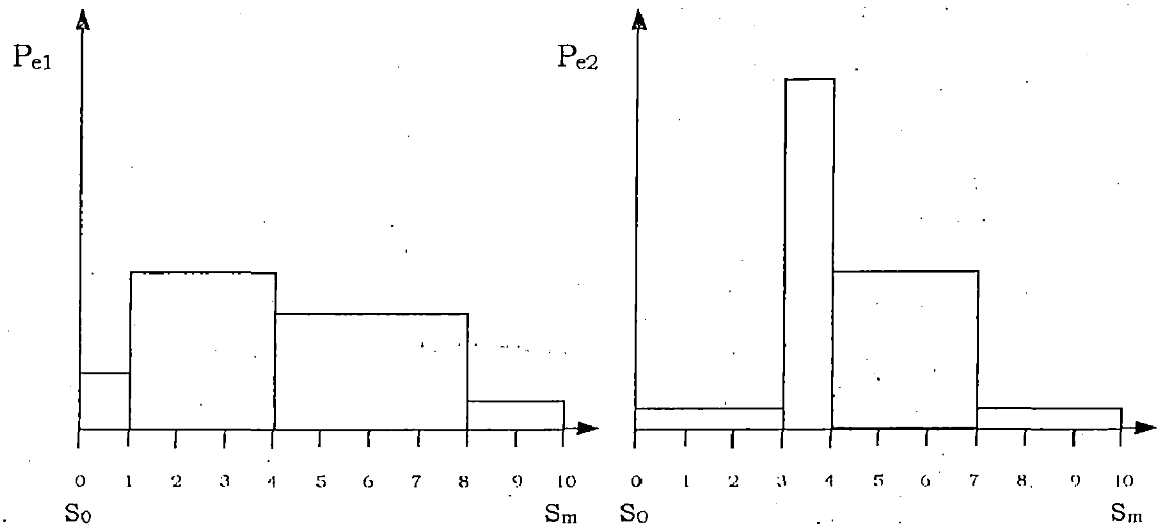


Fig. 4.3. PDF Constructed with the 5%, 50% and 95% quantities

Let us verify the assessment of expert, e_1 in relation to variable v_6 : his accuracy is $A(e_1, v_6) = \pi e_1, v_6\ (4.5) = 0.55$, his precision is

$S_p\ (e_1, v_6) = 1-(0.1 + 3 \times 1 + 4 \times 0.55 + 2 \times 0.05)/10 = 0.46$

and is quality is $Q(e_1, v_6) = 0.55 \times 0.46 = 0.253$.

In Table 4.2 use mean and standard deviation for each experts assessments over the total set of seed variables. We can see that, considering the whole set of seed variables, even if expert e_1 is more accurate than expert e_2, his low performance on precision leads us to consider him less "good" than expert e_2

Table 4.2a: Mean of Experts Assessment (Global Measures)

	Possibilities Evaluation		
Experts	$A(e_1)$	$S_p(e_i)$	$Q(e_i)$
e_2	0.64	0.675	0.432
e_1	0.775	0.46	0.3565

Table 4.2b : Standard Deviation of Experts Assessment

	$\sigma_A(e_j)$	$\sigma_{Sp}(e_j)$	$\sigma_o(e_j)$
e_2	0.3828	0	0.3349
e_1	0.225	0	0.1012

4.7. THE POOLING OF EXPERT JUDGMENTS

The basic principle of the possiblistic approach to the pooling of expert judgements is that there is no unique mode of combination that fits all situations: the choice of combination mode depends on an assumption about the reliability of experts, as formulated by the analyst. No a priori knowledge about the variable understudy is needed, and the experts are viewed as a set of parallel sources to be combined in a symmetric way only if all experts are equally reliable. There are basically two extreme modes of symmetric combination, the conjunctive modes when all experts agree and are reliable, and the disjunctive mode when experts disagree and at least one of them is considered to be reliable. A third mode of symmetric combination is averaging, which considers the experts opinions in a more statistical way. In the case of expert knowledge, the pooling mode depends upon the result of assessment step and the extent to which expert responses on the inquired variable agree with one another.

Conjunctive Mode : Let π_i be the possibility distribution supplied by expert i, for $i \in k$. If all the experts are considered to be reliable (e.g., all the ratings $\mu_R(i)$ are high) then the response of the group of experts is defined by

$$\pi_c(x) = \min_{i \in k} \pi_i(x) \quad \text{...(4.19)}$$

This modes makes sense if all the π_i overlap significantly.

Disjunctive Mode : A rather cautious optimistic assumption about a group of experts is that one expert is right, but it is not known which. This assumption corresponds to the following aggregation

$$\pi_D(x) = \max_{i \in k} \pi_i(x) \quad \text{...(4.20)}$$

This is a very conservative pooling mode that allows for contradiction among experts but may not lead to an informative result, although not necessarily a vacuous one either. Note that if the reliability of experts is unknown and that it is not even certain that one of them is right, then the only pooling method that remain is to look for consensus among experts outliers.

Averaging Mode : This mode corresponds to viewing experts as random source and hence potentially unreliable. Values of the parameters that experts agree are possible are considered more plausible than values that most expert reject.

$$\pi_A(x) = \frac{1}{K} \sum_{i \in K} \pi_i(x) \quad \text{..(4.21)}$$

Note that this value is normalized only if the conjunctive rule gives a normalized result. The lack of normalization indicates that the experts may be wrong. The two modes of renormalization still apply, if this option is ruled out. Generally in the case of disagreement among experts, a multimodal possibility distribution is obtained as with the disjunctive mode.

Consistency-Based Trade Offs : A way to trade-off between the conjunctive and disjunctive modes of pooling is to use a measure c of conflict between two experts and to define

$\pi_T(x) = c \max(\pi_1, \pi_2) + (1-c) \min((\pi_1, \pi_2)$ ....(4.22)

This index gives the conjunctive (disjunctive) mode if c=0 (c=1). If easy to define conflict measure between π_1 and π_2 namely

$C = 1 - \text{cons}(\pi_1, \pi_2)$, ...(4.23)

where $\text{cons}(\pi_1, \pi_2) = \sup_x \min[\pi_1(x), \pi_2(x)]$ is the level of consistency between π_1 and π_2.

Priority Aggregation of Expert Opinion : As pointed out earlier, the fuzzy set R of reliable experts is useful to partition the set K of experts into classes k_1, k_2 , ..., kq of equally reliable ones, where k_j corresponds to a higher reliability level than k_j 1 , for $_{j=1,\ldots\ldots q}$. in this case, the symmetric aggregation schemes discussed above can be applied to each classes k_j. The combinations between results obtained from the k_j' s can performed using the following principle, the response of K_2 is used to refine the response of K_1 insofar as it is consistent with it. If π_1 is obtained from k_1 and $\pi 2$ from k_2, the degree of consistency of

π_1 and π_2 is cons (π_1, π_2) = supx min $[\pi_1(X), \pi_2(X)]$ and the following combination rule has been proposed.

$$R_{1\text{-}2} = \min\{\pi_1 \max[\pi_2, 1 - cons(\pi_1, \pi_2)\}] \quad \text{..............................(4.23)}$$

Not that when cons (π_1,π_2)= 0, k_2 contradicts k_1 and the only opinion k_1 s retained ($\pi_{1-2}=\pi_2$). Π_{1-2} can be similarly combined with π_3, $\pi_{(1-2)-3}$ with π_4 and so on.

4.8. TRASFORMATION BETWEEN POSSIBILITY AND PROBABILITY

Let p be a unimodal PDF, and let x_0 be the mode of p. A possibility distribution can be derived from p by applying the transformation T_1

$$T_1 : \pi(x) = \pi(x') = \int_{x_1}^{x} p(v)dv + \int_{x'}^{x_u} p(v)dv \quad \text{.........................(4.24)}$$

where x′ is such that p(x′) = p(x) < (x_0), and there is no y such that x < y < x′, and p(y) < p(x).

Conversely the transformation T_2 can be used to transformation a possibility distribution into a PDF, where T_2 is given by

$$T_2 : p(x) > \int_0^{\pi(x)} \frac{d\alpha}{|A\alpha|} \quad \text{...(4.25)}$$

where $A\alpha = \{x / \pi(x) \geq \alpha\}$. The characteristics of our data allow us to use the discrete equivalent of T_2

$$p(x) = \sum_{i=1}^{n} \frac{\alpha_i - \alpha_{i+1}}{|Ai|} \mu Ai(x) \quad \text{..(4.26)}$$

where A_1............., A_n correspond to $\alpha_1 = 1 > \alpha_2 > \ldots\ldots > \alpha_n > \alpha_{n+1} = 0$, and function $\mu_{Ai}(x)$ is such that $\mu_{Ai}(x) = 1$ when $x \in Ai$ and zero otherwise

We can use the transformation T_2 to transform the possibility distribution obtained after the pooling step to obtain the probability distribution. The probability distribution thus obtained can be used in Monte Carlo Simulation method for evaluation of plant availability.

CHAPTER-V

PROGRAMME DEVELOPMENT FOR SIMULATION

5.1. LOGIC'S AND ASSUMPTION FOR SIMULATION

In the simulation model considered here, we have assumed that unit i can be replaced only,

(i) on failure

or

(ii) if it is in operation for $a_{1i}*m_i$ time units (where a_{1i} is a constant and m_i is the mean time to failure of I) along with any other unit

or

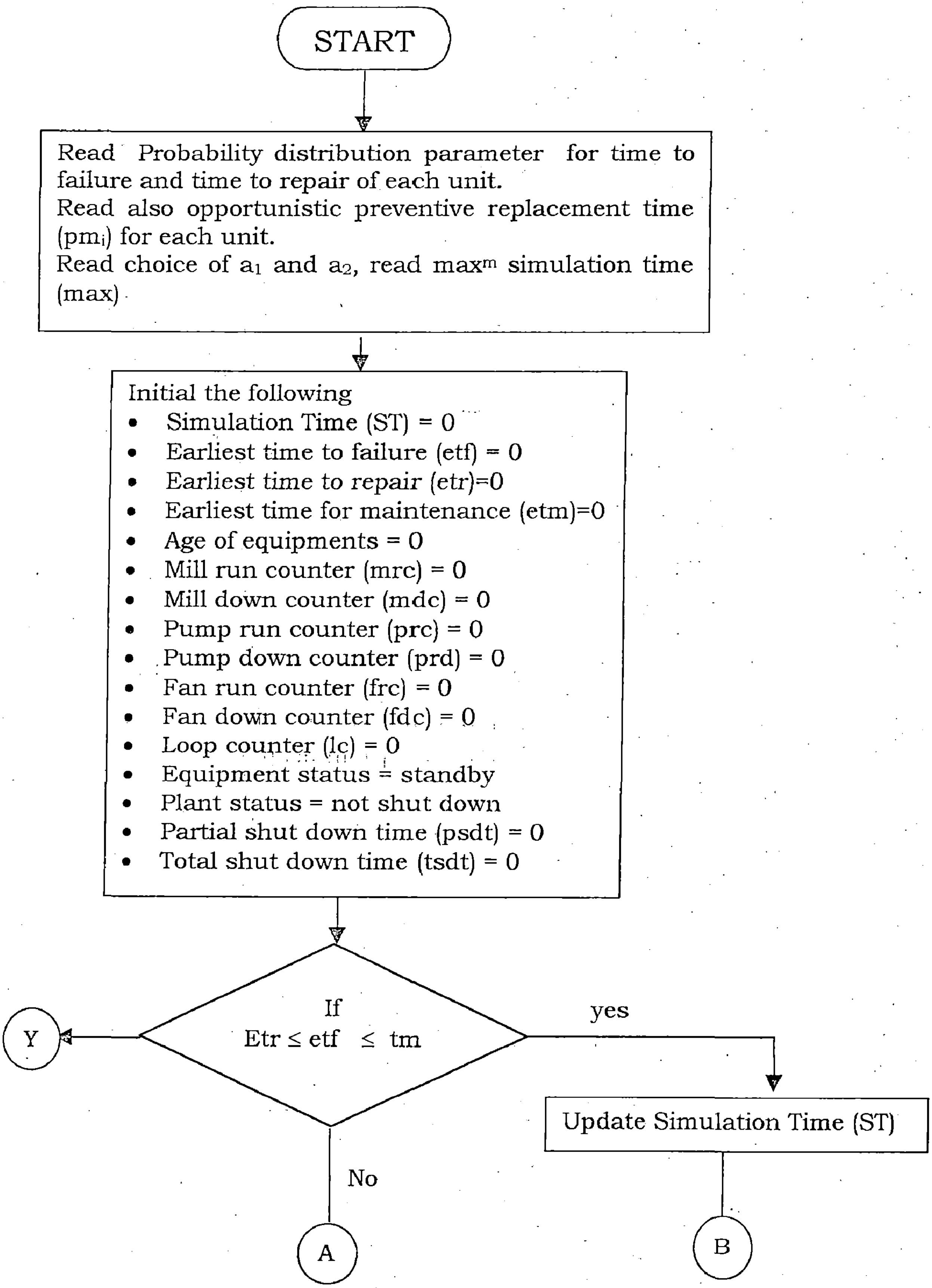
(iii) after $a_{2i}*m_i$ time unit (where a_{2i} is a constant) of continuous operation

