PREVENTIVE MAINTENANCE POLICY FOR HYDRO POWER PLANTS USING FUZZY SET THEORY

A DISSERTATION

submitted in partial fulfillment of the
requirements for the award of the degree
of
MASTER OF ENGINEERING
in
WATER RESOURCES DEVELOPMENT
(Hydro Electric System Engineering Management)

By
HIMADRI BOSE

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

January, 2001

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the dissertation entitled, **"PREVENTIVE MAINTENANCE POLICY FOR HYDRO POWER PLANTS USING FUZZY SET THEORY"**, in partial fulfilment of the requirements for the award of Degree of Master of Engineering in **Hydro Electric System Engineering and Management** submitted in the Water Resources Development Training Centre, University of Roorkee, Roorkee is an authentic record of my own work carried out since 16th July, 2000 till the date of submission under the supervision of **Prof. Devadutta Das**, Director, WRDTC and **Dr. N.P. Padhy**, Faculty, Electrical Engineering Department, University of Roorkee, Roorkee, India.

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

Dated: January, 08, 2001

Himadri Bose

(HIMADARI BOSE)

This is to certify that the above statement made by the candidate is correct to the best our knowledge.

N Prasad 08/01/2001

(DR. N.P. PADHY)
Faculty
Elect. Engg. Deptt.
University of Roorkee
Roorkee – 247667

Devadutta Das

(PROF. DEVADUTTA DAS)
Director
W.R.D.T.C.
University of Roorkee
Roorkee - 247667

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude and sincere thanks to **Prof. Devadutta Das**, Professor, WRDTC, **Dr. N.P. Padhy**, Assistant Professor, Electrical Engineering Department, University of Roorkee, Roorkee for their valuable guidance, advice and encouragement during the preparation of this thesis.

I would also like to place on record my sincere thanks to the Faculty Members, WRDTC for their care and help and teachers of Electrical Engineering Department of University of Roorkee who extended their help and guidance during this M.E. Course.

I would also like to thanks for the facilities of WRDTC Library, Computer Centre and Central Library which, helped me in a great way to prepare this thesis.

I am very grateful to West Bengal State Electricity Board which gave me this wonderful opportunity to attend the M.E. Course in WRDTC, University of Roorkee, Roorkee.

I also thankfully acknowledge to all my Trainee Officers colleagues of 44th WRD and 20th IWM for their kind co-operation from time to time.

Lastly, I want to pay regards to my parents who, are the constant inspirations motivations in my life.

Himadri Bose

(HIMADRI BOSE)

Trainee Officer (WRD- Elect.)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

α ----------- degree of possibility
λ ----------- levels of confidence
p ----------- weights of nested sets
ε ---------- residual uncertainty
ν ----------- linguistic variable
i,j,k --------- arbitrary identifier (indices)
$\mu_A(x)$ -------- membership grade of x in fuzzy set A
π_ν ---------- possibility distribution attached to parameter ν
Π ---------- possibility measure
N --------- necessity measure
Σ ---------- summation
$\forall$ ---------- for all (universal quantifier)
$x \in A$ -------- element x belongs to crisp set A
$X \rightarrow Y$ ------ function from X to Y
$A \subseteq B$ ------- set inclusion
$A \cap B$ ------- set intersection
$|A|$ --------- cardinality of crisp or fuzzy set A
$\bar{A}$ ----------- complement of A
max $(a_1,a_2,----,a_n)$ --- maximum of $a_1,a_2,----,a_n$
min $(a_1,a_2,----,a_n)$ --- minimum of $a_1,a_2,----,a_n$
sup (A) ------- supremum of A
inf (A) ------- infimum of A
Cons (A) ------ sum of individual conflicts of evidential claims with respect to a particular A in evidence theory
$\{x \mid p(x)\}$ ------- set determined by property p

SYNOPSIS

In countries like India, there is a huge gap between the electricity generation and demand. To meet this demand a bulk investment is needed to install new generating plants immediately. Our country's economy does not permit such a bulk investment immediately. Even if funds are arranged, it will take a quite long gestation period to realize the benefits. Hence, besides installing new generating stations it is needed to enhance the efficiency of the already installed power generating stations. One way to increase the efficiency is adopting a proper preventive maintenance policy for optimising the generations.

This study is intended to produce a simple yet realistic process for evaluation of an effective preventive maintenance policy, which yields optimal energy generation. To attain this objective, an opportunistic preventive maintenance model is incorporated which resembles the practical situation. This maintenance model is realized by Monte Carlo simulation method to find the impact of preventive maintenance on productivity of hydro generating plants.

The history of equipment outages and repair is the basic requirements of any reliability study. In India, hardly any data of equipment outage and repair is available. Moreover the equipment configuration and ratings of hydro power plants differ from installation to installation. A procedure has been suggested to collect information from human experts, assessing them and aggregating them to obtain realistic probability distribution of equipment outage and repair. This evaluation process has been carried out within the basic framework of possibility set theory and fuzzy set theory to take into account the uncertainty involved of human originated information.

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. Electric power utility's have always employed preventive maintenance programs to keep their equipment in good working condition for as long as it was economic. At the present when system extensions are heavily constrained and the purchase of new and better equipment may not be feasible, the role of maintenance is particularly significant. Thus, an efficient maintenance program has become an important part of what is often called asset management.

The main objective of this study is to develop practical model for preventive maintenance scheduling for hydro generating plants so that the plant availability can be maximized. To develop a mathematical model for maintenance we required some basic outage and repair data of all critical equipments.

System planners and plant managers in India are crippled due to unavailability of any such records. In this study a procedure has been devised for processing human-originated information. This procedure can help the power system analysts and power plant managers to generate the much-needed input data for analysing their maintenance model and take some optimal decisions.

The study can be broadly divided into two parts :

(i) Formulation of appropriate process to evaluate probability distribution of outage and repair of critical equipments from the expert originated information based on possibility and fuzzy set theory.

(ii) Evaluating a proper simulation model to plan preventive maintenance policy for optimizing the availability of hydro generating plants.

It is important to note that study has following limitations :

- Cost factor is not taken into account in this study.
- The model deals only with replacement policy and does not take into account the quantitative connection between reliability and maintenance.
- The model does not take into account the effect of the whole power system and considers whatever power is generated can be absorbed by the system.

The chapterwise contents are given below :

Chapter – 2 : Literature review

Chapter – 3 : Preventive maintenance model of hydro generating plant and simulation model.

Chapter – 4 : Methodology of processing human originated information within the possibilistic frame work and fuzzy set theory.

Chapter – 5 : Development of Software for simulation.

Chapter – 6 : Results and discussions.

LITERATURE REVIEW

2.1 JUSTIFICATION OF PREVENTIVE MAINTENANCE

With cost and difficulty of constructing new generating units increasing, utilities are being forced to focus more efforts on improving the productivity of existing units. In simplest terms, the utilitiy's 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 [11]. Preventive maintenance of generating equipment is required in order to reduce the risk of capacity outage and improve the overall availability of such units. Preventive maintenance of generating units in each individual area in a pool is an important factor of overall system reliability. The required availability-goal can be achieved by reliability and maintainability analysis. Chang [10] has given a co-hesive and comprehensive approach to improve power plant availability. Maintenance scheduling should permit the required work to be accomplished but not create excessive risk to the system. During the maintenance period, the unit capacity is not available to the system, therefore the total install capacity of the system is decreased which creates an increase in system risk. This risk should not exceed the acceptable planning value at any time during the annual load cycle [9]. Since the object of any betterment program is to reduce the impact of a system on unit productivity, mathematical models are needed which predict the productivity impact of various combinations of system configuration and component performance.

2.2 MAINTENANCE MODEL

The purpose of this study is to develop a suitable maintenance model of hydro power plant that 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 right 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 [1] has made an attempt to given a simple model for evaluating availability of hydro power plants. This model deals with opportunistic maintenance policy for a repairable system with several types of units, each with its own increasing hazard rate. Preventive replacement is carried out when the hazard rate reaches an upper limit. But the maintenance model does not incorporate any minimal repair policy.

Endrenyi et al [3] given a mathematical model incorporating the concept of "maintenance when needed". The basic idea in this model is the probabilistic representation of the deterioration process through discrete stages and describing the impact on the reliability of gradually deteriorating equipment of periodic inspections which can lead to various possible maintenance modes.

2.3 EVALUATION METHODS OF SYSTEM RELIABILITY

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 plan system. It has been found that for many situations a straightforward simulation of the system (Monte Carlo Analysis) has several advantages [11]. 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.
- It provides solution to problem with so many variables that can not be described by specific mathematical formulae or by single probability equations.

Monte Carlo simulation methods are applied to the model to simulate random occurrences such as forced outages, random variations in daily load etc. The logic of the system operation or human elements is built into the model. The model than simulates the random events that occur and the human decisions made; therefore the system is operated and planned by model in a manner that closely approaches reality [14].

EPRI (Electric Power Research Institute, Palo Alto) project RP 1534-1,2 reveals most detailed and accurate modeling of complex operating consideration presently requires the use of Monte Carlo Simulation [8]. 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 BASIC INDICES

A proper indices of reliability should be chosen when reliability methods are applied for comparative analysis of alternate maintenance scheduling programme and take the best decision. For those applying prediction techniques, measurement of the system reliability has the potential of being a validation tools for the predictive models. Several fundamental indices are proposed in a report by a IEEE working group on measurement indices of reliability [6]. Some of them are given below:

$$\text{Interruption frequency} = \frac{\text{Number of interruptions}}{\text{Period (years)}}$$

$$\text{Annual interrupion duration} = \frac{\text{Sum of interruption durations}}{\text{Period (years)}}$$

$$\text{Average duration per interruption} = \frac{\text{Sum of inrruption duration}}{\text{Number of interruptions}}$$

$$\text{Annual load interruption} = \frac{\text{Sum of MW interrupted}}{\text{Period (years)}}$$

$$\text{Annual unsupplied energy} = \frac{\text{Sum of unsupplied energy}}{\text{Period (years)}}$$

This indices can be expanded based on individual utility needs. Any one of the them can be adopted based on individual utility's approach, requirements, as well as the flexibility to expand on essentially the same maintenance schedule for each individual utility [13].

2.5 DATA REQUIREMENTS

There were suggestions 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 reports on outage experience in 1954 and in 1957 [13]. Power system equipment outage data are well collected in Western Countries.

A bibliography of equipment outage data is available in [5]. No such endeavor has been taken up in our country like (India) to collect the equipment outage data and analyze them. It is not realistic to use the data compiled by organizations of western countries in India, as the manufacturing and maintenance practices are quite different along with the operating conditions.

The question arises 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 considerable time.

One possible solution can be, using information originating from expert in the field of reliability, engineers and manufacturers. To develop a outage distribution of equipments within a possibilistic framework. The main reason for adopting such a framework is that possibility theory offers a simple theory of uncertainty that explicitly take into account the lack of precision of expert knowledge. The whole task can be divided into three parts, collection of information from experts, assessment of experts and clustering of data supplied by expert. Experts can be evaluated in term of accuracy and level of precision, respectively measured by membership grades and fuzzy cardinality [4]. The clustering of data can be done by possibility theory such that objects within the same cluster have a high degree of similarity with expert to precision and accuracy [2].

2.6 CONCLUSION

In India there is a huge gap between the electricity demand and generation. To meet the demand a large number of generating plants are to be installed which requires a huge amount of investment. With the scarcity of funds this is not possible. One of the viable alternatives to reduce the energy gap is by increasing the plant availability of the presently existing generating stations. This can be accomplished by proper maintenance of plant equipments to extend by equipment lifetime.

The power plant managers need to choose some preventive maintenance policy to achieve the maximum availability of the plant. A simple yet reliable maintenance model is to be developed to predict the productivity impact of the plant on various system configurations and component performance by the plant managers or power system analyst.

Among different evaluation methods of system reliability, Monte Carlo simulation method seems to be suitable for hydro power plants with so many variables and complex systems of operation. As hydro power stations are energy restricted generation plants the interruption frequency or duration do not reflect the reliability of the plant, therefore, annual unsupplied energy can be used as reliability index for comparison purposes between different configurations and policies.

Basic data required for reliability study are scarce in India. Hence an attempt can be taken to formulate a methodology to gather the required data from the human experts. This can be done within the possibilistic framework to take care of the uncertainties involved within. Due to scarcity of real time data the model can be based on replacement policy only and does not incorporate deterioration process through discrete stages.

PREVENTIVE MAINTENANCE MODEL OF HYDRO POWER PLANT

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 can not 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 - reduces 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 Replacement :* 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 component. 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 HYDRO POWER PLANT

The main objective of adopting a preventive maintenance policy for a hydro power plant is to increase the availability of power plant. Though it can also be adopted on the basis of economics, the most economical solution may not necessarily provide the highest degree of reliability. It is a perennial problem in the decision making process of electric utility managers. As in many other areas, costs an reliability must be balanced. 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 are preventively maintained (at an age $\geq T_1$) 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 generation 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 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 simulation method. Monte Carlo simulation provides a flexible tool for incorporating modeling details, such as operating considerations and constraints in the system. This is an approach of actual realization of the process which 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 can be solved by parallel processing techniques. This only difficulty of Monte

Carlo method is out weighed by its advantages and we adopt this method in our study.

3.3.1 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 (cumulative 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.

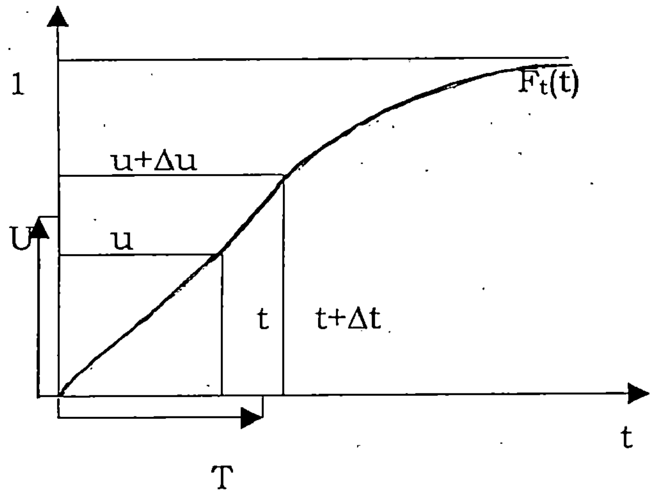


Fig. 3.1 : The Random Variable T and the Associated Random Variable U with Uniform Distribution

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

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

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

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

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

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

This result indicates that U has a uniform distribution between 0 and 1 (or more formally, f_u (u) = 1, 0 < u ≤ 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.

$$t = F_1^{-1}(u) \tag{3.4}$$

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

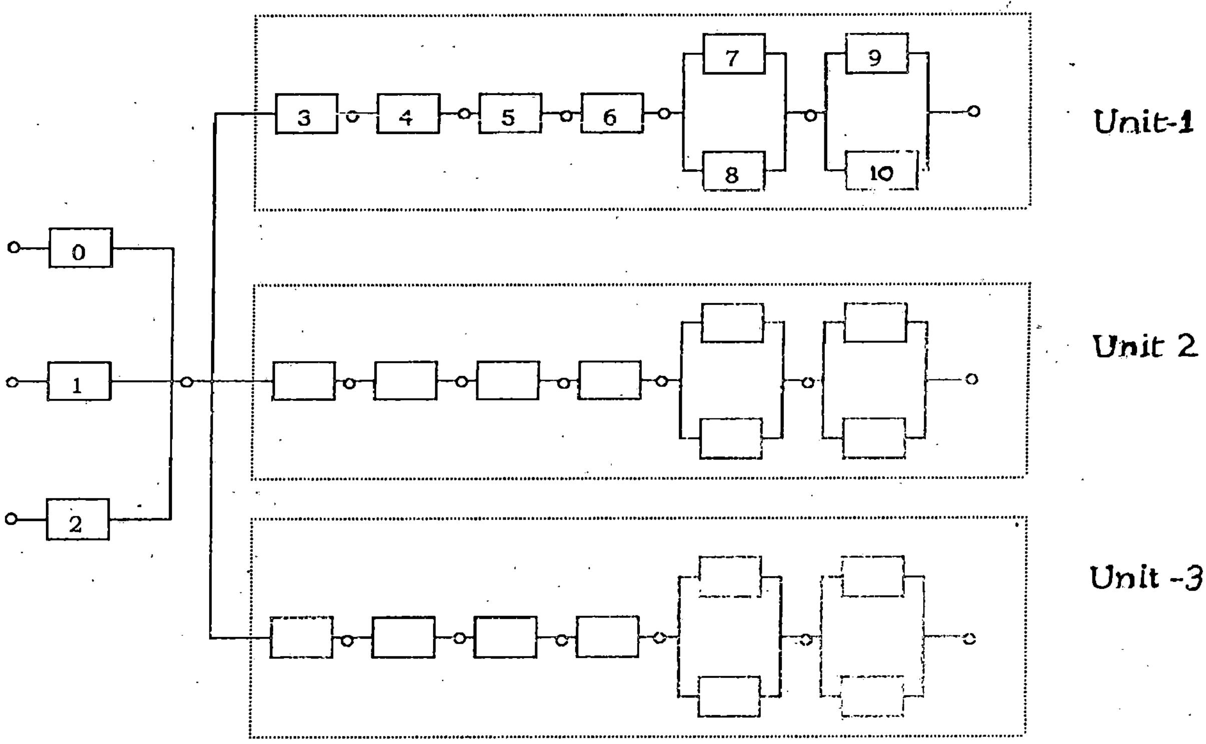
Returning to the simulated realisation of the life history of a system, one proceeds by creating separate life histories for all the components, and examining all the system failure. Each time a equipment fails system failure is encountered, the human decisions are taken and checked for system failures, its duration is registered, and the failure count is advanced by one. If there is a deficiency in generation that is added to the expected unserved energy. Finally the yearly expected unserved energy for the system (generation plant) is obtained.

3.4 SYSTEM CONFIGURATION

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

The plant having three units and three cooling water pumps, serving all the three units on a station basis. Cooling water pump requirements are as following :

- If one unit is running one cooling water pump is required.
- If two or three units are in operation than two cooling water pumps are required.
- One cooling water pump is always in standby mode.



Identity of equipments-

0,1,2 ⟶ Cooling Water Pumps

3 ⟶ Bearing

4 ⟶ Draft Tube and Runner

5 ⟶ Governor

6 ⟶ Electrical Auxliaries

7,8 ⟶ Lubrication Oil Pumps

9,10 ⟶ Governor Oil Pumps

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

The cooling water pumps are started or taken out of service as and when required. One governor oil pump and one lubrication oil pump should run for satisfactory operation of each unit. There is one standby pump for each of the governor oil pump and lubrication oil pump.

In the plant model it is assumed that the head is fixed and is equal to 7 m. The rating of each turbine is 7.5 MW (megawatt). Each unit is loaded at a maximum of 7.5 MW and if any excess water is available another unit is taken into service. Whenever water availability recedes units running in excess are shutdown, so that reduced number of unit(s) can utilize the whole amount of water.

3.5 RELIABILITY INDEX

Each hourly expected energy generation as per water availability considering all equipments of the plant running is calculated. The equipment conditions of the plant are monitored and any deficiency in generation is also monitored and recorded hourly. The deficiencies in hourly generation are summed up for the whole simulation time. This is then divided by the number of years of simulation which gives annual expected unserved energy, which serves as the reliability index. For different sets of a_1 and a_2 the simulation can be done and the set corresponding to the lowest value of annual unserved energy gives the optimal value of a_1 and a_2.

$$AUE = \frac{\sum_{i=1}^{MST}(EEG_i - AEG_i)}{N}.$$

where,

AUE = Annual unsupplied energy

MST = Maximum simulation time

EEG_i = Expected energy generation at ith hour.

AEG_i = Actual energy generation at ith hour

N = Number of years under simulation study.

3.6 RELEVANT DATA

The time to failure and time to repair distribution of all critical equipments are basic requirement of any reliability study. This can be obtained from recorded history of failure and repair time of such equipments. But when the equipment outage history are not maintained properly we can take help of the experts related to the fields, to generate the probability of equipment outage and repair distribution. The next chapter describes the procedures of doing this.

MAINTENANCE SCHEDULING APPROACH USING FUZZY SET THEORY

4.1 INTORDUCTION

One of the important aspect and pre-requisites of Monte Carlo simulation for realistic results is obtaining a correct probability distribution for failure and repair time of the equipments. This can be obtained if a systematic record for the time to failure and time to repair completion of equipments are maintained. Realizing this need, systematic records of equipment outage data are maintained in the western countries. Unfortunately no such efforts has been made in India. It does not seem to be realistic approach to use the outage data of western countries in the Indian context as the environment, operating and maintenance procedures, manufacturing standards etc. are quite different. Moreover the hydropower plants do not have any unique design and each installation is different from the another. Hence the use of information originating from human experts in the field of reliability and safety analysis of newly designed hydropower installations can be used.

In this procedure the distribution of time to repair and time to failure probability are obtained from the information given by human experts in the relative field. The uncertainty model plays 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 experts, and how these two kind of uncertainty interact and impact on credibility of 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 denotes 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 μ_A; that is,

$$\mu_A: X \longrightarrow [0,1].$$

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 sets 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 and 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 soon 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 states and, consequently, possess a natural capability to express 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 reality than crisp variables. It is an interesting paradox that data based on fuzzy variable provides us, in fact, with more accurate evidence about real phenomenon than 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 accuracy 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 are developed based on possibility theory.

4.3.1. 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 $x \in [s_l, s_v]$, 0 otherwise. This type of possibility distribution results when experts claim that "v lies between s_l and s_v" (Note 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 precise ones.

This 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., $s_l = x_l$, $s_u = x_u$). 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, safest intervals contain some residual uncertainty, here denoted by ε. These nested intervals will lead to membership functions of fuzzy intervals.

A fuzzy interval can be viewed as a finite set of nested (focal) 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 $\{\alpha_1=1 \geq \alpha_2 \geq \alpha_3 \geq \ldots\ldots\ldots \geq \alpha_m\}$, and letting $\alpha_{m+1} = 0$ we have

$$A_i = \{x \mid \pi_v(x) \geq \alpha_i\}; \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.2)$$

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

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 $A \cap B \neq 0$, 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 A_i.

It can be defined

$$\Pi (B) = \sum_{i=1,m} p_i \Pi_i (B) = \sup_{x \in A} \pi_v(x) \qquad (4.3)$$

$$N(B) = \sum_{i=1,m} p_i N_i (B) = \inf_{x \notin A} (1 - \pi_v(x)) \ldots\ldots\ldots\ldots \qquad [4.4)$$

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

Where $\overline{B}$ is the complement of B with respect to X. This duality expresses the fact that B tends towards certainty as $\overline{B}$ 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 ν hits A_i (e.g., from the point of view of experts, the proportion of cases where $\nu \in A_i$ 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 ν, and $A_3 = [s_l, s_u]$ corresponds to the interval which leaves a 0.05 probability (=ε) that ν misses A_3, i.e., the residual uncertainty of the conservative evaluation.

The links between λ_1'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 (s_l, s_u, m_l, m_u, c_l, c_u) (in the bold Faced Rectangle)

	Selected intervals	Level of confidence, λ_I	Degree of possibility, α_I	Weights p_i
A_1	$[c_l, c_u]$	0.05	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

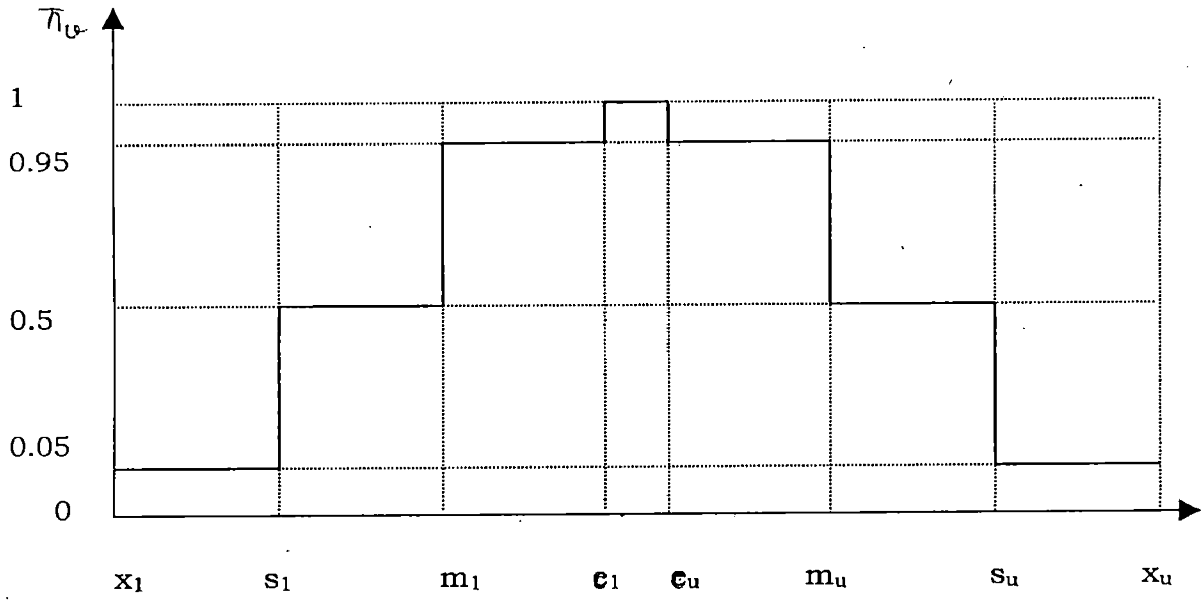


Fig. 4.1 : Expert – Originated Possibility Distribution

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.

4.3.2. 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 is 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 be 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.

4.3.2.1 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 $\mu_E = \pi_v$). 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

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

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

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

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

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

This evaluation may re-scaled so as to account for the residual uncertainty ε and so that it yields one when $s_l = s_u$ (precise response for which is $|E| = \varepsilon.|X|$) and when $s_l = x_l$, $s_u = x_u$ (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.

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

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

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

  where Π^* is the possibility measure attached to $\pi_v{}^*$ and Δ is the symmetric difference of fuzzy sets. More specifically, $\Pi(E) = \sup_x \min [\mu_F(x), \mu_E(x)]$ is the possibility of the fuzzy event E, and $\mu_{E\Delta F}(x) = |\mu_E(x)-\mu_F(x)|$. $\Pi^*[E]$ evaluates the extent to which the expert's response is consistent with the available information about v_1, and $F(|E\Delta F|)$ penalizes both under confidence and over confidence on the experts part. When the possibility (or the probability) distribution of v reduces to deterministic information ($v= x^*$) then the above indexes collapse into the first definition given in (4.9) upto the scaling factor in (4.8) that can be added if needed.

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) 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

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

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

4.3.3. 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 under study 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 enquired 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 and discard 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

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

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

$$c=1 - \text{cons} (\pi_1, \pi_2), \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4.23)$$

where cons (π_1, π_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,.......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) = sup$_x$ min [π_1(x), $\pi 2$(x)] and the following combination rule has been proposed.

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

Note that when cons (π_1, π_2)= 0, k_2 contradicts k_1 and the only opinion of k_1 is retained $\pi_{1\text{-}2} = \pi_2$ while if cons (π_1, π_2) =1 then $\pi_{1\text{-}2}$ = min (π_1, π_2) can be similarly combined with π_3, $\pi_{(1\text{-}2)\text{-}3}$ with π_4 and so on.

At this in this process the analyst will use the best method as determined by comparison of the performance of various pooling methods on seed variables.

4.3.4 Trasformation Between Possibility and Probability

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

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

where x′ is such that p(x′) = p(x) < p(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 $A_\alpha = \{x \mid \pi(x) \geq \alpha\}$. The characteristics of our data allow us to use the discrete equivalent of T_2

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

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.

4.4 EXAMPLE

Let us describe the procedure for expert-supplied information in a given uncertainity model with a simple example. There be 13 seed variables and 3 experts who give estimations in the form of intervals corresponding to confidence levels of 0.05, 0.5 and 0.95. 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 = 3

Number of test variables, m = 13

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

Real value of variables:

$x^*(v_1) = 1.5$

$x^*(v_2) = x^*(v_3) = x^*(v_4) = 2.5$

$x^*(v_5) = x^*(v_6) = x^*(v_7) = x^*(v_8) = 3.5$

$x^*(v_9) = x^*(v_{10}) = 4.5$

$x^*(v_{11}) = 5.5$

$x^*(v_{12}) = 6.5$

$x^*(v_{13}) = 7.5$

The intervals supplied by experts e_1, e_2 and e_3 is shown below

Focal sets	Confidence level λ_I	Expert e_1	Expert e_2	Expert e_3
A_1	0.05	4-5	3-5	5-6
A_2	0.50	3-7	2-6	4-7
A_3	0.95	2-9	1-8	3-9

Now from the two relations $\lambda_i = 1-\alpha_{i+1}$ and $p_i = \alpha_i - \alpha_{i+1}$ the following tables are constructed.

Table 4.2 Mean of Expert Assessment (Global Measures)

Experts	Accuracy $A(e_i)$	Precision $S_p(e_i)$	Quality $Q(e_i)$
e_1	0.75	0.45	0.338
e_2	0.792	0.445	0.352
e_3	0.504	0.54	0.272

Possibility distribution is constructed for each experts opinion from this table.