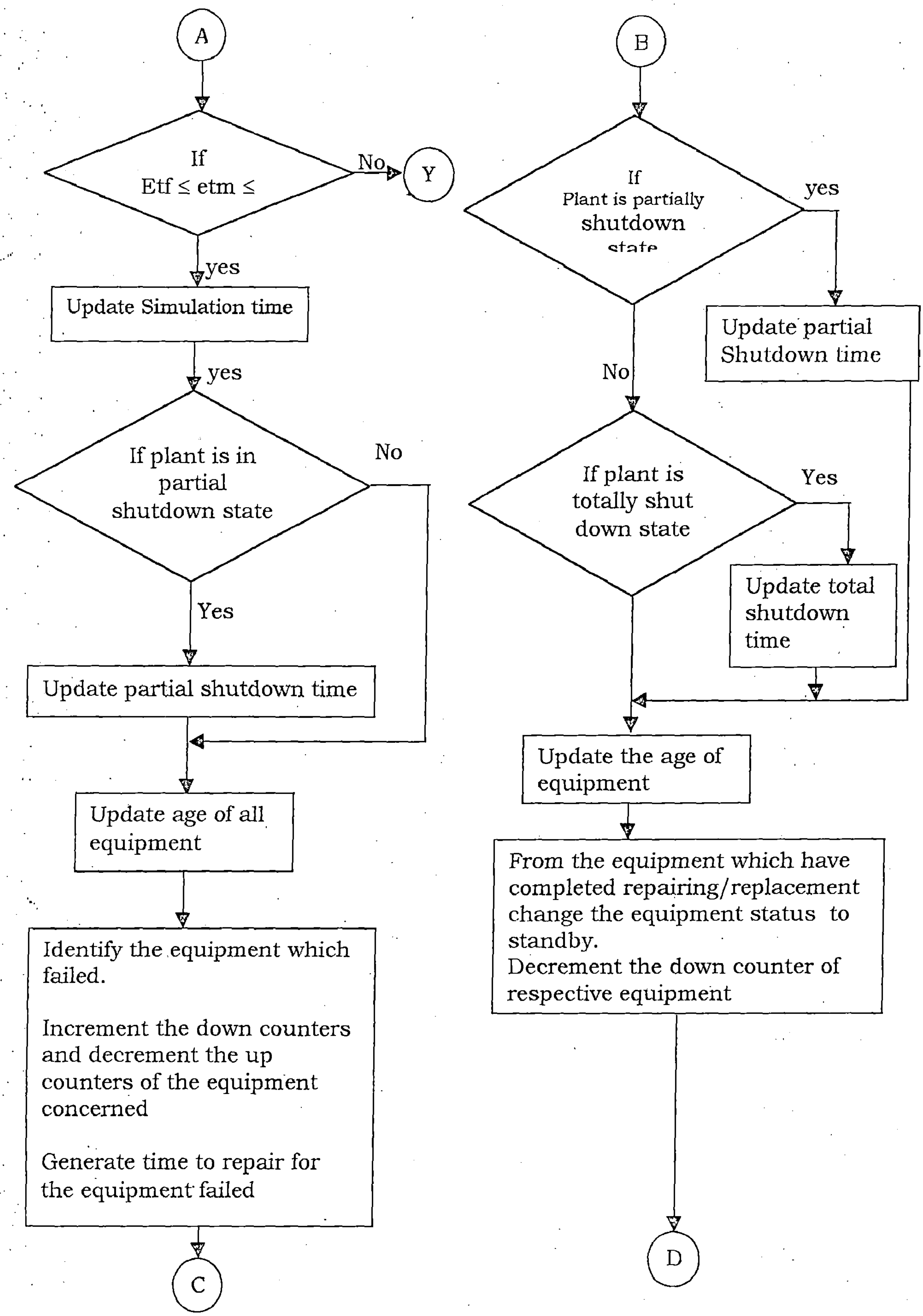
In this policy, one unit (say, I.D. fan No. 1) is repaired on failure. During repair of this unit (I.D. Fan No.1), preventive maintenance is done on any other unit if age of that unit has reached a_1*m time units, where a_1 is a constant and m is the time to failure of that unit. Further for subsystems with standby redundant units (pulverizes and boiler feed pumps), preventive maintenance is done if the unit reaches an age a_2*m time units. Thus for this policy a_1*m is the lower limit and a_2*m is the upper limit of age at preventive maintenance for units of a subsystem having standby redundant units. However, the subsystems which do not have redundant units are preventively maintained (at and age $\geq a_1 \cdot m$) only when another unit/subsystem is under repair.

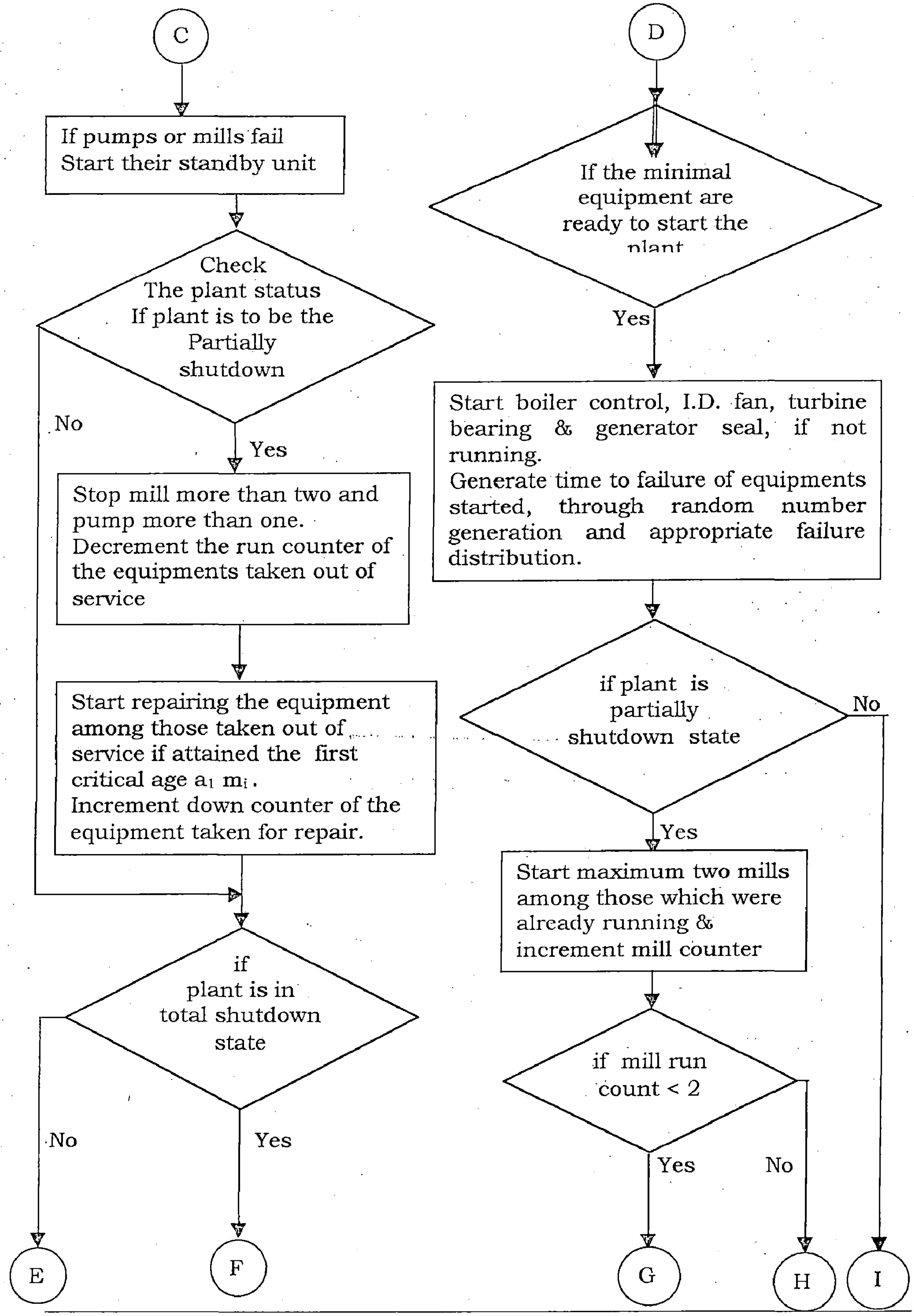
Further we assume, for simplicity, $a_{11} = a_{12} = a_{13} = \ldots\ldots\ldots a_{1n} = a_1$ and $a_{21} = a_{22} = a_{23} = \ldots\ldots\ldots\ldots a_{2n} = a_2$. The simulation model follows step to find optimal values of a_1 and a_2 for a system with n units where more than one unit has increasing failure rate.

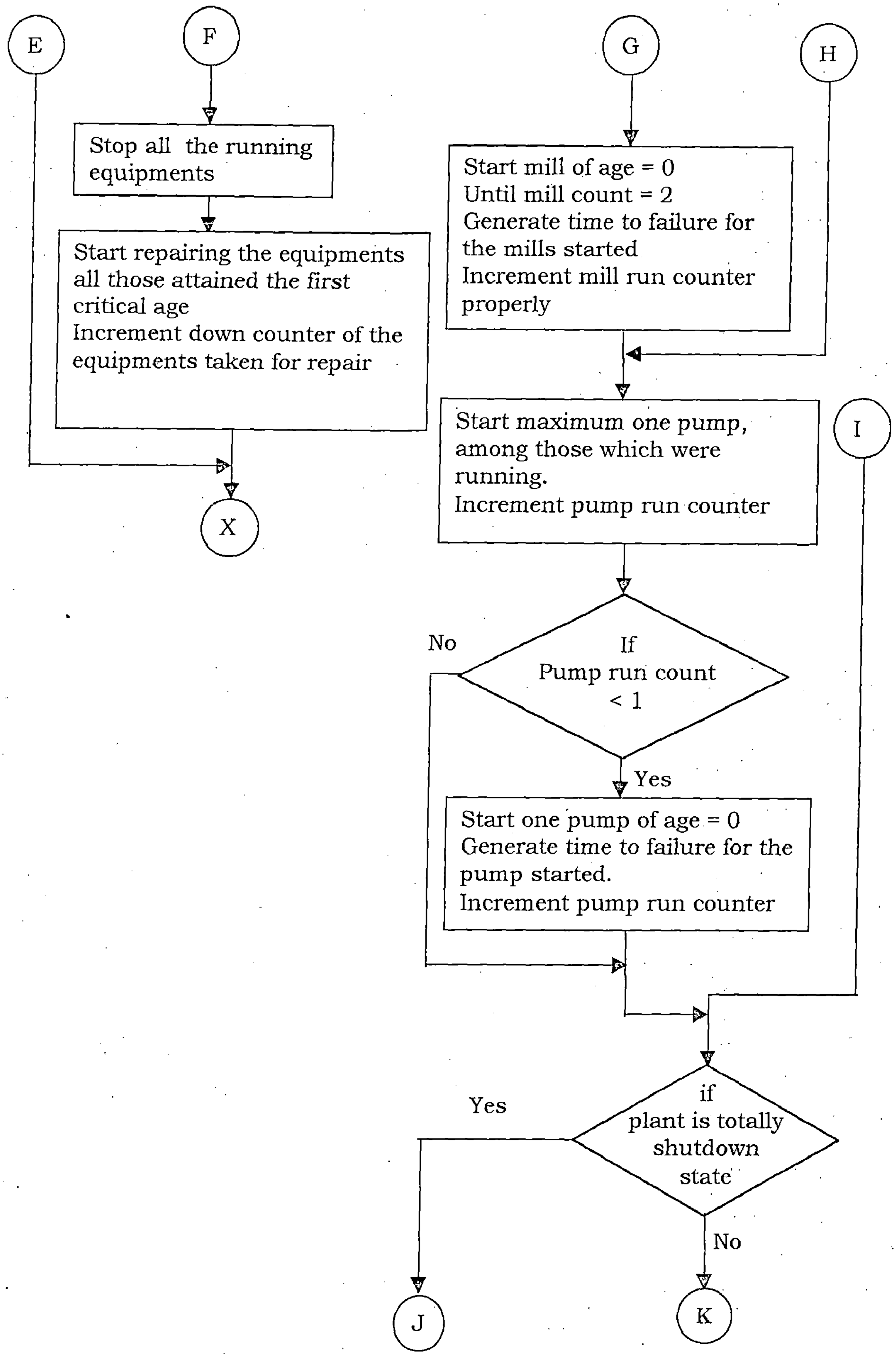
For initial set of condition we assumed that all units have just gone into service. A flowchart of computer program to compute the availability of thermal power plant with various a_1 and a_2 is given below. The program developed in C++ language based on this flowchart given in the appendix-1.