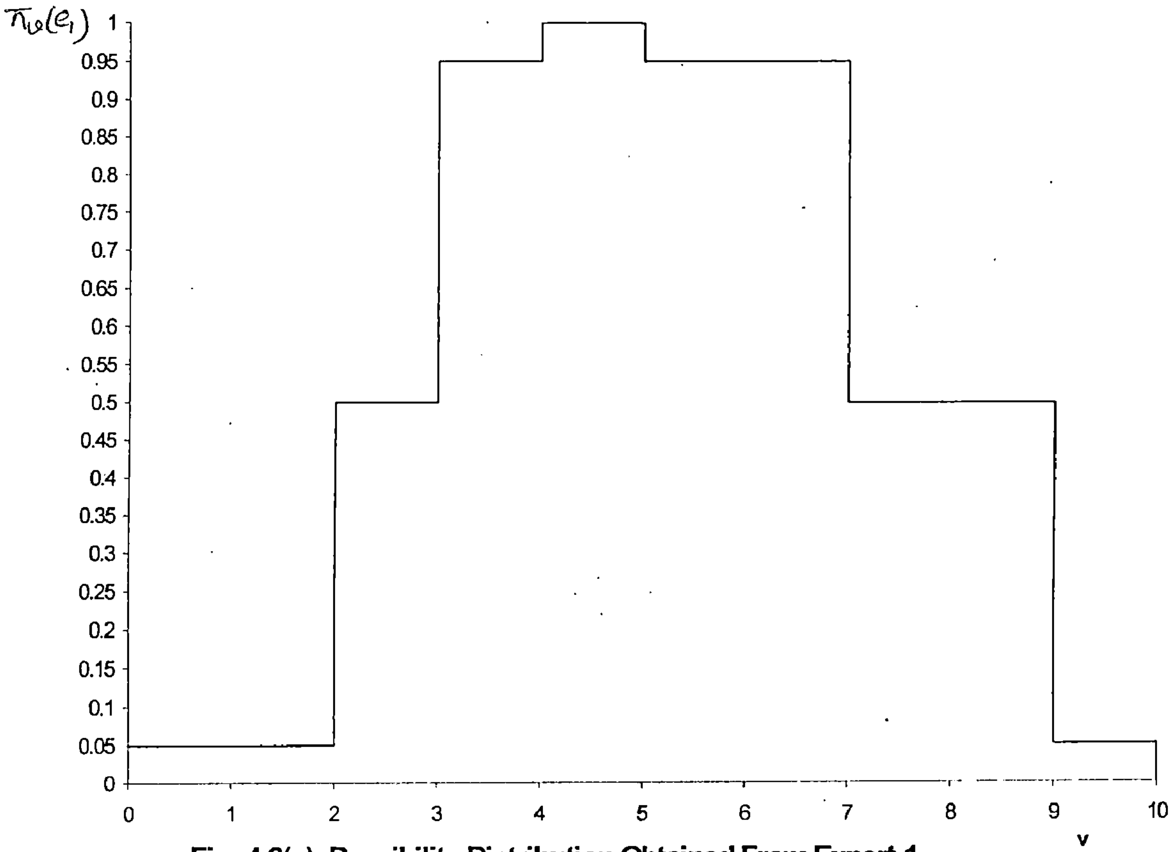


Fig: 4.2(a) Possibility Distribution Obtained From Expert-1

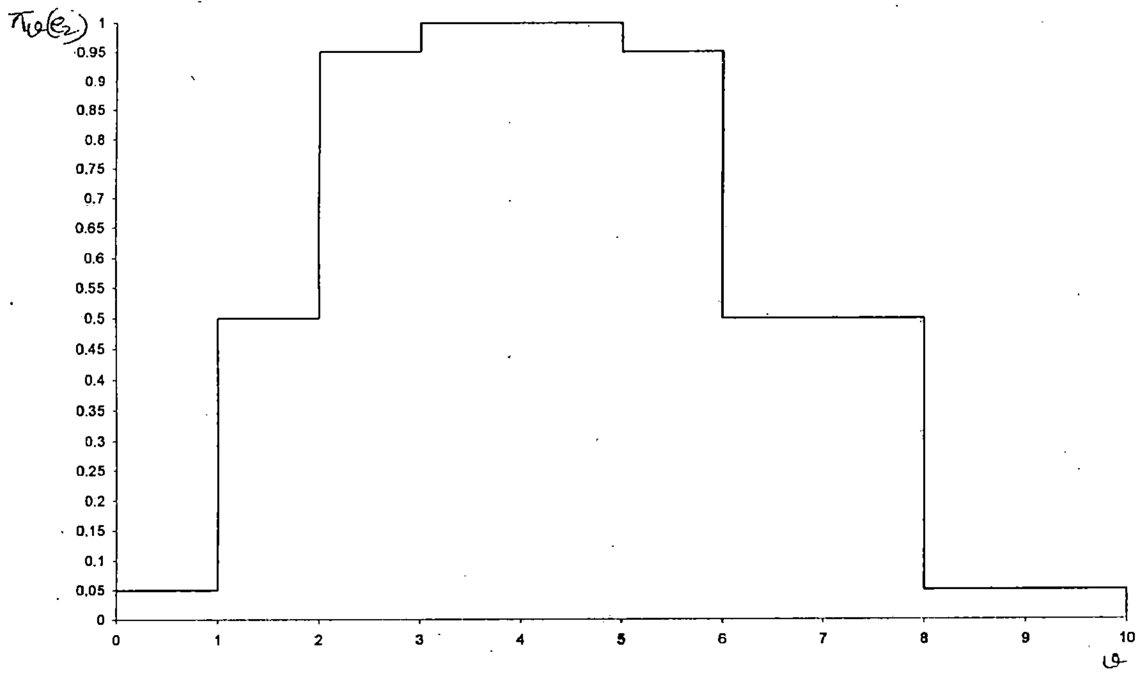


Fig: 4.2(b) Possibility Distribution Obtained From Expert-2

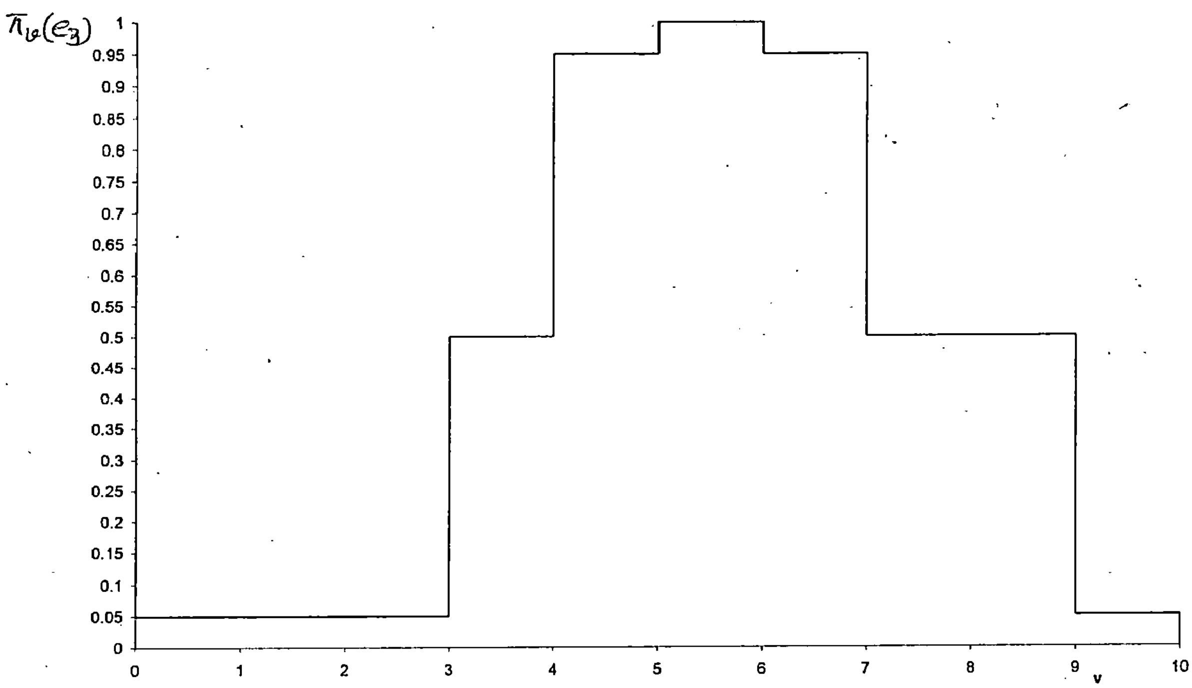


Fig: 4.2(c) Possibility Distribution Obtained From Expert-3

Fig. 4.2(a), 4.2(b) and 4.2(c) shows the possibility distribution given by experts e_1,e_2 and e_3 respectively. Let us verify the assessment of expert e_1 in relation to variable v_6: his accuracy is $A(e_1,v_6) = \pi_{e1,v6}(3.5) = 0.95$, his precision is $S_p(e_1) = 1-(1 \times 0.05+4 \times 0.45 + 7 \times 0.45 + 10 \times 0.05)/10 = 0.45$ and his quality is $Q(e_1,v_6) = 0.95 \times 0.45 = 0.4275$.

In table 4.3 $A(e_i)$, $S_p(e_i)$ and $Q(e_i)$ of all three experts are shown. We can see that considering the whole set of seed variables, expert e_2 is most precise as well as accurate.

Different pooling methods are used with the same example.

Conjunctive Pooling

Among the three experts e_1 and e_2 seems reliable and $\pi_1$and π_2 overlaps significantly . Hence the conjuctive pooling yields the following possibility distribution.

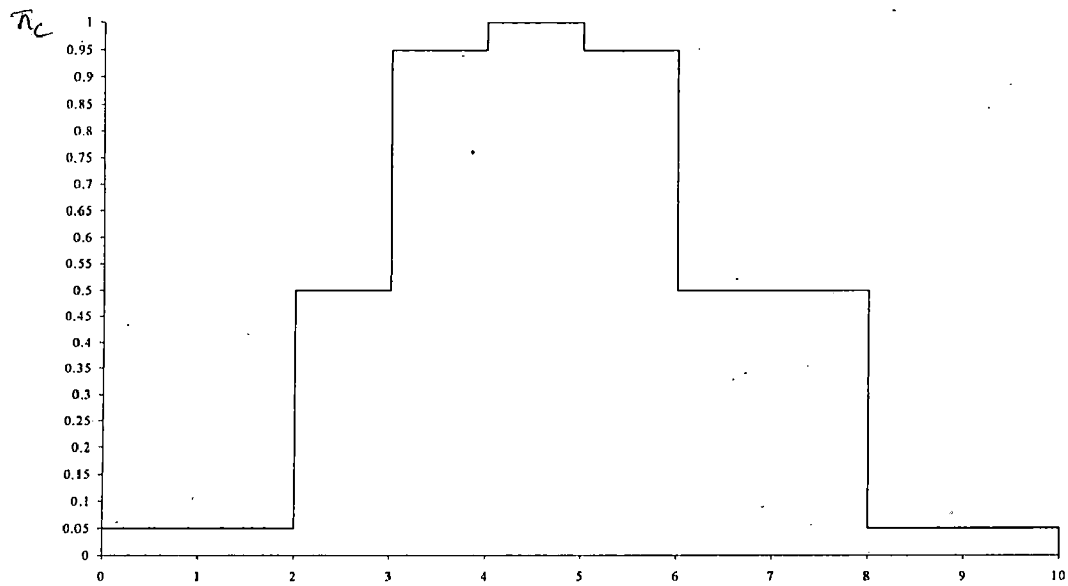


Fig 4.3 AGGREGATION BY CONJUNCTIVE METHOD

Disjunctive Pooling :

The disjunctive pooling method yields the following possibility distribution.

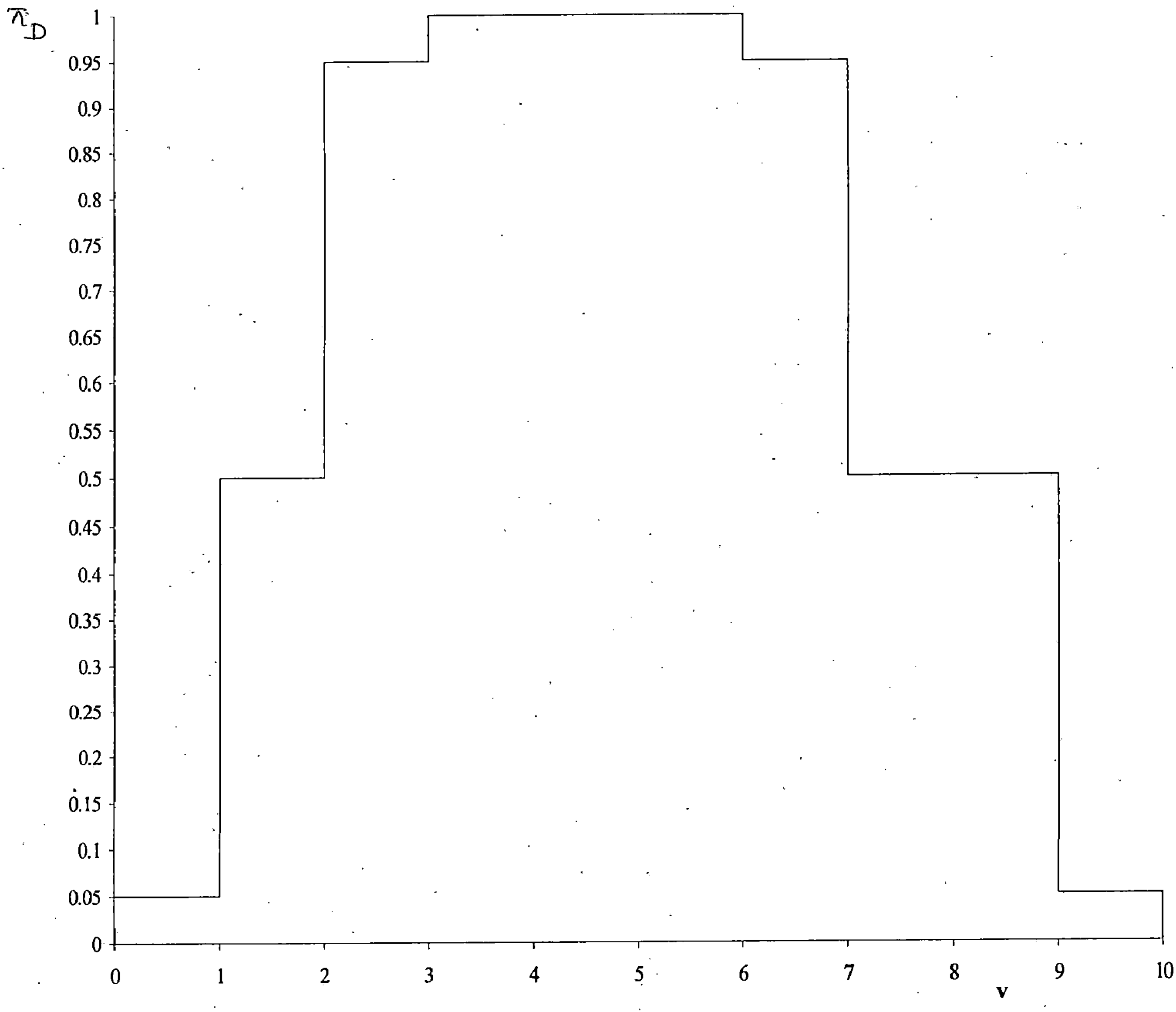


Fig 4.4 AGGREGATION BY DISJUNCTIVE MODE

Average Pooling Mode:

Average mode of pooling yields the following possibility distribution.