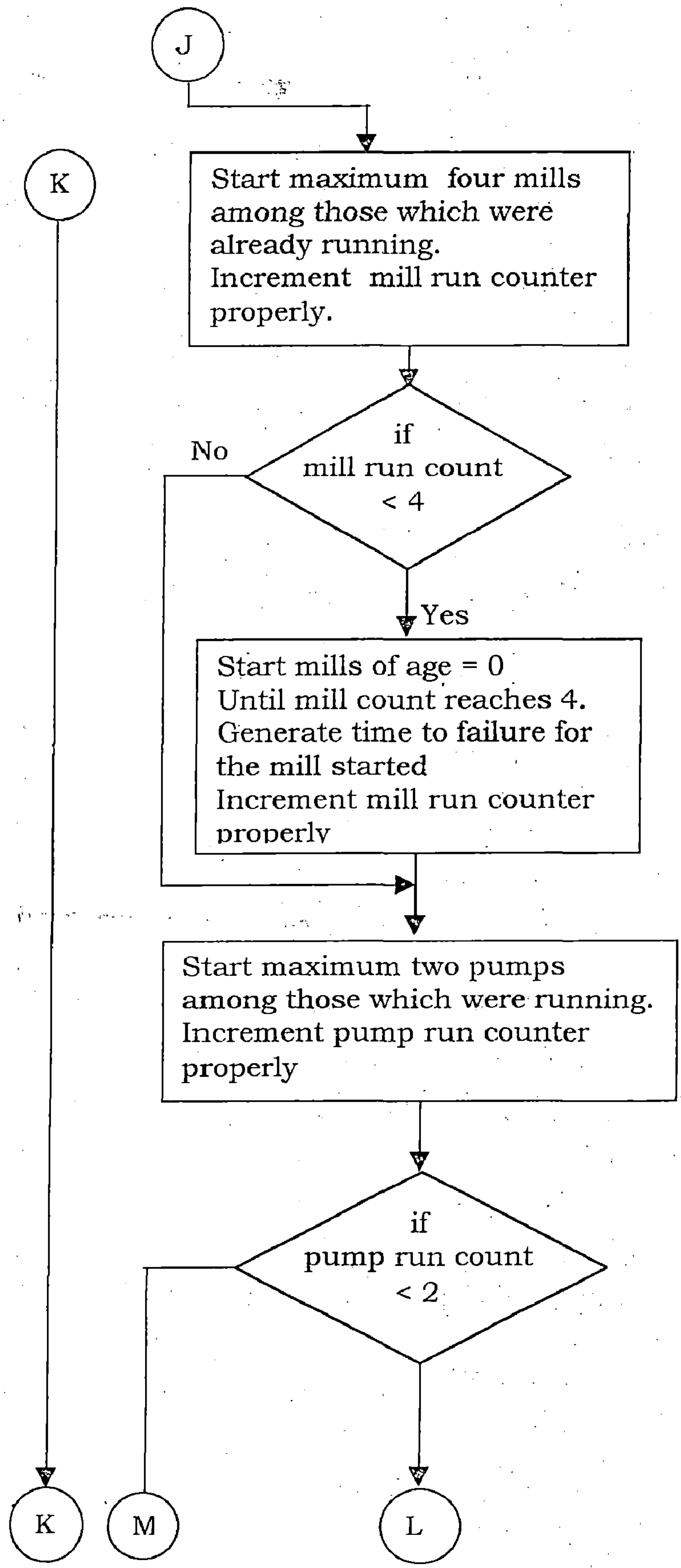
4.2. FLOWCHART FOR MONTE CARLO SIMULATION

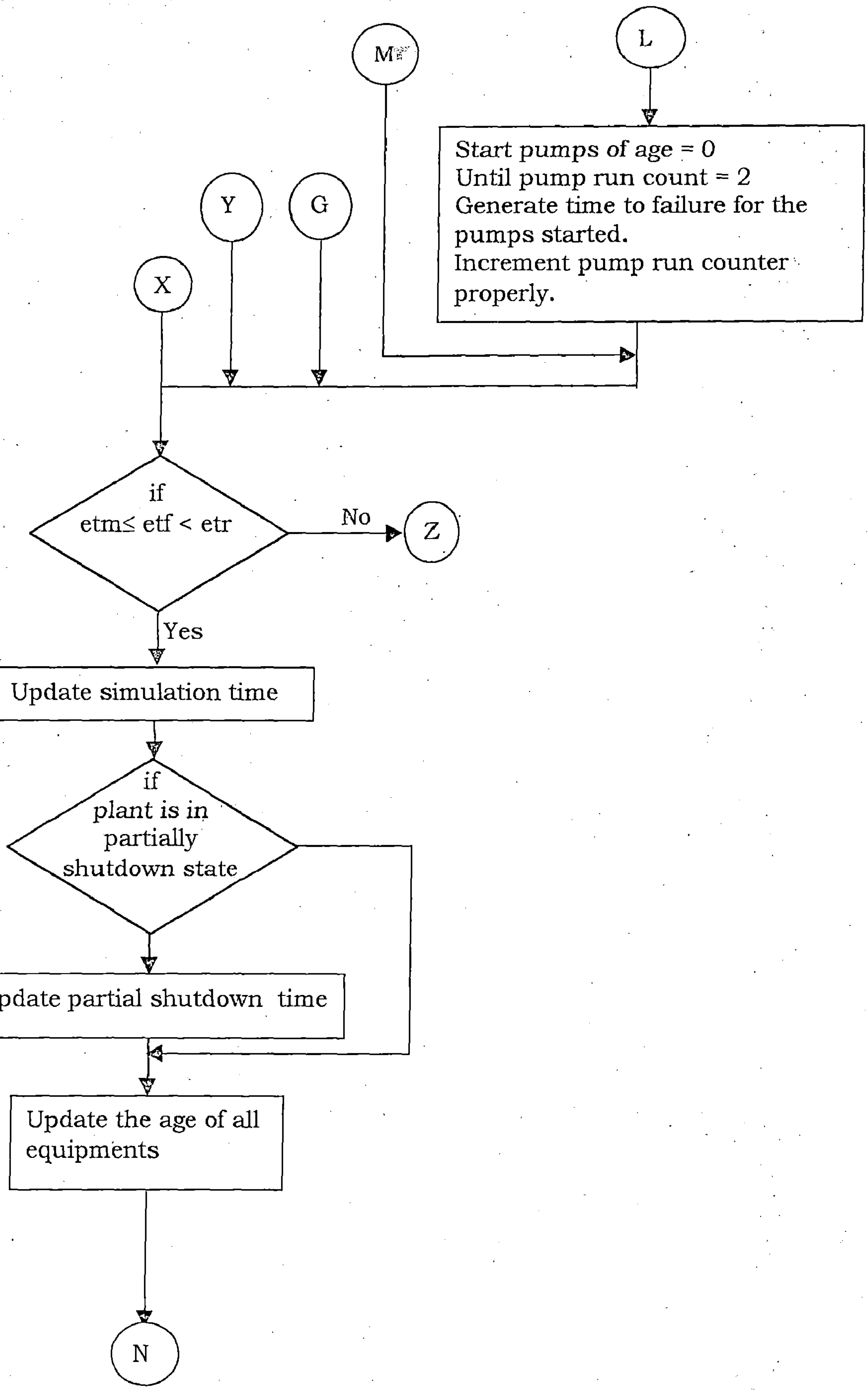


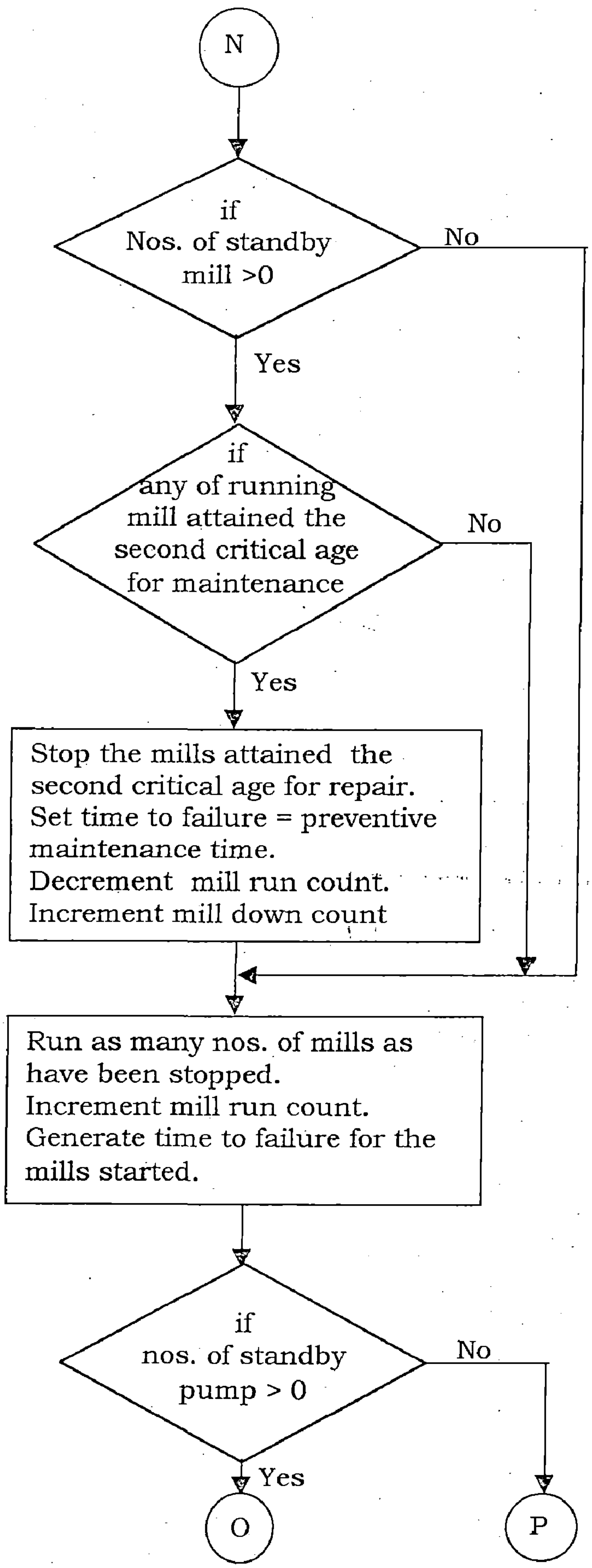

A
If
Etf ≤ etm ≤
No
Y
yes
Update Simulation time
yes
If plant is in partial shutdown state
No
Yes
Update partial shutdown time
Update age of all equipment
Identify the equipment which failed.
Increment the down counters and decrement the up counters of the equipment concerned
Generate time to repair for the equipment failed
C
B
If Plant is partially shutdown state
yes
Update partial Shutdown time
No
If plant is totally shut down state
Yes
Update total shutdown time
Update the age of equipment
From the equipment which have completed repairing/replacement change the equipment status to standby.
Decrement the down counter of respective equipment
D

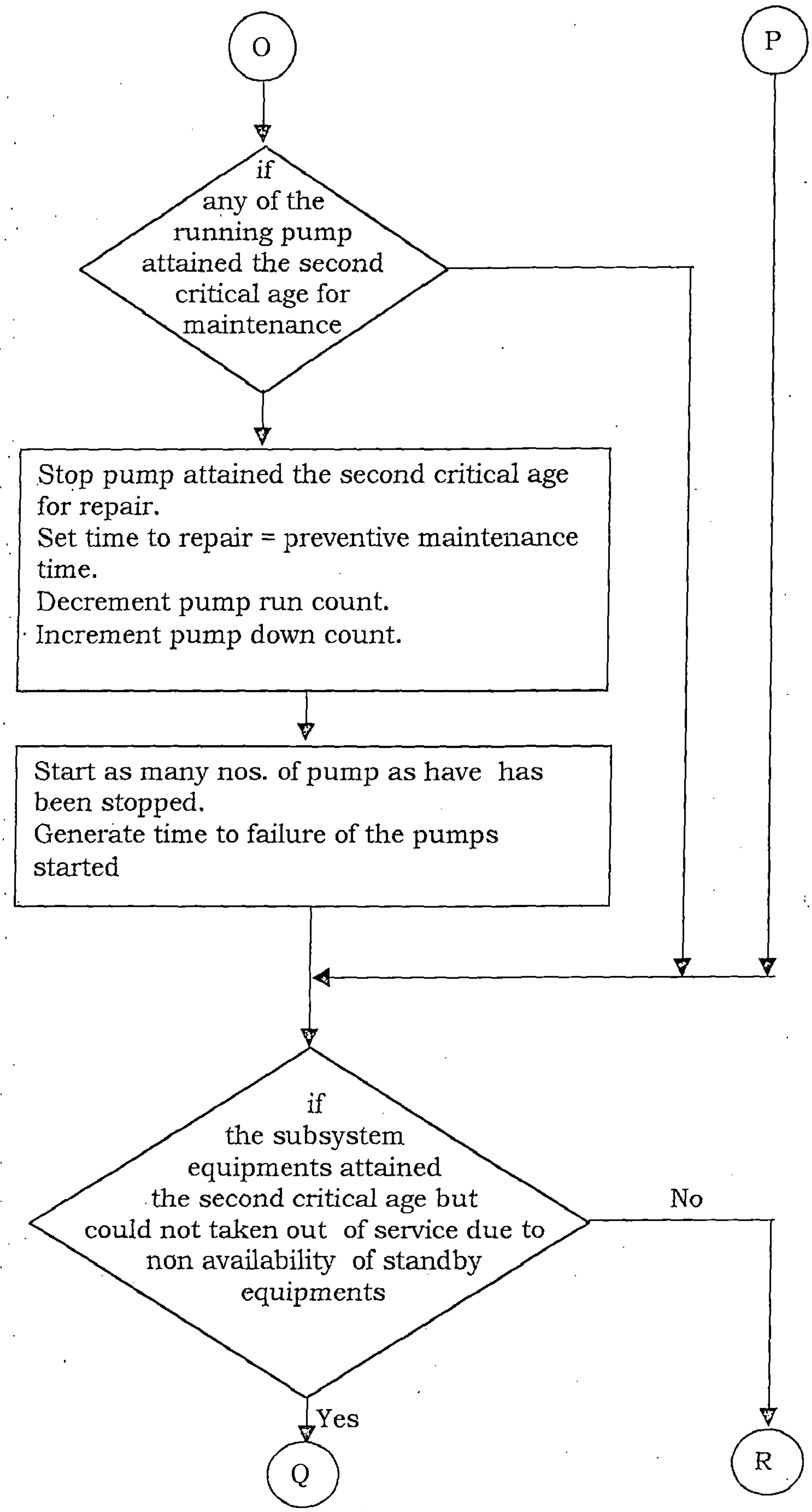

C
If pumps or mills fail
Start their standby unit
Check
The plant status
If plant is to be the
Partially
shutdown
No
Yes
Stop mill more than two and pump more than one.
Decrement the run counter of the equipments taken out of service
Start repairing the equipment among those taken out of service if attained the first critical age $a_1 m_i$.
Increment down counter of the equipment taken for repair.
if
plant is in
total shutdown
state
No
Yes
E
F
D
If the minimal
equipment are
ready to start the
plant
Yes
Start boiler control, I.D. fan, turbine bearing & generator seal, if not running.
Generate time to failure of equipments started, through random number generation and appropriate failure distribution.
if plant is
partially
shutdown state
No
Yes
Start maximum two mills among those which were already running & increment mill counter
if mill run
count < 2
Yes
No
G
H
I

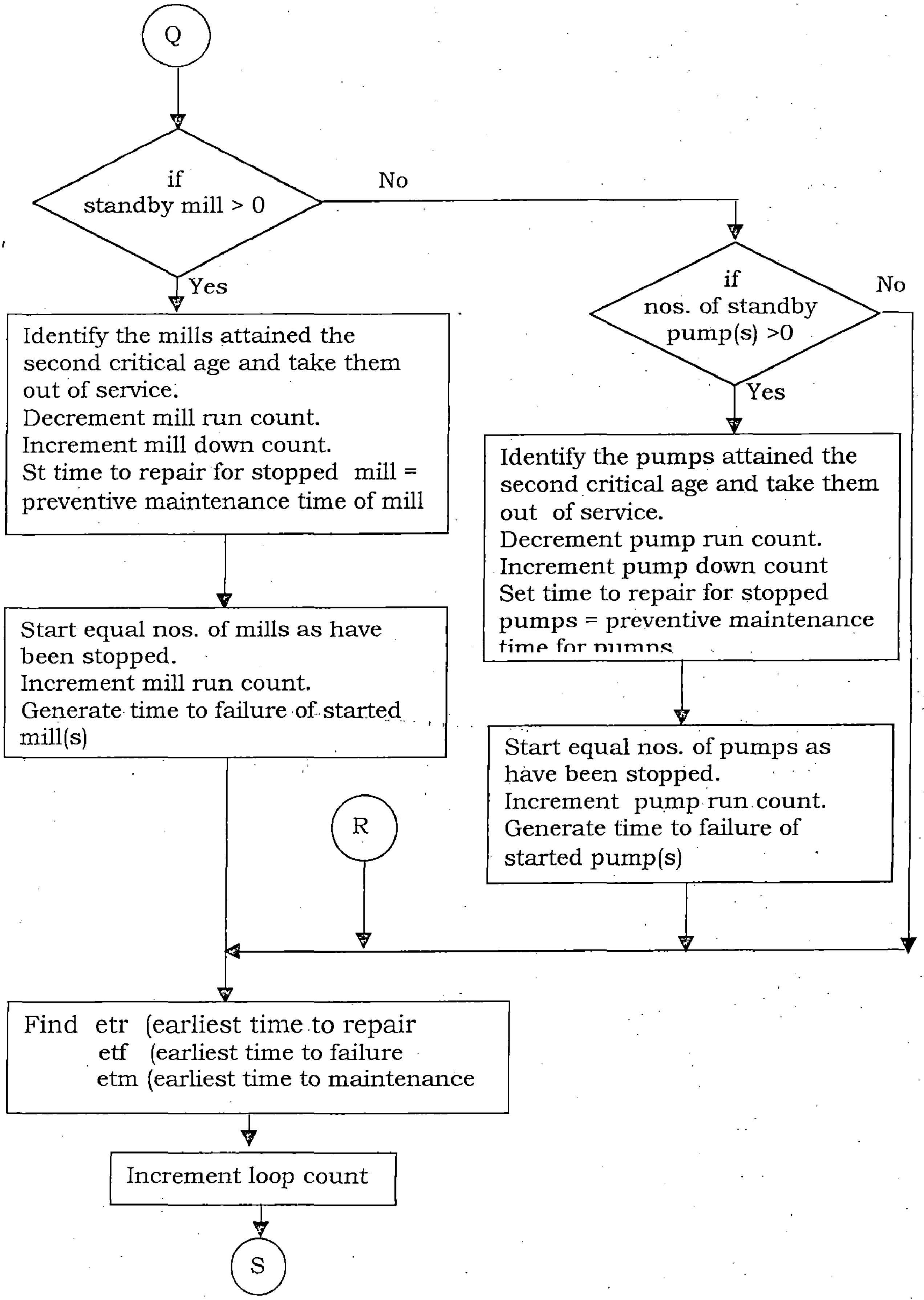

E
F
Stop all the running equipments
Start repairing the equipments all those attained the first critical age
Increment down counter of the equipments taken for repair
X
G
H
Start mill of age = 0
Until mill count = 2
Generate time to failure for the mills started
Increment mill run counter properly
Start maximum one pump, among those which were running.
Increment pump run counter
I
No
If Pump run count < 1
Yes
Start one pump of age = 0
Generate time to failure for the pump started.
Increment pump run counter
Yes
if plant is totally shutdown state
No
J
K

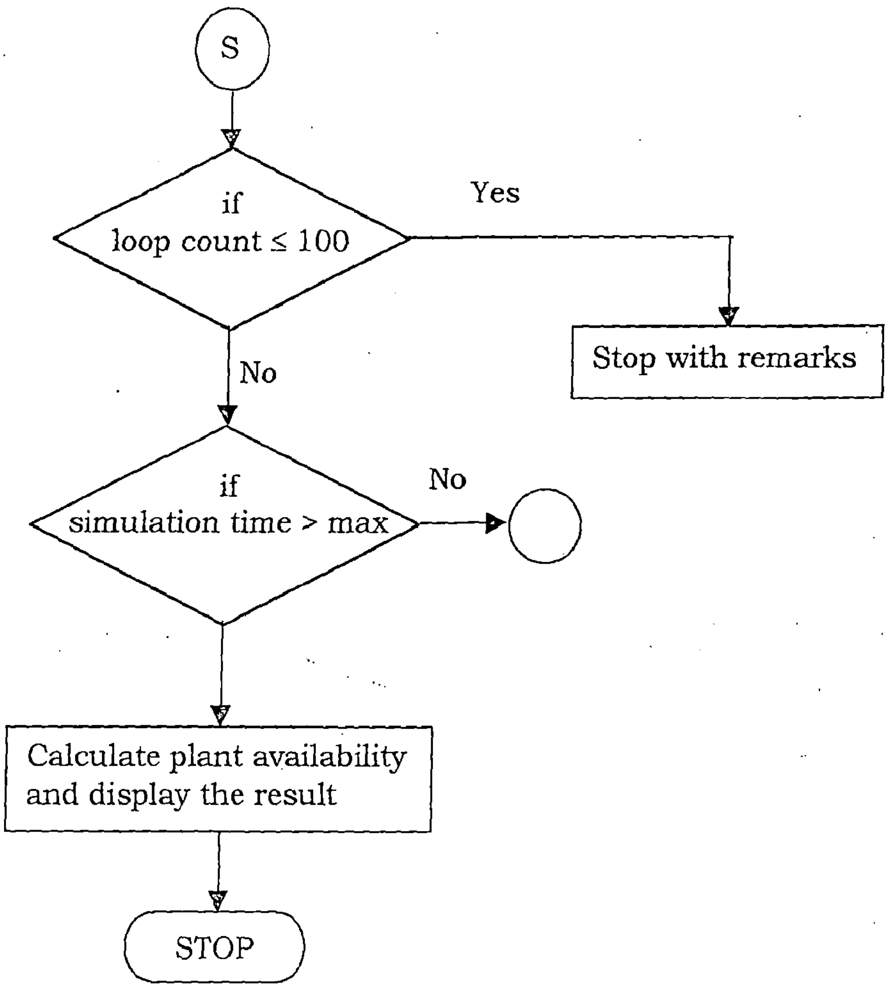

J
K
Start maximum four mills among those which were already running.
Increment mill run counter properly.
if mill run count < 4
No
Yes
Start mills of age = 0
Until mill count reaches 4.
Generate time to failure for the mill started
Increment mill run counter properly
Start maximum two pumps among those which were running.
Increment pump run counter properly
if pump run count < 2
K
M
L


M
L
Start pumps of age = 0
Until pump run count = 2
Generate time to failure for the pumps started.
Increment pump run counter properly.
Y
G
X
if
etm≤ etf < etr
No
Z
Yes
Update simulation time
if
plant is in partially shutdown state
Update partial shutdown time
Update the age of all equipments
N


N
if
Nos. of standby
mill >0
No
Yes
if
any of running
mill attained the
second critical age
for maintenance
No
Yes
Stop the mills attained the
second critical age for repair.
Set time to failure = preventive
maintenance time.
Decrement mill run count.
Increment mill down count
Run as many nos. of mills as
have been stopped.
Increment mill run count.
Generate time to failure for the
mills started.
if
nos. of standby
pump > 0
No
Yes
O
P


O
P
if
any of the
running pump
attained the second
critical age for
maintenance
Stop pump attained the second critical age for repair.
Set time to repair = preventive maintenance time.
Decrement pump run count.
Increment pump down count.
Start as many nos. of pump as have has been stopped.
Generate time to failure of the pumps started
if
the subsystem
equipments attained
the second critical age but
could not taken out of service due to
non availability of standby
equipments
No
Yes
Q
R


Q
if
standby mill > 0
No
Yes
Identify the mills attained the second critical age and take them out of service.
Decrement mill run count.
Increment mill down count.
St time to repair for stopped mill = preventive maintenance time of mill
Start equal nos. of mills as have been stopped.
Increment mill run count.
Generate time to failure of started mill(s)
if
nos. of standby pump(s) >0
No
Yes
Identify the pumps attained the second critical age and take them out of service.
Decrement pump run count.
Increment pump down count
Set time to repair for stopped pumps = preventive maintenance time for pumps
Start equal nos. of pumps as have been stopped.
Increment pump run count.
Generate time to failure of started pump(s)
R
Find etr (earliest time to repair
etf (earliest time to failure
etm (earliest time to maintenance
Increment loop count
S


S
if
loop count ≤ 100
Yes
Stop with remarks
No
if
simulation time > max
No
Calculate plant availability
and display the result
STOP

5.3. COMPUTER SIMULATION

The thermal power plant operation is realized with the help of the developed program. As per Indian Boiler Act, thermal power plant has to take shutdown once in a year for annual inspection when thorough overhauling of the plant is done. Hence the model is simulated for a maximum simulation period of 9000 hours. The computer simulation are done for different combinational values of a_1 and a_2 and unit availability is evaluated. The result are tabulated in table (5.1). it is seen from the table that the availability is maximum and equal to 69.80 % for $a_1 = 0.95$ and $a_2 = 1.3$.