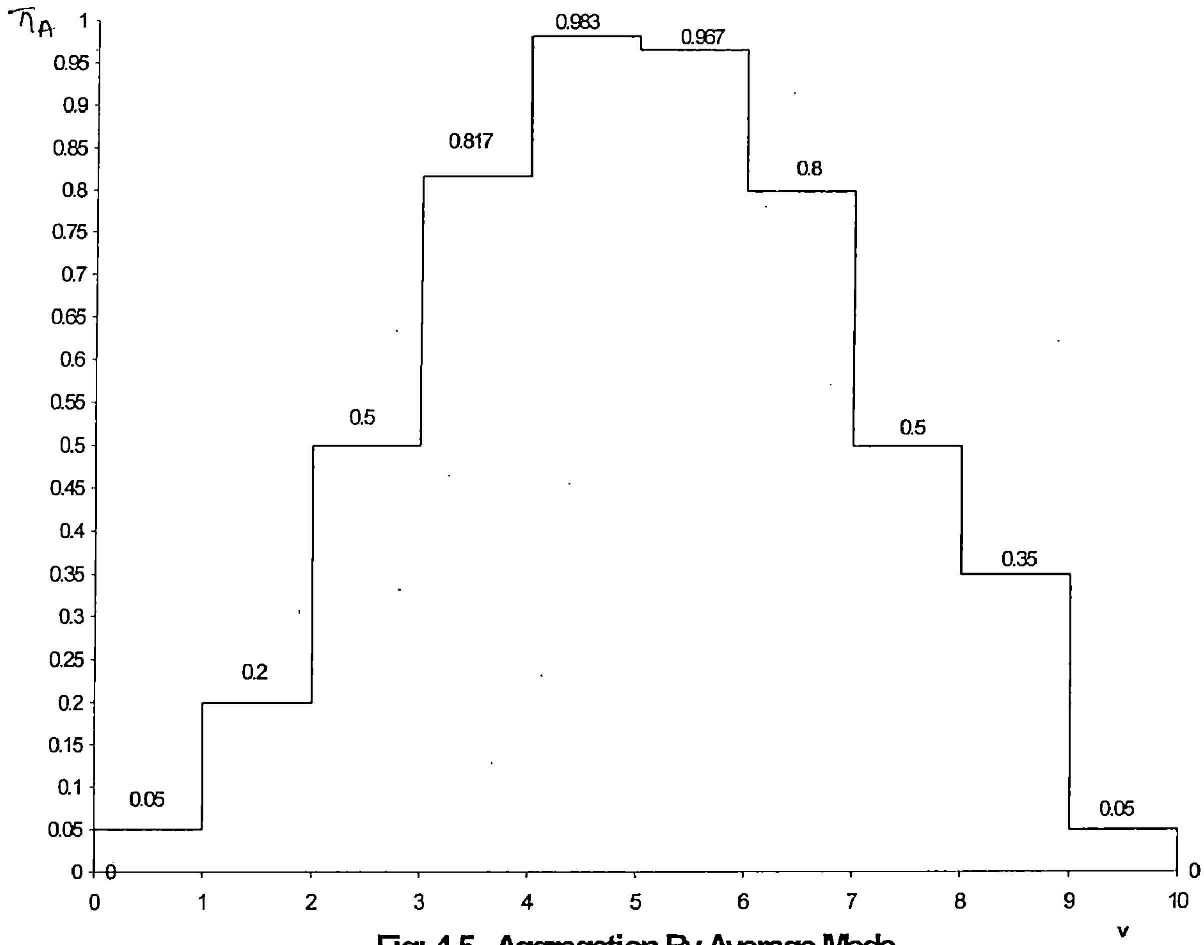


Fig: 4.5 Aggregation By Average Mode

Consistency Based Trade-off:

Consistency based trade-off between experts e_1 and e_2 yields same possibility distribution as obtained in conjunctive mode.

Priority Aggregation Mode:

In priority aggregation mode of pooling experts e_1 and e_2 constitutes of same group as they corresponds to similar and higher reliability. Therfore, the information provided by experts e_1 and e_2 are aggregated using conjunctive mode which yields a distribution as shown in fig (4.3). This result is then combined with the information obtained from the expert e_3 in the combination process as given in equation (4.24).

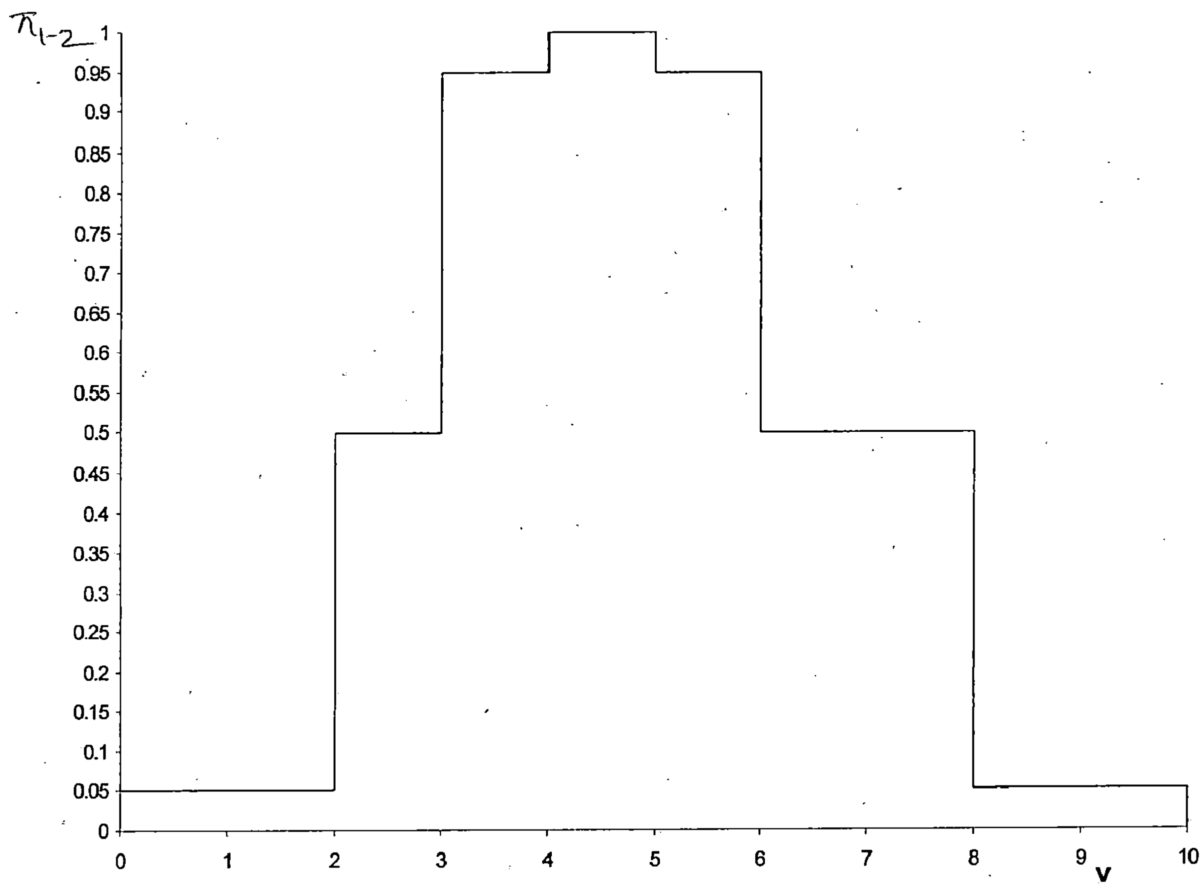


Fig : 4.6 Priority Aggregation Mode

Table 4.3 Assessment of Different Aggregating Method

Aggregation method	$A(\pi_p)$	$S_p(\pi_p)$	$Q(\pi_p)$
π_C	0.715	0.54	0.386
π_D	0.908	0.35	0.318
π_A	0.715	0.469	0.335
π_T	0.715	0.54	0.386
$\pi_{1\text{-}2\text{-}3}$	0.715	0.54	0.386

The global measures of assessment is shown in table 4.3. It is clear from the table that conjunctive method, consistency based trade-off method or priority aggregation methods give same and precession and accuracy for the chosen example. Hence any of these three methods can be used according to discretion of the system analyst.

It is to be noted that pooling methods mentioned here are still ad-hoc as the possibility theory is still in a developing stage. Lot of works is still to be carried out before describing a clear and distinctive means of pooling of experts opinion.

SIMULATION AND RESULTS

5.1 LOGICS AND ASSUMPTIONS FOR SIMULATION

In the simulation model considered here, we have assumed that equipment u can be replaced repaired 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 u) along with any other equipments of the same unit

or

(iii) after $a_{2i}*m_i$ time unit (where a_{2i} is a constant) of continuous operation and the equipment has a standby equipment ready for switchover.

In this policy, one equipment (say, bearing of unit No. 1) is repaired on failure. During repair of this equipment (bearing of Unit No.1), preventive maintenance is done on any other equipments of unit –1 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 equipment. Further for subsystems with standby redundant equipment (governor oil pump and lubrication oil pumps), preventive maintenance is done if the equipment 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 equipments of a subsystem having standby redundant equipments. However, the subsystems which do not have redundant units are preventively maintained (at and age $\geq a_1{*}m$) 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 hydro power plant with various set of a_1 and a_2 is given below. The program developed in C++ language based on this flowchart given in the annexure.

5.2 SIMULATION PROCEDURE

There are two procedures of Monte Carlo simulation (i) synchronised with time and (ii) event wise.

In the first procedure the simulation progresses with minimum unit of time and watching if any events occur at that moment. In this procedure the logics are simpler but the program is synchronised with the clock of digital computer. The computer time needed in this procedures is much more than the second procedure.

In second procedure the simulation proceeds eventwise. The time jumps from one event to next event chronologically and keeps on adding, the time. The program becomes a bit complicated to find the next earliest event but it saves much computer time. This is very beneficial when we want to simulate for a very long time. This second procedure has been followed in this study.

Before beginning the simulation, variables for the system and expected unserved energy (EUE) is set to zero. Then states of equipments are selected randomly from their probability distribution. This procedure requires a maintenance model and component failure model which is discussed in Chapter 3. The result of the simulation are random variables of interest (expected generation, equipment status and unserved energy etc.). These results are utilized in the computation of appropriate relaibility indexes.

A flow chart for Monte Carlo Simulation is given in Fig. 5.1.

5.3. RESULTS OF SIMULATION

The hydro power plant operation is realized with the help of the developed program. The model is simulated for a maximum simulation period of 3 years. 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 n_1 expected energy not supplied is minimum for a_1 = 1.1 and a_2 = 1.2 and this unserved energy is equal to 1415 MWH

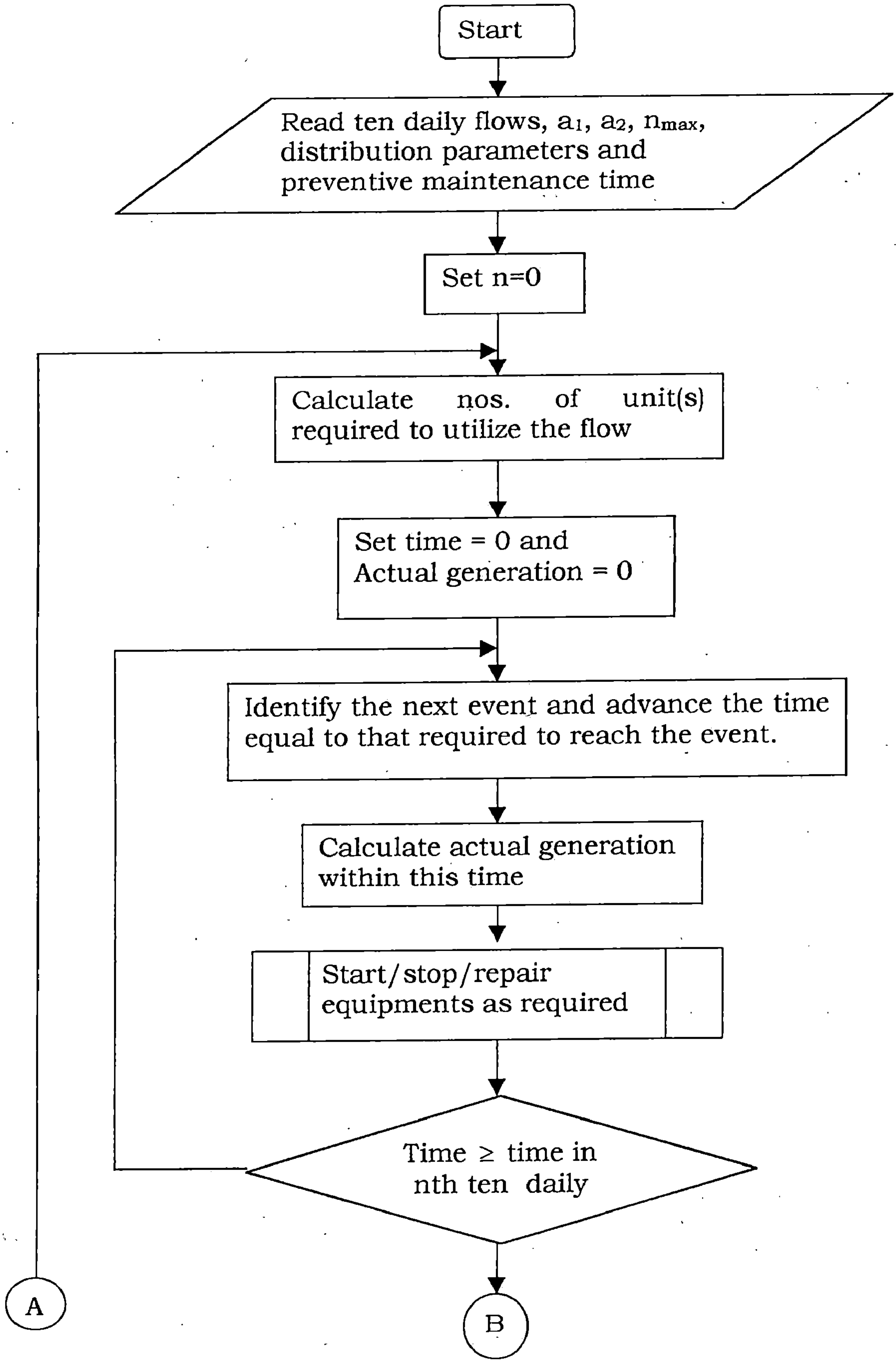
Fig. 5.2 gives an three dimensional view of the expected energy supplied for different points in the solution space. The plant managers can take decisions based on this results.