The availability of individual for this optimal condition is shown in table (5.2)

Table 5.1. Percentage Unit Availability, with various values of a1 and a2

a1 \ a2	1.1	1.2	1.25	1.3
0.85	67.83	66.35	66.52	68.45
0.9	68.96	67.33	67.59	67.48
0.95	67.81	68.95	69.45	69.80
1	68.86	68.93	67.18	68.90
1.1	67.79	67.94	68.53	68.90

Tabel 5.2. Availability of individual unit for optimal condition.

Equipment	Running time	Down time	Availability
Boiler control	5011	996	0.834193
I.D. Fan No.1	5011	657	0.884086
I.D. Fan No.2	5011	513	0.907133
Mill No. 1	3425	213	0.941451
Mill No. 2	4425	72	0.983989
Mill No. 3	5661	48	0.991592
Mill No. 4	5541	72	0.987173
Mill No. 5	1874	48	0.975026
Mill No. 6	3148	96	0.970407
B.F. Pump No.1	4166	443	0.903884
B.F. Pump No.2	4209	144	0.966919
B.F. Pump No.3	2817	144	0.951368
Turbine bearing	5011	1465	0.77378
Generator seal	5011	2766	0.644336

CHAPTER-VI

CONCLUSION

Preventive maintenance is adopted in thermal power plants to the improve the reliability. This is necessary as any loss of load due to break down of any or all unit of plant do not only result loss of revenue but also may resulting threat to security of the whole system.

Among the different methods of reliability evaluation Monte Carlo method is selected to evaluate the reliability of different present maintenance scheduling of equipment's. This method is selected because of its relative advantage over analytical methods which become very complex to use for thermal power plants with complex equipments wit complex failure and repair time distribution.

One major problem faced by the plant managers while preventive maintenance scheduling is the non availability of systematic record of failure and repair time of similar type of equipments. To overcome this, procedure for processing human originated information has been devised. The method consist of three steps, get useful information from the experts, assessment of the quality of the expert and lastly combining the responses of several expert to yield a unique, hopefully better response. All the three steps is proposed to be done in the framework of possibility theory offers a simple theory of uncertainty that explicitly takes into account the lack of precision which is more realistic. Assessing and pooling step of experts opinion by possibility approach presents less difficulties relative to the probabilistic (classical) approach.

A real world experiment as described in [9] verified in practice the applicability of the possibilistic approach in the expert judgment domain relative to the evaluation and the pooling methods.

The probability distribution function derived from the possibilistic approach is then used in the Monte Carlo Simulation Methods. Where member of simulation for different preventive maintenance interval are observed. The preventive maintenance schedule corresponding to the highest availability may be selected.

LIST OF PROGRAM OPTMAIN

```
# include <iostream.h>
# include <conio.h>
# include <stdlib.h>        // For randomize(), rand
# include <time.h>          // For randomize
# include <math.h>          // For pow & log
# include <iomanip.h>       // For setw

enum plantstatus {nsd,psd,tsd};
enum eqptstatus {run, stby, brdn};

struct equipment
 {
  int gmf,itf,gmr,itr,prmt,tnf,tnr,rnt,dnt,mf;
  float btf,btr;
  eqptstatus est;

 };

void main ( )
 {
  clrscr ();
  plantstatus sd;
  void update_time (equipment& eqp, int tm);
  int time_to_failrep(int gm, int it, float bt);
  int mean_time_to_faliure(int gm, int it, float bt);
  const int max = 9000;
  equipment eqp[14];
  float a1, a2;
  cout<< "\n Enter the value of a1 : "; cin>>a1;
  cout<< "\n Enter the value of a2 : "; cin>>a2;
  int st = 0;        // initialize simulation time.
  int tsdt = 0;      // initialize total shut down time.
  int psdt = 0;      // initialize partial shut down time.
  //int rt = 0;      // initialize run time.
  int etf, etm, etr;
  int gf[14] = {60,240,192,120,100,150,165,70,65,96,120,72,130,140};
  int it_f[14]={550,750,600,1200,1500,2000,2000,1000,900,1320,1500,
                1000,2400,700};
  float bf[14]={1.50,1.90,1.59,1.70,1.80,1.60,1.50,1.60,1.40,1.21,
                1.55,1.60,1.52,1.65};
  int gr[14]={48,16,24,8,12,24,20,30,40,16,48,120,100,72};
```

```
int it_r[14]={96,40,60,48,40,60,50,70,100,96,120,240,200,350};
float br[14]={1.65,1.08,1.94,1.55,1.65,1.70,1.60,1.50,1.70,1.21,
                    1.50,1.80,1.55,1.90};
int pm[14]={100,30,30,24,24,24,24,24,24,48,48,48,100,100};
for ( int i = 0; i<14; i++)       // initialize the known parameters -
{                                 // - of all the equipments.
    eqp[i].gmf = gf[i];
    eqp[i].itf = it_f[i];
    eqp[i].btf = bf[i];
    eqp[i].gmr = gr[i];
    eqp[i].itr = it_r[i];
    eqp[i].btr = br[i];
    eqp[i].prmt= pm[i];
    eqp[i].mf = mean_time_to_faliure(eqp[i].gmf,eqp[i].itf,eqp[i].btf);
    eqp[i].rnt = 0;
    eqp[i].dnt = 0;
    eqp[i].est = stby;
    eqp[i].tnf = max;
    eqp[i].tnr = max;
  }
etf=0; //eqp[0].tnf-eqp[0].rnt;
etr=0; //eqp[0].tnr-eqp[0].dnt;
etm=0; //eqp[0].mf*a2 - eqp[0].rnt;


//starting of setting run & down counters of fans, mills & pumps.
int avoid;
int mark;

int mrn = 0;                      //mrn=counter for running mills.
int mdn = 0;                      //mdn=counter for down mills.
int prn = 0;                      //prn=counter for running pumps.
int pdn = 0;                      //pdn=counter for down pumps.
//int frn = 0;                    //fdn=counter for running fans.
int fdn = 0;
int frn = 0;                      //fdn=counter for down fans.
/* for (int v=1; v<3; v++)
{
 if (eqp[v].est==2) fdn++;        //counting nos. of down fans.
}
for (int l=3;l<9;l++)
{
 if (eqp[l].est==2)  { mdn++; break;}     //counting nos. of down mills.
 else if (eqp[l].est==0)  {mrn++; break;} //counting nos. of running mills.
}
```

```
for (int m=9;m<12;m++)
{
 if (eqp[m].est==2) {pdn++; break; }      //counting nos. of down pumps.
 else if (eqp[m].est==0)  {prn++; break;} //counting nos. of down pumps.
}          //end of setting run &down counters of fans, mills & pumps.
*/
sd = nsd;
int loop_count=0;
do
 {

  if((etr<=etf)&&(etr<=etm))
     {
      cout<<endl<<"etr loop";
      st+=etr;

      if (sd==psd) psdt+=etr;
      else if (sd==tsd) tsdt+=etr;

      for(int k=0; k<14; k++)
       {
        update_time(eqp[k], etr);
        if((eqp[k].dnt==eqp[k].tnr)&&(eqp[k].est==2))
                                        //checking the condition for rep time over.
         {
          eqp[k].dnt=0;
          eqp[k].est=stby;//changing the status of mill from down to standby.
          if((k>0)&&(k<3)) fdn--;          //less fan_down count.
              else if((k>2)&&(k<9)) mdn--;    //less mill_down count.
               else if((k>8)&&(k<12)) pdn--;  //less pump_down count.

         }
       }
// start

if((eqp[0].est!=2)&&((eqp[1].est!=2)&&(eqp[2].est!=2))&&
  (mdn<5)&&(pdn<3)&&(eqp[12].est!=2)&&(eqp[13].est!=2))
 {cout<<" mill start";           // checking for minimal start condition
  if(eqp[0].est==1)
     {
      eqp[0].est=run;
      if (eqp[0].rnt==0)
      {
       eqp[0].tnf=time_to_failrep(eqp[0].gmf, eqp[0].itf, eqp[0].btf);
       eqp[0].tnr=max;
```

```
        }
      }
   if(eqp[1].est==1)
      {
       eqp[1].est=run;
       if (eqp[1].rnt==0)
       {
        eqp[1].tnf=time_to_failrep(eqp[1].gmf, eqp[1].itf, eqp[1].btf);
        eqp[1].tnr=max;
       }
      }
   if(eqp[2].est==1)
      {
       eqp[2].est=run;
       if (eqp[2].rnt==0)
       {
        eqp[2].tnf=time_to_failrep(eqp[2].gmf, eqp[2].itf, eqp[2].btf);
        eqp[2].tnr=max;
       }
      }
   if(eqp[12].est==1)
      {
       eqp[12].est=run;
       if (eqp[12].rnt==0)
       {
        eqp[12].tnf=time_to_failrep(eqp[12].gmf, eqp[12].itf, eqp[12].btf);
        eqp[12].tnr=max;
       }
      }
   if(eqp[13].est==1)
      {
       eqp[13].est=run;
       if (eqp[13].rnt==0)
       {
        eqp[13].tnf=time_to_failrep(eqp[13].gmf, eqp[13].itf, eqp[13].btf);
        eqp[13].tnr=max;
       }
      }

   if ((2<mdn<5)&&(pdn==2))    // check for partial start condition
     { cout<<" pstr ";
      for (int n=3; n<9; n++)
       {
        if((mrn<2)&&(eqp[n].est==1)&&(eqp[n].rnt!=0))
         {
```

```
    eqp[n].est=run;
    mrn++;
    if (eqp[n].rnt==0)
        {
        eqp[n].tnf=time_to_failrep(eqp[n].gmf, eqp[n].itf, eqp[n].btf);
        eqp[n].tnr=max;
        }
  }
}

for (int o=3; o<9; o++)
 {
  if ((mrn<2)&&(eqp[o].est==1))
   {
    eqp[o].est=run;
    mrn++;
    if (eqp[o].rnt==0)
        {
        eqp[o].tnf=time_to_failrep(eqp[o].gmf, eqp[o].itf, eqp[o].btf);
        eqp[o].tnr=max;
        }
   }
 }

for (int p=9;p<12;p++)
 {
  if((prn<1)&&(eqp[p].est==1)&&(eqp[p].rnt!=0))
   {
    eqp[p].est=run;
    prn++;
    if (eqp[p].rnt==0)
        {
        eqp[p].tnf=time_to_failrep(eqp[p].gmf, eqp[p].itf, eqp[p].btf);
        eqp[p].tnr=max;
        }
   }
 }

for (int q=9;q<12;q++)
 {
  if((prn<1)&&(eqp[q].est==1))
   {
    eqp[q].est=run;
    prn++;
    if (eqp[q].rnt==0)
```

```
            {
            eqp[q].tnf=time_to_failrep(eqp[q].gmf, eqp[q].itf, eqp[q].btf);
            eqp[q].tnr=max;
            }
        }
      }  // end of for loop
  }      //end of if loop with (2<mdn<5)&&(pdn==2)


else if ((mdn<3)&&(pdn<2)&&(fdn<1))  //checking for full start
 {cout<<" fstr ";
  for (int r=3; r<9; r++)
  {
     if((mrn<4)&&(eqp[r].est==1)&&(eqp[r].rnt !=0))
      {cout<<" mrn ";
       eqp[r].est=run;
       mrn++;
      }
  }
  for (int s=3; s<9; s++)
  {
     if((mrn<4)&&(eqp[s].est==1))
     {cout<<" mrn0 ";
      eqp[s].est=run;
      mrn++;
      if (eqp[s].rnt==0)
            {
            eqp[s].tnf=time_to_failrep(eqp[s].gmf, eqp[s].itf, eqp[s].btf);
            eqp[s].tnr=max;
            }
     }
  }

  for (int t=9; t<12; t++)
  {
     if((prn<2)&&(eqp[t].est==1)&&(eqp[t].rnt!=0))
     {cout<<" prn ";
      eqp[t].est=run;
      prn++;
     }
  }
  for (int u=9; u<12; u++)
  {
     if((prn<2)&&(eqp[u].est==1))
     {cout<<" prn0 ";
```

```
        eqp[u].est=run;
        prn++;
        if (eqp[u].rnt==0)
              {
              eqp[u].tnf=time_to_failrep(eqp[u].gmf, eqp[u].itf, eqp[u].btf);
              eqp[u].tnr=max;
              }
      }
   }     // end of for loop
  }      //end of else if loop with((mdn<3)&&(pdn<2)&&(fdn<1))
 }       //end of start loop
}        //end of if loop for lowest repair time.

 else if((etf<etm)&&(etf<etr))
  {cout<<endl<<"etf loop";
     st+=etf;                          // update simulation time
      if (sd==psd) psdt+=etf;
      else if (sd==tsd) tsdt+=etf;

     for(int w=0; w<14; w++)
      {
       update_time(eqp[w], etf);  //update run/down time of all equipments
       if ((eqp[w].est==0)&&(eqp[w].rnt==eqp[w].tnf))
        {
         eqp[w].est=brdn;
         if((w>0)&&(w<3)) {fdn++; frn--;}
            else if((w>2)&&(w<9)) {mdn++;  mrn--;}
             else if((w>8)&&(w<12)) {pdn++; prn--;}  // stops the machine
         eqp[w].tnf=max;
         eqp[w].tnr=time_to_failrep(eqp[w].gmr, eqp[w].itr, eqp[w].btr);
         eqp[w].rnt=0;
        }
     }    //end of for loop


    if (eqp[0].est==2) sd=tsd;          //checking plant status
     else if (fdn>1) sd=tsd;
      else if (mdn>4) sd=tsd;
       else if (pdn>2) sd=tsd;
        else if(eqp[12].est==2) sd=tsd;
         else if(eqp[13].est==2) sd=tsd;
    else if(fdn==1) sd=psd;
     else if((mdn>2)&&(mdn<5)) sd=psd;
      else if(pdn==2) sd=psd;
    else sd= nsd;
```

```
  //     void partialshutdown ()
 if(sd==psd)
  {
   for (int mm=8;mm>2;mm--)
        {
         if((eqp[mm].est==0)&&(mrn>2))
          {
           eqp[mm].est=stby;
           mrn--;
          }
        }
   for (int kk=11;kk>8;kk--)
        {
         if((eqp[kk].est==0)&&(prn>1))
          {
           eqp[kk].est=stby;
           prn--;
          }
        }
 for (int oo=3;oo<12;oo++)
  {
   if(((eqp[oo].mf*a1)<=(eqp[oo].rnt))&&(eqp[oo].est==1))
        {
         eqp[oo].est=brdn;
         if((oo>0)&&(oo<3)) fdn++;
          else if((oo>2)&&(oo<9)) mdn++;
           else if((oo>8)&&(oo<12)) pdn++;
         eqp[oo].tnf=max;
         eqp[oo].tnr=time_to_failrep(eqp[oo].gmr, eqp[oo].itr,
                                          eqp[oo].btr);
         eqp[oo].rnt=0;
        }
  }   // end of for loop
}  // end of psd loop
if (sd==tsd)
 { cout<<" TSD ";
 for (int nn=0; nn<14; nn++)
  {
   if (eqp[nn].est==0)
   {cout<<" SB"<<nn;
    eqp[nn].est=stby;
    if((nn>0)&&(nn<3)) frn--;
```

```
                else if((nn>2)&&(nn<9)) mrn--;
                  else if((nn>8)&&(nn<12)) prn--;
             }
            if(((eqp[nn].mf*a1)<=(eqp[nn].rnt))&&(eqp[nn].est==1))
             { cout<<" DN"<<nn;
                 eqp[nn].est=brdn;
                 if((nn>0)&&(nn<3)) fdn++;
                  else if((nn>2)&&(nn<9)) mdn++;
                   else if((nn>8)&&(nn<12)) pdn++;

                 eqp[nn].tnf=max;
                 eqp[nn].tnr=time_to_failrep(eqp[nn].gmr, eqp[nn].itr,
                                                        eqp[nn].btr);
                 eqp[nn].rnt=0;
            }

          }    //end of for loop
        }      //end of tsd loop
    }   //end of loop of earliest time of faliure

    else if((etm<=etf)&&(etm<etr))
    {
       cout<<endl<<"etm loop";                    //start of scheduled maintenance
loop
       st+=etm;
        if (sd==psd) psdt+=etm;
        else if (sd==tsd) tsdt+=etr;
       for(int pk=0; pk<14; pk++)
          update_time(eqp[pk], etm);
       cout<<" MDN "<<mdn<<" MRN "<<mrn;
       int mscnt = 6-mdn-mrn; cout<<" MSCNT "<<mscnt; // mill shut down
count
       int pscnt = 3-pdn-prn; cout<<" PSCNT "<<pscnt; // pump shut down count
       int mstp=0;
       int pstp=0;
       for (int pp=3; pp<9; pp++)
        {
        if ((mscnt-mstp)>0)
          {
           //update_time(eqp[pp], etm);
          if ((int(eqp[pp].mf*a2)<=(eqp[pp].rnt))&&(eqp[pp].est==0))
              { cout<<" pp= "<<pp<<" ";
               eqp[pp].est=brdn;
               mdn++;
               mrn--;
```

```
            mstp++;
            eqp[pp].tnf=max;
            eqp[pp].tnr=time_to_failrep(eqp[pp].gmr, eqp[pp].itr,
                                                              eqp[pp].btr);
            eqp[pp].rnt=0;
            if (mark==pp) mark=max;
           }
       }
     else if((int(eqp[pp].mf*a2)<=(eqp[pp].rnt))&&(eqp[pp].est==0))
        mark = pp;
    }
   for (int qq=8; qq>2; qq--)
    {
     if((mstp>0)&&(eqp[qq].est==1))
      {
          eqp[qq].est=run;
          mstp--;
          mrn++;

       if (eqp[qq].rnt==0)
          {
          eqp[qq].tnf=time_to_failrep(eqp[qq].gmf, eqp[qq].itf, eqp[qq].btf);
          eqp[qq].tnr=max;
          }
      }
    }
   for (int rr=9; rr<12; rr++)
    {
     if ((pscnt-pstp)>0)
      {
       if((int(eqp[rr].mf*a2)<=(eqp[rr].rnt))&&(eqp[rr].est==0))
          {cout<<" RR "<<rr;
          eqp[rr].est=brdn;
          pdn++;
          prn--;
          pstp++;
          eqp[rr].tnf=max;
          eqp[rr].tnr=time_to_failrep(eqp[rr].gmr, eqp[rr].itr,
                                                              eqp[rr].btr);
          eqp[rr].rnt=0;
          if (rr==mark)  mark=max;
         }
      }
    else if((int(eqp[rr].mf*a2)<=(eqp[rr].rnt))&&(eqp[rr].est==0))
      mark = rr;
```

```
        }
        for (int ss=11; ss>8; ss--)
         {
          if (pstp>0)
           {
            if (eqp[ss].est==1)
             {
                eqp[ss].est=run;
                pstp--;
                prn++;
             if (eqp[ss].rnt==0)
                {
                eqp[ss].tnf=time_to_failrep(eqp[ss].gmf, eqp[ss].itf,
                                                eqp[ss].btf);
                eqp[ss].tnr=max;
                }
             }
           }
         }
    } // end of scheduled maintenance loop.

    if ((avoid!=0)&&(etr<etm)&&(etf<etm))
     {
       cout<<endl<<"etm loop";       //start of scheduled maintenance loop
       if(etr<=etf) etm=etr;
       else if(etr>etf) etm=etf;
       for(int pk=0; pk<14; pk++)
          update_time(eqp[pk], etm);
       st+=etm;
       cout<<" MDN "<<mdn<<" MRN "<<mrn;
       int mscnt = 6-mdn-mrn; cout<<" MSCNT "<<mscnt; // mill shut down
count
       int pscnt = 3-pdn-prn; cout<<" PSCNT "<<pscnt; // pump shut down count
       int mstp=0;
       int pstp=0;
       for (int pp=3; pp<9; pp++)
        {
         if ((mscnt-mstp)>0)
           {
            //update_time(eqp[pp], etm);
            if ((int(eqp[pp].mf*a2)<=(eqp[pp].rnt))&&(eqp[pp].est==0))
               { cout<<" pp= "<<pp<<" ";
                eqp[pp].est=brdn;
                mdn++;
                mrn--;
```

```
            mstp++;
            eqp[pp].tnf=max;
            eqp[pp].tnr=time_to_failrep(eqp[pp].gmr, eqp[pp].itr,
                                                                  eqp[pp].btr);
            eqp[pp].rnt=0;
            if (mark==pp) mark=max;
           }
      }
     else if((int(eqp[pp].mf*a2)<=(eqp[pp].rnt))&&(eqp[pp].est==0))
       mark = pp;
    }
   for (int qq=8; qq>2; qq--)
    {
     if((mstp>0)&&(eqp[qq].est==1))
       {
          eqp[qq].est=run;
          mstp--;
          mrn++;

        if (eqp[qq].rnt==0)
          {
          eqp[qq].tnf=time_to_failrep(eqp[qq].gmf, eqp[qq].itf, eqp[qq].btf);
          eqp[qq].tnr=max;
          }
       }
    }
   for (int rr=9; rr<12; rr++)
    {
     if ((pscnt-pstp)>0)
      {
       if((int(eqp[rr].mf*a2)<=(eqp[rr].rnt))&&(eqp[rr].est==0))
          {cout<<" RR "<<rr;
           eqp[rr].est=brdn;
           pdn++;
           prn--;
           pstp++;
           eqp[rr].tnf=max;
          eqp[rr].tnr=time_to_failrep(eqp[rr].gmr, eqp[rr].itr,
                                                                  eqp[rr].btr);
           eqp[rr].rnt=0;
           if (rr==mark)  mark=max;
          }
      }
    else if((int(eqp[rr].mf*a2)<=(eqp[rr].rnt))&&(eqp[rr].est==0))
      mark = rr;
```

```
   }
   for (int ss=11; ss>8; ss--)
   {
    if (pstp>0)
     {
      if (eqp[ss].est==1)
       {
           eqp[ss].est=run;
           pstp--;
           prn++;
       if (eqp[ss].rnt==0)
           {
           eqp[ss].tnf=time_to_failrep(eqp[ss].gmf, eqp[ss].itf,
                                             eqp[ss].btf);
           eqp[ss].tnr=max;
           }
       }
     }
   }
}
etf=eqp[0].tnf-eqp[0].rnt;
etr=eqp[0].tnr-eqp[0].dnt;
etm=max;
for(int j=1; j<14; j++)
   {
   if ((etf>(eqp[j].tnf-eqp[j].rnt))&&(eqp[j].est==0))
           etf = (eqp[j].tnf-eqp[j].rnt);
   if ((etr>(eqp[j].tnr-eqp[j].dnt))&&(eqp[j].est==2))
           etr = (eqp[j].tnr-eqp[j].dnt);

   }
avoid=0;
for( int jj=3; jj<12; jj++ )
   {
    if (mark==jj)  avoid++;
     else if ((etm>(int(eqp[jj].mf*a2) - eqp[jj].rnt))
                                 &&(eqp[jj].est==0))
                 etm = (int(eqp[jj].mf*a2) - eqp[jj].rnt);

   }

for( int jk=0; jk<14; jk++ )
   { cout<<endl<<" "<<jk<<" rnt "<<eqp[jk].rnt<<" dnt "<<eqp[jk].dnt
                 <<" mf "<<eqp[jk].mf<<" mf*a2 "<<int (eqp[jk].mf*a2)
                 <<"tnr "<<eqp[jk].tnr<<" tnf "<<eqp[jk].tnf
```

```
                    <<" stat "<<eqp[jk].est;
    }

    cout<<endl<<" LC"<<setw(3)<<loop_count
                <<" ETR"<<setw(5) <<etr
                <<" ETF"<<setw(5)<<etf
                <<" ETM"<<setw(5)<<etm
                <<" MDN "<<setw(2)<<mdn
                <<" MRN "<<setw(2)<<mrn
                <<" PDN "<<setw(2)<<pdn
                <<" PRN "<<setw(2)<<prn
                <<"\n ST"<<setw(5)<<st
                <<" PSDT"<<setw(5)<<psdt
                <<" TSDT"<<setw(5)<<tsdt
                <<"\n Enter next loopcount ";

      if (eqp[0].est==2) sd=tsd;          //checking plant status
       else if (fdn>1) sd=tsd;
        else if (mdn>4) sd=tsd;
         else if (pdn>2) sd=tsd;
          else if(eqp[12].est==2) sd=tsd;
           else if(eqp[13].est==2) sd=tsd;
      else if(fdn==1) sd=psd;
       else if((mdn>2)&&(mdn<5)) sd=psd;
        else if(pdn==2) sd=psd;
      else sd= nsd;

   // loop_count++;
   cin>>loop_count;
   if(loop_count>= 100) break;

  }    // end of do loop
  while ( st <= max );     // loop condition

  cout<<endl<<" a1="<<a1<<"  a2="<<a2<<" SIMULATION TIME ="<<st
       <<"loop count="<<loop_count;
  cout<<endl<<" PARTIAL SHUT DOWN TIME = "<<psdt
       <<" TOTAL SHUT DOWN TIME = "<<tsdt;
  cout<<endl<<" UNIT AVAILABILITY = "<<(1-(tsdt+0.5*psdt)/st);
 /* for(int n=0; n<14; n++)
  {
   if ((n+10)%10==0) cout<<endl<<eqp[n].tnf<<"/"<<eqp[n].tnr<<":";
   else cout<<eqp[n].tnf<<"/"<<eqp[n].tnr<<":";
  }
  cout<<endl<<"etr="<<etr<<"etf="<<etf<<"etm="<<etm;*/
```

```
getche ();
}          // end of main



int time_to_failrep (int gama, int ita, float beta )
    {
     int y;
     double z = (1/beta);
     //randomize ();
     double ttf;
     int x = random (1000);
     double u = 1/double(x);
     ttf = (gama +( ita *( pow((-log(u)), z))) );
     y = int(ttf);
     return y;
    }
int mean_time_to_faliure (int gama, int ita, float beta)
    {
     float mttf;
     float x = 1 + 1/beta;
     if (x>=1)
      mttf = gama + ita * (1.0 -(1.0-0.988844)*(x-1.0)/(1.02-1.0));
     else if ( x >= 1.02 )
      mttf=gama+ita*(0.988844-(0.988844-0.978438)*(x-1.02)/(1.04-1.02));
     else if ( x >= 1.04 )
      mttf=gama+ita*(0.978438-(0.978438-0.968744)*(x-1.04)/(1.06-1.04));
     else if ( x >= 1.06 )
      mttf=gama+ita*(0.968744-(0.968744-0.959725)*(x-1.06)/(1.08-1.06));
     else if ( x >= 1.08 )
      mttf=gama+ita*(0.959725-(0.959725-0.951351)*(x-1.08)/(1.10-1.08));
     else if ( x >= 1.10 )
      mttf=gama+ita*(0.951351-(0.951351-0.943590)*(x-1.10)/(1.12-1.10));
     else if ( x >= 1.12 )
      mttf=gama+ita*(0.943590-(0.943590-0.936416)*(x-1.12)/(1.14-1.12));
     else if ( x >= 1.14 )
      mttf=gama+ita*(0.936416-(0.936416-0.929803)*(x-1.14)/(1.16-1.14));
     else if ( x >= 1.16 )
      mttf=gama+ita*(0.929803-(0.929803-0.923728)*(x-1.16)/(1.18-1.16));
     else if ( x >= 1.18 )
      mttf=gama+ita*(0.923728-(0.923728-0.918169)*(x-1.18)/(1.20-1.18));
     else if ( x >= 1.20 )
      mttf=gama+ita*(0.918169-(0.918169-0.913106)*(x-1.20)/(1.22-1.20));
     else if ( x >= 1.22 )
```

```
 mttf=gama+ita*(0.913106-(0.913106-0.908521)*(x-1.22)/(1.24-1.22));
else if ( x >= 1.24 )
 mttf=gama+ita*(0.908521-(0.908521-0.904397)*(x-1.24)/(1.26-1.24));
else if ( x >= 1.26 )
 mttf=gama+ita*(0.904397-(0.904397-0.900718)*(x-1.26)/(1.28-1.26));
else if ( x >= 1.28 )
 mttf=gama+ita*(0.900718-(0.900718-0.897471)*(x-1.28)/(1.30-1.28));
else if ( x >= 1.30 )
 mttf=gama+ita*(0.897471-(0.897471-0.894640)*(x-1.30)/(1.32-1.30));
else if ( x >= 1.32 )
 mttf=gama+ita*(0.894640-(0.894640-0.892216)*(x-1.32)/(1.34-1.32));
else if ( x >= 1.34 )
 mttf=gama+ita*(0.892216-(0.892216-0.890185)*(x-1.34)/(1.36-1.34));
else if ( x >= 1.36 )
 mttf=gama+ita*(0.890185-(0.890185-0.888537)*(x-1.36)/(1.38-1.36));
else if ( x >= 1.38 )
 mttf=gama+ita*(0.888537-(0.888537-0.887264)*(x-1.38)/(1.40-1.38));
else if ( x >= 1.40 )
 mttf=gama+ita*(0.887264-(0.887264-0.886356)*(x-1.40)/(1.42-1.40));
else if ( x >= 1.42 )
 mttf=gama+ita*(0.886356-(0.886356-0.885805)*(x-1.42)/(1.44-1.42));
else if ( x >= 1.44 )
 mttf=gama+ita*(0.885805-(0.885805-0.885604)*(x-1.44)/(1.46-1.44));
else if ( x >= 1.46 )
 mttf=gama+ita*(0.885604-(0.885604-0.885747)*(x-1.46)/(1.48-1.46));
else if ( x >= 1.48 )
 mttf=gama+ita*(0.885747-(0.885747-0.886227)*(x-1.48)/(1.50-1.48));
else if ( x >= 1.50 )
 mttf=gama+ita*(0.886227-(0.886227-0.887039)*(x-1.50)/(1.52-1.50));
else if ( x >= 1.52 )
 mttf=gama+ita*(0.887039-(0.887039-0.888178)*(x-1.52)/(1.54-1.52));
else if ( x >= 1.54 )
 mttf=gama+ita*(0.888178-(0.888178-0.889639)*(x-1.54)/(1.56-1.54));
else if ( x >= 1.56 )
 mttf=gama+ita*(0.889639-(0.889639-0.891420)*(x-1.56)/(1.58-1.56));
else if ( x >= 1.58 )
 mttf=gama+ita*(0.891420-(0.891420-0.893515)*(x-1.58)/(1.60-1.58));
else if ( x >= 1.60 )
 mttf=gama+ita*(0.893515-(0.893515-0.895924)*(x-1.60)/(1.62-1.60));
else if ( x >= 1.62 )
 mttf=gama+ita*(0.895924-(0.895924-0.898642)*(x-1.62)/(1.64-1.62));
else if ( x >= 1.64 )
 mttf=gama+ita*(0.898642-(0.898642-0.901668)*(x-1.64)/(1.66-1.64));
else if ( x >= 1.66 )
 mttf=gama+ita*(0.901668-(0.901668-0.905001)*(x-1.66)/(1.68-1.66));
```

```
else if ( x >= 1.68 )
 mttf=gama+ita*(0.905001-(0.905001-0.908639)*(x-1.68)/(1.70-1.68));
else if ( x >= 1.70 )
 mttf=gama+ita*(0.908639-(0.908639-0.912581)*(x-1.70)/(1.72-1.70));
else if ( x >= 1.72 )
 mttf=gama+ita*(0.912581-(0.912581-0.916826)*(x-1.72)/(1.74-1.72));
else if ( x >= 1.74 )
 mttf=gama+ita*(0.916826-(0.916826-0.921375)*(x-1.74)/(1.76-1.74));
else if ( x >= 1.76 )
 mttf=gama+ita*(0.921375-(0.921375-0.926227)*(x-1.76)/(1.78-1.76));
else if ( x >= 1.78 )
 mttf=gama+ita*(0.926227-(0.926227-0.931384)*(x-1.78)/(1.80-1.78));
else if ( x >= 1.80 )
 mttf=gama+ita*(0.931384-(0.931384-0.936845)*(x-1.80)/(1.82-1.80));
else if ( x >= 1.82 )
 mttf=gama+ita*(0.936845-(0.936845-0.942612)*(x-1.82)/(1.84-1.82));
else if ( x >= 1.84 )
 mttf=gama+ita*(0.942612-(0.942612-0.948687)*(x-1.84)/(1.86-1.84));
else if ( x >= 1.86 )
 mttf=gama+ita*(0.948687-(0.948687-0.955071)*(x-1.86)/(1.88-1.86));
else if ( x >= 1.88 )
 mttf=gama+ita*(0.955071-(0.955071-0.961766)*(x-1.88)/(1.90-1.88));
else if ( x >= 1.90 )
 mttf=gama+ita*(0.961766-(0.961766-0.968774)*(x-1.90)/(1.92-1.90));
else if ( x >= 1.92 )
 mttf=gama+ita*(0.968774-(0.968774-0.976099)*(x-1.92)/(1.94-1.92));
else if ( x >= 1.94 )
 mttf=gama+ita*(0.976099-(0.976099-0.983743)*(x-1.94)/(1.96-1.94));
else if ( x >= 1.96 )
 mttf=gama+ita*(0.983743-(0.983743-0.991708)*(x-1.96)/(1.98-1.96));
else if ( x >= 1.98 )
 mttf=gama+ita*(0.991708-(0.991708-1.000000)*(x-1.98)/(2.00-1.98));
// else if ( x = 2.00 )
// mttf=gama+ita*(0.931384-(0.931384-0.936845)*(x-1.80)/(1.82-1.80))

return int(mttf);
}


void update_time(equipment& eqpt, int time)
 {
 if (eqpt.est==0)
 {
  eqpt.rnt+=time;
  eqpt.dnt=0;
```

```
    }
    else if (eqpt.est==2)
    {
     eqpt.dnt+=time;
     eqpt.rnt=0;
    }
   }
```