TABLE 5.2 Expected Annual Enegy Not Supplied For Different Values Of a1 And a2

a2 \ a1	0.7	0.75	0.8	0.85	0.9	0.95	1	1.05	1.1
0.9	4508	4230	1959	2968	2747	2001	2368	2368	2368
0.95	3078	3581	2084	2736	2689	1952	2342	2342	2342
1	1729	1729	2952	2948	2324	1911	2935	2935	2935
1.05	3092	3181	3425	2588	2110	2327	2487	2010	2010
1.1	1458	1983	2800	2939	2170	2329	2140	2767	2767
1.15	2039	2192	1938	2213	2255	2149	2520	2042	1784
1.2	1975	2762	2738	1975	1975	2145	1788	1810	1415
1.25	2828	2263	3193	2983	2983	4121	4457	4161	2133
1.3	2750	2762	5265	4224	4224	3711	4314	4441	4215
1.35	6173	1922	4049	2382	2382	5931	3695	2437	3945
1.4	5304	2572	3114	2912	2912	4621	4967	4557	3806
1.45	3237	2818	4319	2578	2578	3060	3175	2728	2728
1.5	2650	2711	4319	2578	2578	2606	2579	2838	2838

Note:- The distribution parameters (of time to failure and time to repair) are not any real values, but assumed values.

FLOWCHART FOR COMPUTER SIMULATION



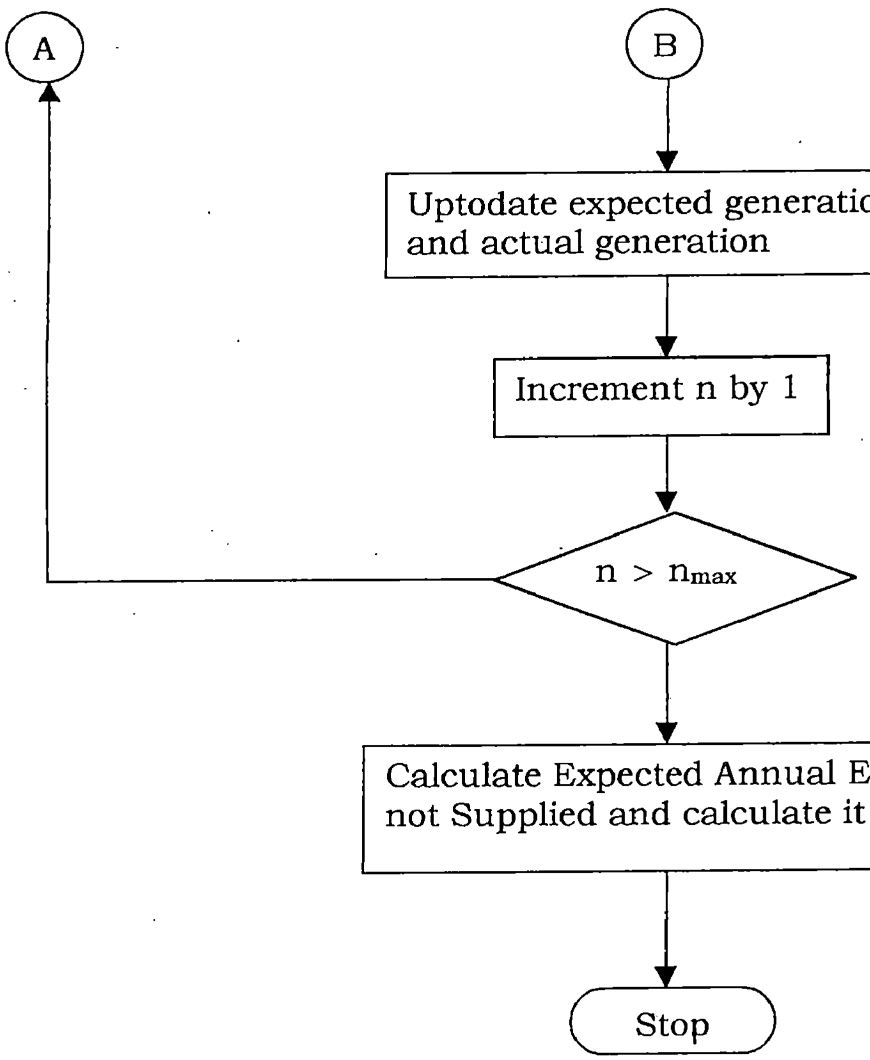


Fig. 5.1 Flow Chart for Computer Simulation

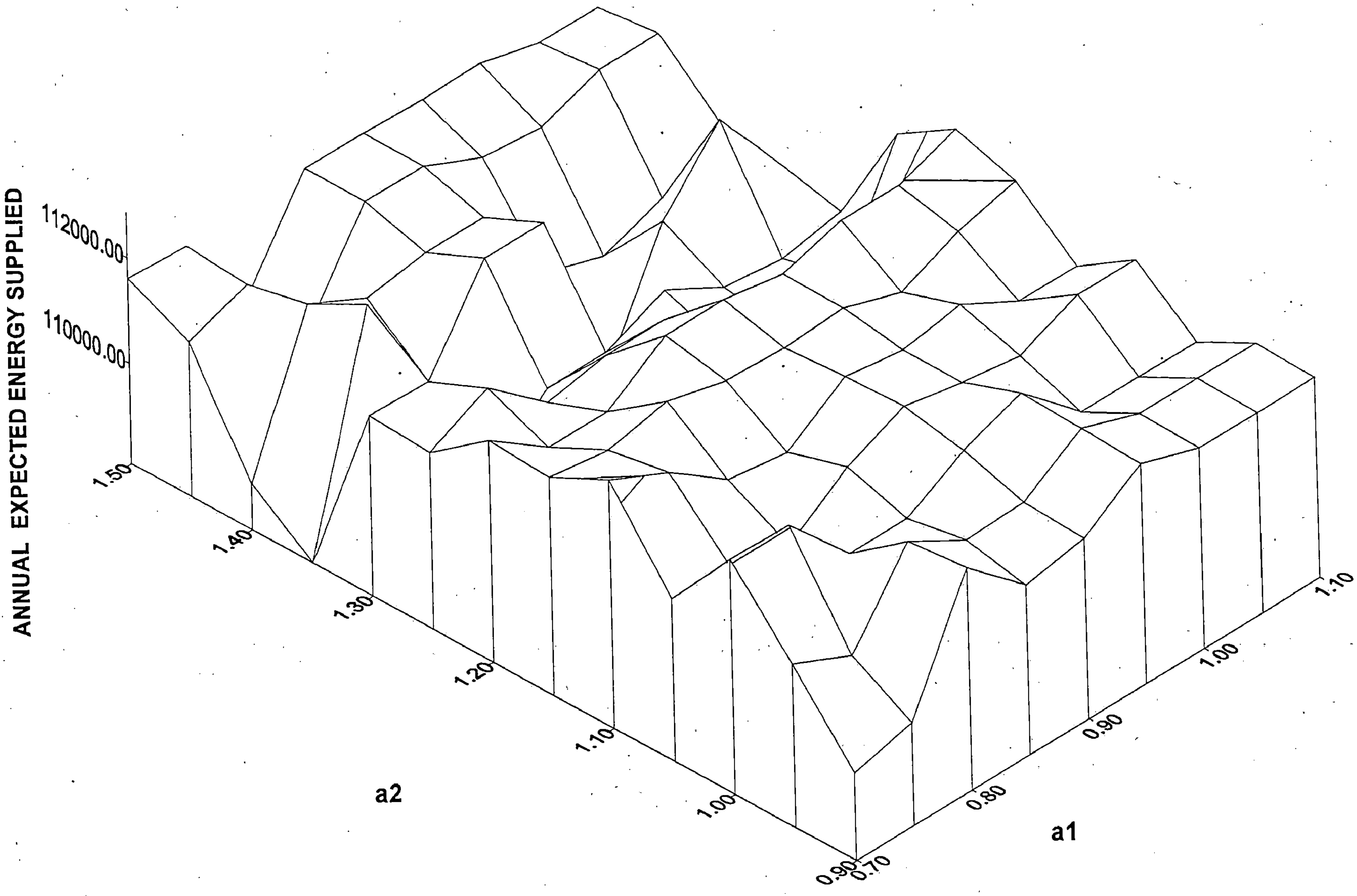


Fig. 5.2 : Expected Energy Supplied for Different Values of a1 and a2

CONCLUSION

Preventive maintenance is adopted in hydro power plants to improve the generation and availability. This is necessary as any loss of load due to break down of any or all unit of plant do not only result in 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 impact of different present maintenance scheduling on plant availability. This method is selected because of its relative advantage over analytical methods, which become very complex to use for hydro generating plants with complex operating situations, multiple variables and numbers of constraints.

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, a procedure for processing human originated information has been devised. The method consist of three steps, getting useful information from the experts, assessment of the quality of the expert and lastly combining the responses of several experts to yield a unique, hopefully better response. All the three steps is proposed to be done in the framework of possibility theory, which offers a simple theory of uncertainty that explicitly takes into account the lack of precision which is more realistic. Assessing and pooling step of expert opinion by possibility approach presents less difficulties relative to the probabilistic (classical) approach.

A real world experiment as described in [4] 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 number of simulations for different preventive maintenance intervals are observed. The preventive maintenance schedule corresponding to the highest availability may be selected.

The most important features of this study is to evolve a simple yet realistic procedure of deriving the impact of different preventive maintenance scheduling on reliability (availability) of a hydro power plant. This will help the power plant managers to take the best decisions. This study can be applied (extended) for hydro power plants with storage capacity and thermal power plants, with addition or alternation of some logical steps in the plant model. It leaves a future scope of study of deterioration process through discrete stages and incorporating the concept of "maintenance when needed".

One important point to be noted, that it is beneficial to follow this procedure of processing with human originated data only when adequate data is not available. Recorded outage data and repair data of equipments if recorded systematically are always more reliastic than those originated from human experts. Hence due importance should be given for recording the data, which will verify the initial assumption and help to take in corrective measures. A fuzzy controller can be build up based on the informations recorded to take any corrective action, but this requires further study.

The methodology described in this thesis can be applied not only for selecting preventive maintenance policy but has a wide scope such as system planning, unit commitment etc.

REFERENCES

1. 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.
2. Yasser El Sobaty and M. A.Ismail, "Fuzzy Clustering or Symbolic Data", IEEE Trans on Fuzzy System, Vol-6 No.2, May 1998.
3. J.Endreny I, G.J. Anders, A.M.Lecte Da Silva, "Probabilistic Evaluation of the Effect of Maintenance on Relibility-an Application", IEEE Trans and Power Systems, Vol,-13, May 1998, pp 576-583.
4. 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.
5. "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.

6. 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 Trains on Power Apparatus and System, Vol-4, No.3, pp 829-835, Aug. 1989.

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. 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.

9. Faray. A. El-Sheikhi, Roy Billinton, "Generating Unit Maintenance Scheduling For Single and Two Interconnected Systems", IEEE Trans. on Power Apparatus and Systems, Vol-PAS-103, No.5, May 1984.

10. N.E.Chang, "Application of Availability Analysis Techniques to Improve Power Plant Productivity", IEEE Trans. PAS, Vol-102, pp 1521-1529, June. 1993.

11. 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.

12. "Reliability Modeling in Electrical Power System", (Book), J.Endrenyi, Wiley Interscience Publication, 1979.

13. Report of AIEE Subcommittee on Application of Probability Methods, "Application of Probability Methods to Generating Capacity Problem", AIEE Trans. On PAS, Vol-79, pp 1165-1182, Feb.1961.

14. 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.

ANNEXURE

```
//mcs_hydro.cpp
//Program For Monte Carlo Simulation Of Hydro Power Plant.

 # include <fstream.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 status {run, stby, dwn};
  //enum pumpcond {yes,no};

  struct equipment
   {
    int gmf,itf,gmr,itr,prmt,tnf,tnr,rnt,dnt,mf,crnt,cdnt,no_of_mnt,
      no_of_pmnt,no_of_omnt;
    float btf,btr;
    status stat;
   };
  struct unit
   {
    int rnt,dnt,crnt,cdnt;
    status stat;
   };
  void main ( )
   {
    clrscr ();
    ofstream outfile("plant.dat");
    //pumpcond pmpcon;
    int rmt,nurq,nurn,nudn,nsu,nprn,npdn,npsb,mntcp[3],mntcs[3][2][2],
      rep_maint ;
    //rmt=remaining time; nurq=nos. of unit reqired; nurn=nos. of unit running.
    //nus= nos of standby unit; pg[td]=possible generation for each ten daily.
    //nprn= nos of pump running; npsb= nos. of standby pumps;
    float total_gen, total_exgen, gen[36],pg[36] ;
    int unitrun (int np, int nu);
    int pumpreq (int nunt);  //finds nos. of pump(s) required.
    //pumpcond chkpmp(int nur, int npr);//checks if cwp requirement is
fullfilled.
    float generation (int nurqd, float flow, int time);//calculates possible
generation.
    void update_time (equipment& eqp, int tm);//updates eqpt. run & down time.
    //void update_time (unit& unt, int tm);
    int time_to_failrep(int gm, int it, float bt);//calculates next flr/rpr
time.
    int mean_time_to_faliure(int gm, int it, float bt);//calculates mean-time-
                                                  //to faliure of each eqpt.
    const int max = 9000;
    equipment cwp[4], eqp[3][4], eqps[3][2][2];
    unit unt[3];
    float a1, a2;
    cout<< "\n Enter the value of a1 : "; cin>>a1;
    cout<< "\n Enter the value of a2 : "; cin>>a2;
    float flow[36] ={267.0, 92.0,143.0,110.0,112.0,108.0,102.0,120.0,148.0,
                198.0,182.0,137.0,225.0,132.0,229.0,197.0,283.0,324.0,
                277.0,415.0,338.0,294.0,312.0,287.0,252.0,228.0,212.0,
```

```
              187.0,191.0,178.0,167.0,173.0,151.0,103.0,127.0, 95.0};
int days[36]={10,10,11, 10,10, 8, 10,10,11, 10,10,10, 10,10,11, 10,10,10,
          10,10,11, 10,10,11, 10,10,10, 10,10,11, 10,10,10, 10,10,11};
int st = 0;             // initialize simulation time.
//int rt = 0;            // initialize run time.
int etf, etm, etr;
int gfp[3] = {96,82,72};
int it_fp[3]={750,800,780};
float bfp[3]={1.21,1.50,1.60};
int grp[3]={30,25,26};
int it_rp[3]={156,170,145};
float brp[3]={1.5,1.08,1.64};
int pmp[3]={24,24,24};
int gfu[3][4]={{200,228,65,16},{185,2150,50,10},{350,185,75,12}};
int it_fu[3][4]={{1500,1485,835,750},{1450,1340,750,700},
                              {1420,1285,745,680}};
float bfu[3][4]={{1.21,1.7,1.4,1.18},{1.5,1.63,1.4,1.12},
                          {1.37,1.45,1.35,1.4}};
int gru[3][4]={{180,132,18,16},{114,140,10,14},{120, 135,15,15}};
int it_ru[3][4]={{410,320,36,62},{450,288,25,65},{440,315,48,58}};
float bru[3][4]={{1.56,1.4,11.3,1.4},{1.8,1.5,1.21,1.27},
                            {1.18,1.7,1.6,1.13}};
int pmu[3][4]={{144,120,24,10},{144,120,24,10},{144,120,24,10}};
int gfs[3][2][2]={{{114,87},{45,50}},{{75,90},{60,55}},
                              {{123,108},{63,72}}};
int it_fs[3][2][2]={{{680,505},{674,489}},{{700,491},{650,515}},
                                  {{715,475},{682,454}}};
float bfs[3][2][2]={{{1.16,1.4},{1.2,1.5}},{{1.52,1.32},{1.7,1.3}},
                                  {{1.43,1.27},{1.6,1.25}}};
int grs[3][2][2]={{{12,20},{8,12}},{{18,15},{10,8}},{{10,16},{13,8}}};
int it_rs[3][2][2]={{{65,85},{57,67}},{{72,65},{50,58}},{{81,68},{71,60}}};
float brs[3][2][2]={{{1.7,1.28},{1.7,1.2}},{{1.1,1.4},{1.3,1.35}},
                                  {{1.4,1.8},{1.3,1.8}}};
int pms[3][2][2]={{{24,24},{16,16}},{{24,24},{16,16}},{{24,24},{16,16}}};
for ( int xi = 0; xi<3; xi++)         // initialize the known parameters -
 {                                   // - of all the equipments.
   cwp[xi].gmf = gfp[xi];
   cwp[xi].itf = it_fp[xi];
   cwp[xi].btf = bfp[xi];
   cwp[xi].gmr = grp[xi];
   cwp[xi].itr = it_rp[xi];
   cwp[xi].btr = brp[xi];
   cwp[xi].prmt= pmp[xi];
   cwp[xi].mf = mean_time_to_faliure(cwp[xi].gmf,cwp[xi].itf,
                                        cwp[xi].btf);
   cwp[xi].rnt = 0;
   cwp[xi].dnt = 0;
   cwp[xi].stat = stby;
   cwp[xi].tnf = max;
   cwp[xi].tnr = max;
   cwp[xi].crnt= 0;
   cwp[xi].cdnt= 0;
   cwp[xi].no_of_pmnt=0;
   mntcp[xi]=max;
   unt[xi].stat=stby;
   for (int xj=0; xj<4; xj++)
    {
    eqp[xi][xj].gmf = gfu[xi][xj];
    eqp[xi][xj].itf = it_fu[xi][xj];
    eqp[xi][xj].btf = bfu[xi][xj];
    eqp[xi][xj].gmr = gru[xi][xj];
```

```
        eqp[xi][xj].itr = it_ru[xi][xj];
        eqp[xi][xj].btr = bru[xi][xj];
        eqp[xi][xj].prmt= pmu[xi][xj];
        eqp[xi][xj].mf = mean_time_to_faliure(eqp[xi][xj].gmf,
                              eqp[xi][xj].itf,eqp[xi][xj].btf);
        eqp[xi][xj].rnt = 0;
        eqp[xi][xj].dnt = 0;
        eqp[xi][xj].stat = stby;
        eqp[xi][xj].tnf = max;
        eqp[xi][xj].tnr = max;
        eqp[xi][xj].crnt= 0;
        eqp[xi][xj].cdnt= 0;
        eqp[xi][xj].no_of_pmnt=0;
        if (xj<2)
         {
         for (int xk=0; xk<2; xk++)
         {
         eqps[xi][xj][xk].gmf = gfs[xi][xj][xk];
         eqps[xi][xj][xk].itf = it_fs[xi][xj][xk];
         eqps[xi][xj][xk].btf = bfs[xi][xj][xk];
         eqps[xi][xj][xk].gmr = grs[xi][xj][xk];
         eqps[xi][xj][xk].itr = it_rs[xi][xj][xk];
         eqps[xi][xj][xk].btr = brs[xi][xj][xk];
         eqps[xi][xj][xk].prmt= pms[xi][xj][xk];
         eqps[xi][xj][xk].mf = mean_time_to_faliure(eqps[xi][xj][xk].gmf,
                              eqps[xi][xj][xk].itf,eqps[xi][xj][xk].btf);
         eqps[xi][xj][xk].rnt = 0;
         eqps[xi][xj][xk].dnt = 0;
         eqps[xi][xj][xk].stat = stby;
         eqps[xi][xj][xk].tnf = max;
         eqps[xi][xj][xk].tnr = max;
         eqps[xi][xj][xk].crnt= 0;
         eqps[xi][xj][xk].cdnt= 0;
         eqps[xi][xj][xk].no_of_pmnt=0;
        mntcs[xi][xj][xk]=max;
         }
        }//end of if (xj<2)
       }//end of for (int xj=0; xj<4; xj++)
     } //end of for ( int xi = 0; xi<3; xi++)
    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;

    //nurq=0;
    nurn=0;
    nudn=0;
    nsu=3;
    nprn=0;
    npdn=0;
    npsb=3;
    total_gen=0;
    total_exgen=0;
    rep_maint=0;
    int count=0;
   // loops for years
for (int yr=0; yr<3; yr++)
 {
   // loops for 10 dailies.
 for (int td=0; td<36; td++)
  {//cout<<endl<<"td="<<td;
```

```
outfile<<"\n td="<<td;
gen[td] = 0;
st = 0;
if ((flow[td]>0)&&(flow[td]<=109.2)) nurq=1;
 else if ((109.2<flow[td])&&(flow[td]<=218.4)) nurq=2;
  else if (218.4<flow[td]) nurq=3;
outfile<<" flow["<<td<<"]="<<flow[td];
int tendlyhrs=days[td]*24;
rmt = tendlyhrs;
pg[td]=generation (nurq, flow[td], tendlyhrs);//calculate possible generation
outfile<<" NURQ="<<nurq<<" nurn="<<nurn;
int temp;
if ((nurn+nsu)>=nurq) temp=nurq;
else if ((nurn+nsu)<nurq) temp=(nurn+nsu);
outfile<<" TEMP="<<temp;
int temp0=unitrun((nprn+npsb), temp);outfile<<" unitrun="<<temp0;
int temp1=pumpreq(temp0); outfile<<" pumpreq="<<temp1;
// start pumps &units if required.
if ((nurn<temp0)&&(nsu>0))
 {
 for (int a=0; a<3; a++)
  {
  //starting condition for pumps
  if ((nprn<temp1)&&((nprn+npdn)<3))
   {
   if(cwp[a].stat==1)
    {
    cwp[a].stat=run;
    nprn++;
    npsb--;
    if(cwp[a].rnt==0)
     {
     cwp[a].tnf=time_to_failrep(cwp[a].gmf,cwp[a].itf,cwp[a].btf);
     cwp[a].tnr=max;
     }
    }
   }

  } //end of for a=0

//starting the units.
for (int ai=0;ai<3;ai++)
 {
  if ((unt[ai].stat==1)&&(temp0>nurn))     //   (pmpcon==0)&&
   {
   unt[ai].stat=run;
   nurn++;
   nsu--;
   for (int aj=0;aj<4;aj++)
    {
    if (eqp[ai][aj].stat==1)
   {
   eqp[ai][aj].stat=run; //cout<<"
eqp["<<ai<<"]["<<aj<<"]="<<eqp[ai][aj].stat;
   if(eqp[ai][aj].rnt==0)
    {
    eqp[ai][aj].tnf=time_to_failrep(eqp[ai][aj].gmf,eqp[ai][aj].itf,
                                           eqp[ai][aj].btf);
    eqp[ai][aj].tnr=max;
    }
   }
```

```
     if (aj<2)
     {
     if ((eqps[ai][aj][0].stat==1)&&
           (eqps[ai][aj][0].rnt>=eqps[ai][aj][1].rnt))
      {
       eqps[ai][aj][0].stat=run;

       if (eqps[ai][aj][0].rnt==0)
        {
        eqps[ai][aj][0].tnf=time_to_failrep(eqps[ai][aj][0].gmf,
                             eqps[ai][aj][0].itf,eqps[ai][aj][0].btf);
        eqps[ai][aj][0].tnr=max;
        }
       }
      else if(eqps[ai][aj][0].stat==2)
       {
       eqps[ai][aj][1].stat=run;
       if (eqps[ai][aj][1].rnt==0)
        {
        eqps[ai][aj][1].tnf=time_to_failrep(eqps[ai][aj][1].gmf,
                             eqps[ai][aj][1].itf,eqps[ai][aj][1].btf);
        eqps[ai][aj][1].tnr=max;
        }
       }
       //cout<<"eqps["<<ai<<"]["<<aj<<"][0]="<<eqps[ai][aj][0].stat;
       //cout<<"eqps["<<ai<<"]["<<aj<<"][1]="<<eqps[ai][aj][1].stat;
      }//end of if aj<2
      }//end of for aj=0
     }//end of if unit[ai].stat==1
    }//end of for ai=0
   }//end of if ((nurn<temp0)&&(nsu>0))

 // stop pumps & units if required.
  for (int bi=2; bi>=0; bi--)
   {
   /*int temp;
   if ((nurn+nsu)>=nurq) temp=nurq;
    else if ((nurn+nsu)<nurq) temp=(nurn+nsu);
   int temp0=unitrun((nprn+npsb), temp);
   int temp1=pumpreq(temp0);*/
   //conition for pump stop.
   if ((nprn>temp1)&&(cwp[bi].stat==0))
    {
    cwp[bi].stat=stby;
    nprn--;
    npsb++;
    }
   //condition for unit stop.
   if ((nurn>temp0)&&(unt[bi].stat==0))
    {
    unt[bi].stat=stby;
    nurn--;
    nsu++;
    for (int bj=0; bj<4; bj++)
     {
     if (eqp[bi][bj].stat==0) eqp[bi][bj].stat=stby;
     if (bj<2)
      {
      for (int bk=0; bk<2; bk++)
     {
     if (eqps[bi][bj][bk].stat==0) eqps[bi][bj][bk].stat=stby;
```

```
      }
       }//end of if (bj<2)
      }//end of for (int bj=0; bj<2; bj++)
     }//end of if ((nurn>temp0)&&(unt[bi].stat==0))
    }//end of for (int bi=2; bi>=0; bi--)


do
 {
 // repair loop
 if ((etr<=rmt)&&(etr<=etf)&&(etr<=etm))
  {outfile<<" etr loop ";
  rmt-=etr;
  st+=etr;
  gen[td]+=generation (nurn, flow[td], etr);
  //changing the status of the eqpts. from dwn to stby if repair is complete.
  for (int ci=0; ci<3; ci++)
   {
   update_time (cwp[ci], etr);
   if ((cwp[ci].dnt==cwp[ci].tnr)&&(cwp[ci].stat==2))
    {
    cwp[ci].cdnt+=cwp[ci].dnt;
    cwp[ci].dnt=0;
    cwp[ci].stat=stby;
    npdn--;
    npsb++;
    }
   for (int cj=0; cj<4; cj++)
    {
    update_time (eqp[ci][cj], etr);
    if ((eqp[ci][cj].dnt==eqp[ci][cj].tnr)&&(eqp[ci][cj].stat==2))
     {
     eqp[ci][cj].cdnt+=eqp[ci][cj].dnt;
     eqp[ci][cj].dnt=0;
     eqp[ci][cj].stat=stby;
     }
    if (cj<2)
     {
     for (int ck=0; ck<2; ck++)
      {
      update_time (eqps[ci][cj][ck], etr);
      if ((eqps[ci][cj][ck].dnt==eqps[ci][cj][ck].tnr)&&
                          (eqps[ci][cj][ck].stat==2))
     {
     eqps[ci][cj][ck].cdnt+=eqps[ci][cj][ck].dnt;
     eqps[ci][cj][ck].dnt=0;
     eqps[ci][cj][ck].stat=stby;
     }
      }
     }//end of if (cj<2)