a1= 0.85
a2= 1.1
SIMULATION TIME = 9319
loop count= 82
PARTIAL SHUT DOWN TIME = 146
TOTAL SHUT DOWN TIME = 2924
UNIT AVAILABILITY = 0.678399

s/n	Rnt	Dnt	Av
0	5048	985	0.836731
1	5048	671	0.882672
2	4902	492	0.908788
3	3054	72	0.976967
4	3515	167	0.954644
5	4531	246	0.948503
6	4550	224	0.953079
7	3965	96	0.97636
8	2729	423	0.865799
9	4435	240	0.948663
10	3395	570	0.856242
11	3180	192	0.943061
12	5048	1471	0.774352
13	5048	2924	0.633216

a1= 0.85
a2= 1.2
SIMULATION TIME = 9005
loop count= 70
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 3030
UNIT AVAILABILITY = 0.66352

s/n	Rnt	Dnt	Av
0	4864	954	0.836026
1	4864	666	0.879566
2	4864	540	0.900074
3	3241	143	0.957742
4	4772	293	0.942152
5	5226	72	0.98641
6	5115	72	0.986119
7	2665	72	0.973694
8	2181	72	0.968043
9	3872	192	0.952756
10	3501	973	0.782521
11	2601	144	0.947541
12	4864	1422	0.773783
13	4864	3030	0.616164

a1= 0.85
a2= 1.25
SIMULATION TIME = 9051
loop count= 65
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 3030
UNIT AVAILABILITY = 0.66523

s/n	Rnt	Dnt	Av
0	4864	954	0.836026
1	4864	666	0.879566
2	4864	540	0.900074
3	4245	297	0.93461
4	3855	167	0.958478
5	3630	48	0.986949
6	3552	48	0.986667
7	2703	72	0.974054
8	2271	72	0.96927
9	4032	192	0.954545
10	3479	973	0.781447
11	2709	144	0.949527
12	4864	1422	0.773783
13	4864	3030	0.616164

a1= 0.85
a2= 1.3
SIMULATION TIME = 9346
loop count= 68
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2948
UNIT AVAILABILITY = 0.684571

s/n	Rnt	Dnt	Av
0	4864	962	0.834878
1	4864	648	0.882438
2	4864	538	0.900407
3	2222	189	0.921609
4	5444	172	0.969373
5	3774	48	0.987441
6	5541	72	0.987173
7	2811	72	0.975026
8	2361	72	0.970407
9	4166	529	0.887327
10	5228	618	0.894287
11	2955	96	0.968535
12	4864	1510	0.7631
13	4864	2948	0.622632

a1= 0.9
a2= 1.1
SIMULATION TIME = 9433
loop count= 82
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2975
UNIT AVAILABILITY = 0.684618