    }//end of for (int cj=0; cj<4; cj++)
   }//end of for (int ci=0; ci<3; ci++)

 //checking unit status.
 for (int di=0; di<3; di++)
  {
  int temp4 = 0;
  for (int dj=0; dj<4; dj++)
   {
   if (eqp[di][dj].stat==1) temp4++;
```

```
    if (dj<2)
     {
     if((eqps[di][dj][0].stat==1)||(eqps[di][dj][1].stat==1)) temp4++;
     }
    } //end of for int dj=0;
   if ((temp4==6)&&(unt[di].stat==2))
    {
    unt[di].stat=stby;
    nsu++;
    nudn--;
    }
   }// end of for int di=0;
 int temp;
 if ((nurn+nsu)>=nurq) temp=nurq;
  else if ((nurn+nsu)<nurq) temp=(nurn+nsu);
 int temp0=unitrun((nprn+npsb), temp);
 int temp1=pumpreq(temp0);
 // start pumps & units if required.
 //if ((nurn<temp0)&&(nsu>0))
 // {
  for (int e=0; e<3; e++)
   {
   //starting condition for pumps
   if (nprn<temp1)
    {
    if(cwp[e].stat==1)
     {
     cwp[e].stat=run;
     npsb--;
     nprn++;
     if(cwp[e].rnt==0)
      {
      cwp[e].tnf=time_to_failrep(cwp[e].gmf,cwp[e].itf,cwp[e].btf);
      cwp[e].tnr=max;
      }//end of if(cwp[e].rnt==0)
     }//end of if(cwp[e].stat==1)
    }//end of if (nprn<temp1)
   } //end of for e=0

  //starting the units.
  for (int ei=0;ei<3;ei++)
   {
    if ((unt[ei].stat==1)&&(nurn<temp0))
     {
     unt[ei].stat=run;
     nurn++;
     nsu--;

     for (int ej=0;ej<4;ej++)
      {
      if (eqp[ei][ej].stat==1)
     {
     eqp[ei][ej].stat=run;
     if(eqp[ei][ej].rnt==0)
      {
      eqp[ei][ej].tnf=time_to_failrep(eqp[ei][ej].gmf,eqp[ei][ej].itf,
                                        eqp[ei][ej].btf);
      eqp[ei][ej].tnr=max;
      }
     }
      if (ej<2)
```

```
      {
      if (eqps[ei][ej][0].stat==1)
       {
       eqps[ei][ej][0].stat=run;
       if (eqps[ei][ej][0].rnt==0)
        {
        eqps[ei][ej][0].tnf=time_to_failrep(eqps[ei][ej][0].gmf,
                             eqps[ei][ej][0].itf,eqps[ei][ej][0].btf);
        eqps[ei][ej][0].tnr=max;
        }
       }
      else if(eqps[ei][ej][0].stat==2)
       {
       eqps[ei][ej][1].stat=run;
       if (eqps[ei][ej][1].rnt==0)
        {
        eqps[ei][ej][1].tnf=time_to_failrep(eqps[ei][ej][1].gmf,
                             eqps[ei][ej][1].itf,eqps[ei][ej][1].btf);
        eqps[ei][ej][1].tnr=max;
        }
       }//end of else if
      }//end of if ej<2
      }//end of for ej=0
     }//end of if unit[ei].stat==1
    }//end of for ei=0
   //}//end of if ((nurn<nurq)&&(nsu>0))

  }//end of etr loop.

else if ((etf<=rmt)&&(etf<=etm)&&(etf<etr))
 {outfile<<" etf loop ";
 rmt-=etf;
 st+=etf;
 gen[td]+=generation (nurn, flow[td], etf);
 //stopping the equipments which attain the age of faliure.
 for (int fi=0; fi<3; fi++)
  {
  update_time (cwp[fi], etf);
  //stopping the pumps.
  if ((cwp[fi].stat==0)&&(cwp[fi].rnt==cwp[fi].tnf))
   {
   cwp[fi].crnt+=cwp[fi].rnt;
   cwp[fi].rnt=0;
   cwp[fi].stat=dwn;
   cwp[fi].tnf=max;
   cwp[fi].tnr=time_to_failrep(cwp[fi].gmr, cwp[fi].itr, cwp[fi].btr);
   cwp[fi].no_of_mnt++;
   nprn--;
   npdn++;
   }
   int temp5 =0;
   //stopping the equipments.
   for (int fj=0; fj<4; fj++)
    {
    update_time (eqp[fi][fj], etf);
    if ((eqp[fi][fj].stat==0)&&(eqp[fi][fj].rnt==eqp[fi][fj].tnf))
     {
     eqp[fi][fj].no_of_mnt++;
     eqp[fi][fj].crnt+=eqp[fi][fj].rnt;
     eqp[fi][fj].rnt=0;
     eqp[fi][fj].stat=dwn;
```

```
        eqp[fi][fj].tnf=max;
        eqp[fi][fj].tnr=time_to_failrep(eqp[fi][fj].gmr, eqp[fi][fj].itr,
                                            eqp[fi][fj].btr);
        temp5++;
        }
       if (fj<2)
        {
        for (int fk=0; fk<2; fk++)
       {
       update_time (eqps[fi][fj][fk], etf);
       if ((eqps[fi][fj][fk].stat==0)&&
                        (eqps[fi][fj][fk].rnt==eqps[fi][fj][fk].tnf))
       {
       int tmp5 = ((fk+3)%2); outfile<<" fk="<<fk<<" tmp="<<tmp5;
       eqps[fi][fj][fk].no_of_mnt++;
       eqps[fi][fj][fk].crnt+=eqps[fi][fj][fk].rnt;
       eqps[fi][fj][fk].rnt=0;
       eqps[fi][fj][fk].stat=dwn;
       eqps[fi][fj][fk].tnf=max;
       eqps[fi][fj][fk].tnr=time_to_failrep(eqps[fi][fj][fk].gmr,
                        eqps[fi][fj][fk].itr, eqps[fi][fj][fk].btr);
       temp5++;
       if (eqps[fi][fj][tmp5].stat==1)//start the standby gov/lub pump.
        {
        eqps[fi][fj][tmp5].stat=run;
        temp5--;
        if (eqps[fi][fj][tmp5].rnt==0)
         {
         eqps[fi][fj][tmp5].tnf=time_to_failrep(eqps[fi][fj][tmp5].gmf,
                        eqps[fi][fj][tmp5].itf, eqps[fi][fj][tmp5].btf);
         eqps[fi][fj][tmp5].tnr=max;
         }
        }//end of if (eqps[fi][fj][tmp5].stat==1)
       }//end of if
((eqps[fi][fj][fk].stat==0)&&(eqps[fi][fj][fk].rnt==eqps[fi][fj][fk].tnf))
        }//end of for (int fk=0; fk<2; fk++)
       }//end of if (fj<2)
      }//end of for (int fj=0; fj<4; fj++)
     if (temp5>0)  //stop other equipments of the unit.
      {
      if (unt[fi].stat==0)
       {
       unt[fi].stat=dwn;
       nurn--;
       nudn++;
       }
      for (int jf=0; jf<4; jf++)
       {
       if (eqp[fi][jf].stat==0)   eqp[fi][jf].stat=stby;
       if (jf<2)
        {
        for (int kf=0; kf<2; kf++)
       {
       if (eqps[fi][jf][kf].stat==0)  eqps[fi][jf][kf].stat=stby;
       }
        }//end of if (jf<2)
       }//end of for (int jf=0; jf<4; jf++)
      }//end of if (temp5>0)
     }//end of for (int fi=0; fi<3; fi++)

  int temp;
```

```
  if ((nurn+nsu)>=nurq) temp=nurq;
   else if ((nurn+nsu)<nurq) temp=(nurn+nsu);
  int temp0=unitrun((nprn+npsb), temp);
  int temp1=pumpreq(temp0);

// start pumps &units if required.
   for (int g=0; g<3; g++)
    {
    //starting condition for pumps
    if ((nprn<temp1)&&((nprn+npdn)<3))
     {
     if(cwp[g].stat==1)
      {
      cwp[g].stat=run;
      npsb--;
      nprn++;
      if(cwp[g].rnt==0)
       {
       cwp[g].tnf=time_to_failrep(cwp[g].gmf,cwp[g].itf,cwp[g].btf);
       cwp[g].tnr=max;
       }
      }
     }
    } //end of for g=0
   if (nprn>temp1)
    {
    for (int gg=2; gg>=0; gg--)
     {
     if (cwp[gg].stat==0)
      {
      cwp[gg].stat=stby;
      nprn--;
      npsb++;
      }
     }
    }

  //starting the units.
  for (int gi=0; gi<3; gi++)
   {
   if ((unt[gi].stat==1)&&(nurn<temp0))//(pmpcon==0))
    {
    unt[gi].stat=run;
    nurn++;
    nsu--;
    for (int gj=0; gj<4; gj++)
     {
     if (eqp[gi][gj].stat==1)
    {
    eqp[gi][gj].stat=run;
    if(eqp[gi][gj].rnt==0)
     {
     eqp[gi][gj].tnf=time_to_failrep(eqp[gi][gj].gmf,eqp[gi][gj].itf,
                                             eqp[gi][gj].btf);
     eqp[gi][gj].tnr=max;
     }
    }
    if (gj<2)
     {
     if ((eqps[gi][gj][0].stat==1)&&(eqps[gi][gj][1].stat==1))
      {
```

```
      if (eqps[gi][gj][0].rnt>=eqps[gi][gj][1].rnt)
       {
       eqps[gi][gj][0].stat=run;
       if (eqps[gi][gj][0].rnt==0)
        {
        eqps[gi][gj][0].tnf=time_to_failrep(eqps[gi][gj][0].gmf,
                          eqps[gi][gj][0].itf,eqps[gi][gj][0].btf);
        eqps[gi][gj][0].tnr=max;
        }
       }
       else if (eqps[gi][gj][0].rnt<eqps[gi][gj][1].rnt)
        eqps[gi][gj][1].stat=run;
      }
     else if ((eqps[gi][gj][0].stat==1)&&(eqps[gi][gj][1].stat==2))
      {
      eqps[gi][gj][0].stat=run;
      if (eqps[gi][gj][0].rnt==0)
       {
       eqps[gi][gj][0].tnf=time_to_failrep(eqps[gi][gj][0].gmf,
                          eqps[gi][gj][0].itf,eqps[gi][gj][0].btf);
       eqps[gi][gj][0].tnr=max;
       }
      }
     else if((eqps[gi][gj][0].stat==2)&&(eqps[gi][gj][1].stat==1))
      {
      eqps[gi][gj][1].stat=run;
      if (eqps[gi][gj][1].rnt==0)
       {
       eqps[gi][gj][1].tnf=time_to_failrep(eqps[gi][gj][1].gmf,
                          eqps[gi][gj][1].itf,eqps[gi][gj][1].btf);
       eqps[gi][gj][1].tnr=max;
       }
      }
     }//end of if gj<2
     }//end of for gj=0
    }//end of if unit[ai].stat==1
   }//end of for gi=0
  //}//end of if ((nurn<nurq)&&(nsu>0))

for (int ii=2; ii>=0; ii--)
 {
 if ((int(cwp[ii].mf*a1)<=(cwp[ii].rnt))&&(cwp[ii].stat==1))
  {
  cwp[ii].crnt+=cwp[ii].rnt;
  cwp[ii].no_of_omnt++;
  cwp[ii].rnt=0;
  cwp[ii].stat=dwn;
  cwp[ii].tnf=max;
  cwp[ii].tnr=cwp[ii].prmt;
  npdn++;
  npsb--;
  }
 if (unt[ii].stat==2)
  {
  //int temp5=0;
  for (int ij=0; ij<4; ij++)
   {
   if ((int(eqp[ii][ij].mf*a1)<=(eqp[ii][ij].rnt))&&(eqp[ii][ij].stat==1))
    {
    eqp[ii][ij].crnt+=eqp[ii][ij].rnt;
    eqp[ii][ij].no_of_omnt++;
```

```
      eqp[ii][ij].rnt=0;
      eqp[ii][ij].stat=dwn;
      eqp[ii][ij].tnf=max;
      eqp[ii][ij].tnr=eqp[ii][ij].prmt;
      //temp5++;
      }
     if (ij<2)
      {
      //int temp6=0;
      for (int ik=0; ik<2; ik++)
       {
       if ((int(eqps[ii][ij][ik].mf*a1)<=(eqps[ii][ij][ik].rnt))
                                &&(eqps[ii][ij][ik].stat==1))
      {
      eqps[ii][ij][ik].crnt+=eqps[ii][ij][ik].rnt;
      eqps[ii][ij][ik].no_of_omnt++;
      eqps[ii][ij][ik].rnt=0;
      eqps[ii][ij][ik].stat=dwn;
      eqps[ii][ij][ik].tnf=max;
      eqps[ii][ij][ik].tnr=eqp[ii][ij].prmt;
      //temp6++;
      }
       }//end of for (int ik=0; ik<2; ik++)
      //if (temp6==2) temp5++;
      }// end of if (ij<2)
     }// end of for (int ij=0; ij<4; ij++)
    //if (temp5>0) unt[ii].stat=2;
    }//end of if ((unt[ii].stat==2)||(unt[ii].stat==1))
   }//end of for (int ii=2; ii>=0; ii--)
  }//end of  else if ((etf<=rmt)&&(etf<=etm)&&(etf<etr))

// preventive maintenance loop
else if ((etm<=rmt)&&(etm<etr)&&(etm<etf))
 {outfile<<" etm loop ";
 rmt-=etm;
 st+=etm;
 gen[td]+=generation (nurn, flow[td], etm);
 int pscnt=(3-nprn-npdn);
 int pstp=0;outfile<<" nprn="<<nprn<<" npdn="<<npdn<<" pscnt="<<pscnt;
 for (int li=0; li<3; li++)
  {
  update_time (cwp[li], etm);
  if (((pscnt-pstp)>0)&&(int(cwp[li].mf*a2)<=cwp[li].rnt)&&(cwp[li].stat==0))
   {
   cwp[li].crnt+=cwp[li].rnt;
   cwp[li].no_of_pmnt++;
   cwp[li].stat=dwn;
   cwp[li].rnt=0;
   cwp[li].tnf=max;
   cwp[li].tnr=cwp[li].prmt;
   npdn++;
   nprn--;
   pstp++;
   if (mntcp[li]==li) mntcp[li]=max;
   }
  else if ((int(cwp[li].mf*a2)<=cwp[li].rnt)&&(cwp[li].stat==0))
   mntcp[li]=li;
  for (int lj=0; lj<4; lj++)
   {
   update_time (eqp[li][lj], etm);
   if (lj<2)
```

```
    {
    for (int lk=0; lk<2; lk++)
     {
     update_time (eqps[li][lj][lk], etm);
     int kl;
     if (lk==0) kl=1;
      else if (lk==1) kl=0;
     if ((int(eqps[li][lj][lk].mf*a2)<=(eqps[li][lj][lk].rnt))&&
           (eqps[li][lj][lk].stat==0)&&(eqps[li][lj][kl].stat==1))
      {
      eqps[li][lj][lk].crnt+=eqps[li][lj][lk].rnt;
      eqps[li][lj][lk].no_of_pmnt++;
      eqps[li][lj][lk].stat=dwn;
      eqps[li][lj][lk].tnf=max;
      eqps[li][lj][lk].tnr=eqps[li][lj][lk].prmt;
      eqps[li][lj][lk].rnt=0;
      eqps[li][lj][kl].stat=run;
      if (eqps[li][lj][kl].rnt==0)
     {
     eqps[li][lj][kl].tnf=time_to_failrep(eqps[li][lj][kl].gmf,
                           eqps[li][lj][kl].itf, eqps[li][lj][kl].btf);
     eqps[li][lj][kl].tnr=max;
     }
      if (mntcs[li][lj][lk]==(li*100+lj*10+lk))  mntcs[li][lj][lk]=max;
      }