s/n	Rnt	Dnt	Av
0	4843	983	0.831274
1	4843	702	0.873399
2	4843	469	0.911709
3	3054	72	0.976967
4	4992	96	0.981132
5	4406	605	0.879266
6	4332	383	0.91877
7	2379	72	0.970624
8	3330	120	0.965217
9	4435	240	0.948663
10	3395	499	0.871854
11	3975	192	0.953924
12	4843	1461	0.768242
13	4843	2975	0.619468

a1= 0.9
a2= 1.2
SIMULATION TIME = 9032
loop count= 74
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2951
UNIT AVAILABILITY = 0.673273

s/n	Rnt	Dnt	Av
0	4943	917	0.843515
1	4943	692	0.877196
2	4943	516	0.905477
3	2130	143	0.937088
4	4086	72	0.982684
5	5226	72	0.98641
6	5115	72	0.986119
7	2595	72	0.973003
8	2908	96	0.968043
9	3968	192	0.953846
10	4776	965	0.831911
11	2601	144	0.947541
12	4943	1508	0.766238
13	4943	2951	0.626172

a1= 0.9
a2= 1.25
SIMULATION TIME = 9098
loop count= 68
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2948
UNIT AVAILABILITY = 0.675973

s/n	Rnt	Dnt	Av
0	4915	971	0.835032
1	4915	650	0.883199
2	4915	533	0.902166
3	3333	167	0.952286
4	4254	72	0.983356
5	5445	48	0.991262
6	5328	72	0.986667
7	1802	48	0.974054
8	3028	96	0.96927
9	4032	192	0.954545
10	3458	840	0.80456
11	2709	144	0.949527
12	4915	1531	0.762488
13	4915	2948	0.62508

a1= 0.9
a2= 1.3
SIMULATION TIME = 9066
loop count= 66
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2948
UNIT AVAILABILITY = 0.674829

s/n	Rnt	Dnt	Av
0	4915	1019	0.828278
1	4915	591	0.892663
2	4915	531	0.902497
3	4603	355	0.928399
4	4425	72	0.983989
5	3774	48	0.987441
6	5541	72	0.987173
7	1874	48	0.975026
8	2361	72	0.970407
9	4166	443	0.903884
10	5228	547	0.905281
11	1969	741	0.726568
12	4915	1531	0.762488
13	4915	2948	0.62508

a1= 0.95
a2= 1.1
SIMULATION TIME = 9019
loop count= 82
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2903
UNIT AVAILABILITY = 0.678124

s/n	Rnt	Dnt	Av
0	4949	1026	0.828285
1	4949	644	0.884856
2	4949	495	0.909074
3	3054	72	0.976967
4	4992	96	0.981132
5	4666	250	0.949146
6	4563	201	0.957809
7	2379	72	0.970624
8	3395	447	0.883654
9	4343	608	0.877197
10	4752	192	0.961165
11	2481	144	0.945143
12	4949	1443	0.774249
13	4949	2903	0.630285

a1= 0.95
a2= 1.2
SIMULATION TIME = 9082
loop count= 73
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2821
UNIT AVAILABILITY = 0.689386

s/n	Rnt	Dnt	Av
0	4949	964	0.836969
1	4949	710	0.874536
2	4949	515	0.905747
3	2130	143	0.937088
4	4086	72	0.982684
5	5226	72	0.98641
6	5115	72	0.986119
7	2595	72	0.973003
8	2908	96	0.968043
9	3872	192	0.952756
10	5087	588	0.896388
11	3467	789	0.814615
12	4949	1531	0.763735
13	4949	2821	0.636937

a1= 0.95
a2= 1.25
SIMULATION TIME = 9063
loop count= 71
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2768
UNIT AVAILABILITY = 0.694582

s/n	Rnt	Dnt	Av
0	5011	971	0.83768
1	5011	703	0.876969
2	5011	535	0.903534
3	3333	167	0.952286
4	4254	72	0.983356
5	5445	72	0.986949
6	5328	72	0.986667
7	1802	48	0.974054
8	3028	96	0.96927
9	4032	192	0.954545
10	5173	482	0.914766
11	2709	144	0.949527
12	5011	1495	0.770212
13	5011	2768	0.64417

a1= 0.95
a2= 1.3
SIMULATION TIME = 9161
loop count= 71
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2766
UNIT AVAILABILITY = 0.698068

s/n	Rnt	Dnt	Av
0	5011	996	0.834193
1	5011	657	0.884086
2	5011	513	0.907133
3	3425	213	0.941451
4	4425	72	0.983989
5	5661	48	0.991592
6	5541	72	0.987173
7	1874	48	0.975026
8	3148	96	0.970407
9	4166	443	0.903884
10	4209	144	0.966919
11	2817	144	0.951368
12	5011	1465	0.77378
13	5011	2766	0.644336

a1= 1
a2= 1.1
SIMULATION TIME = 9005
loop count= 84
PARTIAL SHUT DOWN TIME = 48
TOTAL SHUT DOWN TIME = 2780
UNIT AVAILABILITY = 0.688617

s/n	Rnt	Dnt	Av
0	4949	1026	0.828285
1	4949	644	0.884856
2	4949	495	0.909074
3	3054	72	0.976967
4	4992	96	0.981132
5	4666	250	0.949146
6	4563	201	0.957809
7	2379	72	0.970624
8	3996	120	0.970845
9	3644	524	0.87428
10	4752	192	0.961165
11	3276	192	0.944637
12	4949	1478	0.770033
13	4949	2780	0.640316

a1= 1
a2= 1.2
SIMULATION TIME = 9082
loop count= 73
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2821
UNIT AVAILABILITY = 0.689386

s/n	Rnt	Dnt	Av
0	4949	964	0.836969
1	4949	710	0.874536
2	4949	515	0.905747
3	2130	143	0.937088
4	4086	72	0.982684
5	5226	72	0.98641
6	5115	72	0.986119
7	2595	72	0.973003
8	2908	96	0.968043
9	3872	192	0.952756
10	5087	588	0.896388
11	3467	789	0.814615
12	4949	1531	0.763735
13	4949	2821	0.636937

a1= 1
a2= 1.25
SIMULATION TIME = 9063
loop count= 71
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2768
UNIT AVAILABILITY = 0.694582

s/n	Rnt	Dnt	Av
0	5011	971	0.83768
1	5011	703	0.876969
2	5011	535	0.903534
3	3333	167	0.952286
4	4254	72	0.983356
5	5445	72	0.986949
6	5328	72	0.986667
7	1802	48	0.974054
8	3028	96	0.96927
9	4032	192	0.954545
10	5173	482	0.914766
11	2709	144	0.949527
12	5011	1495	0.770212
13	5011	2768	0.64417

a1= 1
a2= 1.3
SIMULATION TIME = 9161
loop count= 71
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2766
UNIT AVAILABILITY = 0.689068

s/n	Rnt	Dnt	Av
0	5011	996	0.834193
1	5011	657	0.884086
2	5011	513	0.907133
3	3425	213	0.941451
4	4425	72	0.983989
5	5661	48	0.991592
6	5541	72	0.987173
7	1874	48	0.975026
8	3148	96	0.970407
9	4166	443	0.903884
10	4209	144	0.966919
11	2817	144	0.951368
12	5011	1465	0.77378
13	5011	2766	0.644336

a1= 1.1
a2= 1.1
SIMULATION TIME = 9022
loop count= 79
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2906
UNIT AVAILABILITY = 0.677898

s/n	Rnt	Dnt	Av
0	4942	1015	0.829612
1	4942	651	0.883605
2	4942	490	0.909794
3	3054	72	0.976967
4	4992	96	0.981132
5	4791	72	0.985194
6	4689	72	0.984877
7	2379	72	0.970624
8	3330	120	0.965217
9	3548	192	0.948663
10	4752	192	0.961165
11	3276	192	0.944637
12	4942	1379	0.781838
13	4942	2906	0.629715

a1= 1.1
a2= 1.2
SIMULATION TIME = 9073
loop count= 73
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2908
UNIT AVAILABILITY = 0.679489

s/n	Rnt	Dnt	Av
0	4942	969	0.836068
1	4942	711	0.874226
2	4942	510	0.906456
3	2130	143	0.937088
4	4086	72	0.982684
5	5226	72	0.98641
6	5115	72	0.986119
7	2595	72	0.973003
8	2908	96	0.968043
9	3872	192	0.952756
10	5087	540	0.904034
11	2601	144	0.947541
12	4942	1409	0.778145
13	4942	2908	0.629554

a1= 1.1
a2= 1.25
SIMULATION TIME = 9009
loop count= 67
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2835
UNIT AVAILABILITY = 0.685315

s/n	Rnt	Dnt	Av
0	4658	979	0.826326
1	4658	538	0.896459
2	4658	544	0.895425
3	4397	320	0.93216
4	4254	72	0.983356
5	5445	72	0.986949
6	5328	72	0.986667
7	1802	48	0.974054
8	2271	72	0.96927
9	4032	192	0.954545
10	3924	546	0.877852
11	2882	144	0.952412
12	3034	644	0.824905
13	4658	2835	0.621647

a1= 1.1
a2= 1.3
SIMULATION TIME = 9161
loop count= 71
PARTIAL SHUT DOWN TIME = 0
TOTAL SHUT DOWN TIME = 2766
UNIT AVAILABILITY = 0.698068

s/n	Rnt	Dnt	Av
0	5011	996	0.834193
1	5011	657	0.884086
2	5011	513	0.907133
3	3425	213	0.941451
4	4425	72	0.983989
5	5661	48	0.991592
6	5541	72	0.987173
7	1874	48	0.975026
8	3148	96	0.970407
9	4166	443	0.903884
10	4209	144	0.966919
11	2817	144	0.951368
12	5011	1465	0.77378
13	5011	2766	0.644336

REFERENCE

1. C.J. Baldwin, D.P. Gaver and C.H. Hoffman, "Mathematical Models for use in the Simulation of Power Generation Outages : 1-Fundamental Consideration", AIEE Trans. on PAS, vol-78, pp 1251-1258, Dec. 1959

2. Report of AIEE Subcommittee on Application of Probability Methods, "Application of Probability Methods to Generating Capacity Problem", AIEE Trans. on PAS, vol-79, pt. II, pp 1165-1182 Feb. 1961

3. S.R. Calabro and B.G. Horowitz, " Evaluating Equipment Reliability Using Techniques Based on KAPPA Square Statistic", IEEE Trans. PAS, Vol PAS-100, pp 4074-4082, Aug. 1981

4. R.H. Koppe, R.W. Keller and K.R. Van Howe, "A Power Plant Availability Improvement Methodology – Based on the New NERC Generating Availability Data System (GADS)" IEEE Trans. PAS, Vol PAS-101, pp 2537-2542, Aug. 1982

5. N.E. Chang, "Application of Availability Analysis Techniques to Improve Power Plant Productivity"' IEEE Trans. PAS, Vol PAS-102, pp 1521-2529, June. 1983

6. A.D. Ratton, J.H. Blackstone and N.J. Ballu, " A Monte Carlo approach to Reliability Modeling of Generating Systems Recognizing Operating Considerations", IEEE Trans, on Power System, vol-3, No-3, pp 1174-1180, Aug. 1988

7. O. Bertoldi, L.Salvaderi and S. Sclacino, " Monte Carlo Approach in Planning Studies : an Application to IEEE RTS", IEEE Trans. on Power System, vol-3, No-3, pp 1146-1154, Aug. 1988

8. C.C. Fong. R. Billinton, R.O. Gunderson, P.M. Oniel, J. Raksany, A.W. Schneider, Jr, and B. Silverstein, " Bulk System Reliability-Measurement and Indices" IEEE Trans on Power Apparatus & System, vol-4, No-3, pp 829-835, Aug. 1989

9. Sandra A Sandri. Didier Dabois and Henh W. Kalsbeck, "Elicitation, Assessment and pooling of expert Judgements Using Possibility Theory" IEEE Trans on Fuzzy System, Vol-3, No-3, pp 313-335, Aug. 1995

10. Yasser El Sobaty and M.A. Ismail, " Fuzzy Clustering or Symbolic Data" IEEE Trans on Fuzzy System, Vol-6 No-2, May. 1998

11. K.K. Das, D. Acharya, " On the Choice of Optimal Preventive Maintenance Policy For Maximizing Generation of Thermal Power Plants", IEEE Tran On energy Conversions, Vol-14, No-4, pp 1351-1357, Dec.99

12. "Applied Reliability Assessment in Electric Power System", (book), Ed. By : Roy Billinton, Ronald N. Allan, Luigi Salvaderi. A Volume in the IEEE PRESS Selected Reprint Series, Prepared Under the Sponsorship of the IEEE Power Engineering Society, 1991

13. "Reliability Modeling in Electrical Power System",(book), Ed. By : J. Endrenyi, Wiley Intersciene Publication, 1979.