     else if ((int(eqps[li][lj][lk].mf*a2)<=(eqps[li][lj][lk].rnt))&&
           (eqps[li][lj][lk].stat==0)&&(eqps[li][lj][kl].stat==2))
      mntcs[li][lj][lk]=(li*100+lj*10+lk);
     }//end of for (int lk=0; lk<2; lk++)
    }//end of if (lj<2)
   }//end of for (int lj=0; lj<4; lj++)
  }//end of for (int li=0; li<3; li++)
 for (int il=3; il>=0; il--)
  {
  if ((pstp>0)&&(cwp[il].stat==1))
   {
    cwp[il].stat=run;
    pstp--;
    nprn++;
    npsb--;
    if (cwp[il].rnt==0)
     {
     cwp[il].tnf=time_to_failrep(cwp[il].gmf, cwp[il].itf, cwp[il].btf);
     cwp[il].tnr=max;
     }
    }//end of if (pstp>0)
   }//end of for (int il=3; il>=0; il--)
 }//end of else if ((etm<=rmt)&&(etm<etr)&&(etm<etf))

else if ((rmt<etr)&&(rmt<etf)&&(rmt<etm))
 {outfile<<" rmt loop ";
 st+=rmt;
 gen[td]+=generation (nurn, flow[td], rmt);
 for (int mi=0; mi<3; mi++)
  {
  update_time (cwp[mi], rmt);
  for (int mj=0; mj<4; mj++)
   {
   update_time (eqp[mi][mj], rmt);
   if (mj<2)
    {
```

```
    for (int mk=0; mk<2; mk++)
      update_time (eqps[mi][mj][mk], rmt);
    }//end of if (mj<2)
   }// end of for (int mj=0; mj<4; mj++)
  }// end of for (int mi=0; mi<3; mi++)
 //rmt=0;
 }//end of else if ((rmt<etr)&&(rmt<etf)&&(rmt<etm))
// checking for preventive maintenance time of all equipments.
 if ((rep_maint>0)&&(etr<etm)&&(etf<etm))
  {outfile<<" rep_maint ";
  int pscnt=(3-nprn-npdn);
  int pstp;
  for (int ni=0; ni<3; ni++)
   {
   //update_time (cwp[ni], etm);
   if (((pscnt-pstp)>0)&&(int(cwp[ni].mf*a2)<=cwp[ni].rnt)&&(cwp[ni].stat==0))
    {
    cwp[ni].crnt+=cwp[ni].rnt;
    cwp[ni].no_of_pmnt++;
    cwp[ni].stat=dwn;
    cwp[ni].rnt=0;
    cwp[ni].tnf=max;
    cwp[ni].tnr=cwp[ni].prmt;
    npdn++;
    nprn--;
    pstp++;
    if (mntcp[ni]==ni) mntcp[ni]=max;
    }
   else if ((int(cwp[ni].mf*a2)<=cwp[ni].rnt)&&(cwp[ni].stat==0))
                                  mntcp[ni]=ni;
   for (int nj=0; nj<4; nj++)
    {
    //update_time (eqp[ni][nj], etm);
    if (nj<2)
     {
     for (int nk=0; nk<2; nk++)
      {
      //update_time (eqps[ni][nj][nk], etm);
      int kn=((nk+3)%2);
      if ((int(eqps[ni][nj][nk].mf*a2)<=(eqps[ni][nj][nk].rnt))&&
            (eqps[ni][nj][nk].stat==0)&&(eqps[ni][nj][kn].stat==1))
     {
     eqps[ni][nj][nk].crnt+=eqps[ni][nj][nk].rnt;
     eqps[ni][nj][nk].no_of_pmnt++;
     eqps[ni][nj][nk].stat=dwn;
     eqps[ni][nj][nk].tnf=max;
     eqps[ni][nj][nk].tnr=eqps[ni][nj][nk].prmt;
     eqps[ni][nj][nk].rnt=0;
     eqps[ni][nj][kn].stat=run;
     if (eqps[ni][nj][kn].rnt==0)
      {
      eqps[ni][nj][kn].tnf=time_to_failrep(eqps[ni][nj][kn].gmf,
                         eqps[ni][nj][kn].itf, eqps[ni][nj][kn].btf);
      eqps[ni][nj][kn].tnr=max;
      }
     if (mntcs[ni][nj][nk]==(ni*100+nj*10+nk)) mntcs[ni][nj][nk]=max;
     }
    else if ((int(eqps[ni][nj][nk].mf*a2)<=(eqps[ni][nj][nk].rnt))&&
           (eqps[ni][nj][nk].stat==0)&&(eqps[ni][nj][nk].stat==2))
                 mntcs[ni][nj][nk]=(ni*100+nj*10+nk);
     }//end of for (int nk=0; nk<2; nk++)
```

```
      }//end of if (nj<2)
     }//end of for (int nj=0; nj<4; nj++)
    }//end of for (int ni=0; ni<3; ni++)
   for (int in=3; in>=0; in--)
    {
    if (pstp>0)
     {
     if (cwp[in].stat==1)
      {
      cwp[in].stat=run;
      pstp--;
      nprn++;
      npsb--;
      if (cwp[in].rnt==0)
       {
       cwp[in].tnf=time_to_failrep(cwp[in].gmf, cwp[in].itf, cwp[in].btf);
       cwp[in].tnr=max;
       }
      }//end of if (cwp[in].stat==1)
     }//end of if (pstp>0)
    }//end of for (int in=3; in>=0; in--)
   }//end of if ((rep_maint>0)&&(etr<etm)&&(etf<etm))

  // Finding the values of etr & etf.
  etf = max;
  etr = max;

  for (int oi=0; oi<3; oi++)
   {
   if ((etf>(cwp[oi].tnf-cwp[oi].rnt))&&(cwp[oi].stat==0))
     etf = cwp[oi].tnf-cwp[oi].rnt;
   if ((etr>(cwp[oi].tnr-cwp[oi].dnt))&&(cwp[oi].stat==2))
     etr = cwp[oi].tnr-cwp[oi].dnt;
   for (int oj=0; oj<4; oj++)
    {
    if ((etf>(eqp[oi][oj].tnf-eqp[oi][oj].rnt))&&(eqp[oi][oj].stat==0))
       etf = eqp[oi][oj].tnf-eqp[oi][oj].rnt;
    if ((etr>(eqp[oi][oj].tnr-eqp[oi][oj].dnt))&&(eqp[oi][oj].stat==2))
       etr = eqp[oi][oj].tnr-eqp[oi][oj].dnt;

   if (oj<2)
    {
    for (int ok=0; ok<2; ok++)
     {
     if ((etf>(eqps[oi][oj][ok].tnf-eqps[oi][oj][ok].rnt))&&
                                (eqps[oi][oj][ok].stat==0))
      etf = eqps[oi][oj][ok].tnf-eqps[oi][oj][ok].rnt;
     if ((etr>(eqps[oi][oj][ok].tnr-eqps[oi][oj][ok].dnt))&&
                                (eqps[oi][oj][ok].stat==2))
      etr = eqps[oi][oj][ok].tnr-eqps[oi][oj][ok].dnt;
     } //end of for (int ok=0; ok<2; ok++)
    } //end of if (oj<2)
   }//end of for (int oj=0; oj<4; oj++)
  }//end of for (int oi=0; oi<3; oi++)
 //cout<<"ETM="<<etm;
 etm=max;
 // finding the value of etm.
 for (int pi=0; pi<3; pi++)
  {
  /*cout<<endl<<" cwp["<<pi<<"]="<<cwp[pi].stat<<" "<<cwp[pi].mf*a2;*/
  if (mntcp[pi]==pi) rep_maint++;
```

```
   else if ((etm>(int(cwp[pi].mf*a2)-cwp[pi].rnt))&&(cwp[pi].stat==0))
      etm =(int(cwp[pi].mf*a2)-cwp[pi].rnt);
  for (int pj=0; pj<2; pj++)
   {
   for (int pk=0; pk<2; pk++)
    {
    /*cout<<" eqp["<<pi<<"]["<<pj<<"]["<<pk<<"]="<<eqps[pi][pj][pk].stat
      <<" "<<eqps[pi][pj][pk].mf*a2; */
    if (mntcs[pi][pj][pk]==(pi*100+pj*10+pk))   rep_maint++;
     else if ((etm>(int(eqps[pi][pj][pk].mf*a2)-eqps[pi][pj][pk].rnt))&&
                                  (eqps[pi][pj][pk].stat==0))
       etm = (int(eqps[pi][pj][pk].mf*a2)-eqps[pi][pj][pk].rnt) ;
    }
   }
  }
 // writing results in output file
/*  for (int w=0; w<3; w++)
   {
   outfile<<" unit["<<w<<"].st=";
   if (unt[w].stat==0) outfile<<"run ";
    else if (unt[w].stat==1) outfile<<"stby ";
     else if (unt[w].stat==2) outfile<<"stop ";
   }

outfile<<endl<<"_____________________________________________________________
_________"
            <<"\n     status   rntm  dntm   mf  mf*a1  mf*a2  tmofnxtflr
tmofnxtrpr"
       <<endl<<"------------------------------------------------------------
---------";
  for (int wo=0;wo<3;wo++)
   {
   outfile<<"\ncwp"<<wo<<"  =";
   if (cwp[wo].stat==0) outfile<<" run ";
    else if (cwp[wo].stat==1) outfile<<" stby";
     else if (cwp[wo].stat==2) outfile<<" stop";
   outfile<<setw(6)<<cwp[wo].rnt<<setw(6)<<cwp[wo].dnt
        <<setw(6)<<(int(cwp[wo].mf))<<setw(7)
        <<int((cwp[wo].mf)*a1-cwp[wo].rnt)<<setw(7)
        <<int((cwp[wo].mf)*a2-cwp[wo].rnt)
        <<setw(10)<<int(cwp[wo].tnf)<<setw(10)<<int(cwp[wo].tnr);
  }
  for (int wx=0; wx<3; wx++)
   {

   for (int wy=0; wy<4; wy++)
    {
    outfile<<"\neqp"<<wx<<wy<<" =";
    if (eqp[wx][wy].stat==0) outfile<<" run ";
     else if (eqp[wx][wy].stat==1) outfile<<" stby";
      else if (eqp[wx][wy].stat==2) outfile<<" stop";
    outfile<<setw(6)<<eqp[wx][wy].rnt<<setw(6)<<eqp[wx][wy].dnt
         <<setw(6)<<(int(eqp[wx][wy].mf))<<setw(7)
         <<int((eqp[wx][wy].mf)*a1-eqp[wx][wy].rnt)<<"   ****"
         <<setw(10)<<eqp[wx][wy].tnf<<setw(10)<<eqp[wx][wy].tnr;

    }
   for (int wi=0; wi<2; wi++)
    {
    for (int wj=0; wj<2; wj++)
     {
```

```
    outfile<<"\neqp"<<wx<<wi<<wj<<"=";
    if (eqps[wx][wi][wj].stat==0) outfile<<" run ";
     else if (eqps[wx][wi][wj].stat==1) outfile<<" stby";
       else if (eqps[wx][wi][wj].stat==2) outfile<<" stop";
    outfile<<setw(6)<<eqps[wx][wi][wj].rnt
         <<setw(6)<<eqps[wx][wi][wj].dnt<<setw(6)
         <<(int(eqps[wx][wi][wj].mf))<<setw(7)
         <<int((eqps[wx][wi][wj].mf)*a1-eqps[wx][wi][wj].rnt)
         <<setw(7)<<int((eqps[wx][wi][wj].mf)*a2-eqps[wx][wi][wj].rnt)
         <<setw(10)<<int(eqps[wx][wi][wj].tnf)
         <<setw(10)<<int(eqps[wx][wi][wj].tnr);
    }
   }
  }
 outfile<<endl<<"event No.="<<count<<" flow="<<flow[td]<<" st="<<st
      <<" etr="<<etr<<"  etm="<<etm<<" etf="<<etf<<" rmt="<<rmt
      <<"\ngen["<<td<<"]="<<setiosflags(ios::fixed)
      <<setiosflags(ios::showpoint)<<setprecision(2)<<gen[td]
      <<" pg["<<td<<"]="<<setiosflags(ios::fixed)<<pg[td]
      <<" nurn="<<nurn<<" nsu="<<nsu
      <<" nudn="<<nudn<<" nprn="<<nprn<<" npdn="<<npdn<<" npsb="<<npsb

<<"\n=============================================================="
      <<endl;  */

  //cout<<"\n enter next count=";
  count++;
 }//end of do loop
  //while (count<10);
 while (st<tendlyhrs);
 total_gen+=gen[td];
 total_exgen+=pg[td];
 }//end of for (int td=0; td<36; td++)
}//end of for (int yr=0; yr<3; yr++)
 outfile<<endl<<"Total generation="<<total_gen<<" MWhr"
     <<endl<<"Total expected generation="<<total_exgen<<" MWhr"
     <<endl<<"EENS="<<(total_exgen-total_gen)<<" MWhr";

outfile<<endl<<"________________________________________________________________
_________"
            <<"\n      cum_rnt     Cum_dnt      No_of_mnt       No_Of_pmnt"
       <<endl<<"-------------------------------------------------------------
---------";
  //outfile<<endl;
  for (int wa=0;wa<3;wa++)
   {
   outfile<<"\ncwp"<<wa<<"  =";

   outfile<<setw(6)<<cwp[wa].crnt<<setw(6)<<cwp[wa].cdnt
        <<setw(6)<<cwp[wa].no_of_mnt<<setw(7)
        <<cwp[wa].no_of_omnt<<setw(6)<<cwp[wa].no_of_pmnt;
   }
  for (int wb=0; wb<3; wb++)
   {

   for (int wc=0; wc<4; wc++)
    {
    outfile<<"\neqp"<<wb<<wc<<" =";

    outfile<<setw(6)<<eqp[wa][wb].crnt<<setw(6)<<eqp[wa][wb].cdnt
         <<setw(6)<<eqp[wa][wb].no_of_mnt<<setw(7)
```

```
        <<eqp[wa][wb].no_of_omnt<<setw(6)<<eqp[wa][wb].no_of_pmnt;

    }
   for (int wi=0; wi<2; wi++)
    {
    for (int wj=0; wj<2; wj++)
     {
     outfile<<"\neqp"<<wa<<wi<<wj<<"=";

     outfile<<setw(6)<<eqps[wa][wi][wj].crnt
          <<setw(6)<<eqps[wa][wi][wj].cdnt<<setw(6)
          <<eqps[wa][wi][wj].no_of_mnt<<setw(7)
          <<eqps[wa][wi][wj].no_of_omnt<<setw(6)
          <<eqps[wa][wi][wj].no_of_pmnt;
     }
    }
   }
 cout<<endl<<"Total generation="<<total_gen<<" MWhr"
     <<"  "<<"Total Annual generation="<<(total_gen)/3<<" MWhr"
     <<endl<<"Total expected generation="<<total_exgen<<" MWhr"
     <<"  "<<"Total Annual expected generation="<<(total_exgen)/3<<" MWhr"
     <<endl<<"EENS="<<(total_exgen-total_gen)<<" MWhr"
     <<"  "<<"AEENS="<<(total_exgen-total_gen)/3<<" MWhr";
 getche();

}//end of void main()

  int unitrun (int npr, int nur)
   {
   int nos;
   if (npr==0) nos=0;
    else if ((npr==1)&&(nur>0)) nos=1;
     else if ((npr==1)&&(nur==0)) nos=0;
      else if ((npr==2)&&(nur==0)) nos=0;
       else if ((npr==2)&&(nur==1)) nos=1;
      else if ((npr==2)&&(nur>1)) nos=nur;
       else if (npr>2) nos=nur;
        return nos;
   }

 int pumpreq (int nu)
   {
   int temp3;
   if (nu==0) temp3=0;
    else if (nu==1) temp3=1;
     else if (nu>1) temp3=2;
   return temp3;
   }

  float generation (int nu, float flw, int tm)
   {
   float genr;
   if (flw>=(float(nu)*112.43))     genr=(float(nu)*7.5*tm);
    else if (flw<(float(nu)*112.43)) genr=(0.001*9.81*0.85*8.0*flw*tm);
   return genr;
   }

  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.stat==0)
  {
```

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