ELSEVIER

Journal of Hazardous Materials A75 (2000) 1–27

Journal of Hazardous Materials

www.elsevier.nl/locate/jhazmat

Analytical simulation and PROFAT II: a new methodology and a computer automated tool for fault tree analysis in chemical process industries

Faisal I. Khan, S.A. Abbasi *

Computer-Aided Environment Management Unit, Centre for Pollution Control and Energy Technology, Pondicherry University, Kalapet, Pondicherry 605 014, India

Received 14 October 1999; received in revised form 14 January 2000; accepted 18 January 2000

Abstract

Fault tree analysis (FTA) is based on constructing a hypothetical tree of base events (initiating events) branching into numerous other sub-events, propagating the fault and eventually leading to the top event (accident). It has been a powerful technique used traditionally in identifying hazards in nuclear installations and power industries. As the systematic articulation of the fault tree is associated with assigning probabilities to each fault, the exercise is also sometimes called probabilistic risk assessment. But powerful as this technique is, it is also very cumbersome and costly, limiting its area of application.

We have developed a new algorithm based on analytical simulation (named as AS-II), which makes the application of FTA simpler, quicker, and cheaper; thus opening up the possibility of its wider use in risk assessment in chemical process industries. Based on the methodology we have developed a computer-automated tool. The details are presented in this paper.

Keywords: Probabilistic risk assessment; Fault tree analysis; Reliability analysis; Risk analysis

1. Introduction

The technique of fault tree analysis (FTA) has been extensively applied to risk assessment in nuclear reactors and power industry; however, its use in forecasting accidents in chemical process industries has been surprisingly limited [1,2]. The possible

* Corresponding author. Tel.: +91-413-665262/665263; fax: +91-413-665267/665265.
E-mail address: prof_abbasi@vsnl.com (S.A. Abbasi).

0304-3894/00/$ - see front matter
PII: S0304-3894(00)00169-2

reasons are: (i) the inherently complex nature of the step associated with the construction of fault tree; (ii) the complexity of fault propagation mechanism; (iii) difficult-to-obtain failure/reliability data necessary for accurate analysis; (iv) requirements of comparatively large computation times, hence costs; and (v) less reliable results due to large uncertainty involved in the inputs data (reliability/failure data).

In an attempt to overcome some of these problems, we had developed a software package called PROFAT (PRObabilistic FAult Tree) [3]. We have now refined the methodology on which PROFAT was based and also recoded the package to incorporate several new features in terms of technical content as well as user-savvy. The details are presented here.

2. FTA in the context of risk analysis

Risk analysis has long been a familiar term in the domains of the chemical/power industries. In the field of chemical process industries, it was introduced in 1970s in the wake of increasing frequency and size of accidents occurring all over the world — for example at Seveso in 1972, Flixborough in 1976, Bhopal in 1984, Piper Alpha, 1990, Visakhapatnam, 1997. Techniques and methodologies proposed included checklist, hazard and operability study (HAZOP), failure mode effect analysis (FMEA), hazard survey, HAZAN (hazard analysis). As such, no individual technique has been found suitable for comprehensive risk analysis of chemical process industries [4,5]. Combinations of different techniques have been proposed to overcome this limitation. Among these, quantitative risk analysis [6–8], probabilistic risk analysis [9,10], and probabilistic safety analysis [11–13], have been well recognised for risk/safety analysis of chemical process industries. Recently, Khan and Abbasi [1,2] have proposed a system ORA (optimal risk analysis) for risk analysis in process industries that attempts to cover all the three dimensions: qualitative, quantitative, and probabilistic of risk analysis.

Considering that by definition risk is a combination of hazard (damage/harm) potential and the probability of occurrence of such a hazard, FTA is potentially one of the most powerful techniques of risk analysis as it estimates the probability/frequency of accidents. Furthermore, besides estimating the probability of occurrence of an accident, FTA also throws light on probable causes of such an accident. As FTA focuses on probabilities of events, it is also often called ‘‘probabilistic risk analysis’’.

In summary, application of FTA helps in

(i) directing the analyst to ferret out failures deductively;
(ii) pointing out aspects of the system relevant to significant failures;
(iii) providing a graphical aid, giving visibility to fault dependencies
(iv) providing options for qualitative as well as quantitative system reliability analysis;
(v) allowing the analyst to concentrate on one particular system failure at a time;
(vi) providing the analyst with an insight into the system behaviour.

According to Lees [4], FTA is a sophisticated form of reliability assessment and it requires considerable time and effort by skilled analysts. Although it is the best tool

available for a comprehensive analysis, it is not fool-proof, and in particular, it does not, by itself, assure detection of all failures, especially common failures.

Attempts have been made to make FTA more robust, user-friendly and less time-consuming, notably by: Lees [4], Lapp [10], Lapp and Powers [14,15], Power and Tompkins [16], Shafaghi [17], Hauptmanns and Yllera [18], Camarinpoulous and Yllera [19], Lai et al. [20], Thangamani [21], Bossche [22] and Rauzy [23]. We have now worked out an algorithm to evaluate FTA, which is easier to apply, faster in computation, gives reliable results, and is more robust as it can tolerate a greater degree of uncertainty in the input reliability data. The details are presented below.

3. Elements of FTA

A fault tree is a logical and hierarchical model of an undesirable situation (accident) expressed in terms of all possible sequences and combinations of intermediate events and basic causes, leading to the ultimate undesired situation, or ''top event'' [24]. In general, a fault tree model of accident consists of three fundamental types of events described as:

(i) an event that corresponds to a primary failure in the system,
(ii) an event that corresponds to a non-primary failure that is not decomposed into more basic events,
(iii) an event that does not correspond to a fault or a failure but is an ordinary event existing inherently within the system.

FTA is both a qualitative and quantitative technique. It is qualitative in the sense that it identifies the individual basic events and paths that lead to the top event, and it is quantitative in the sense that it estimates the frequency or probability of occurrence of an event. In risk analysis, FTA is normally used in quantitative way, although it requires as an initial step qualitative study of the system under consideration. The fault tree reflects the outcome of the qualitative part of the analysis, in which questions of the type ''how an accident can take place'' are answered. This is done by a combination of different types of gates namely: the ''OR'' and the ''AND'' gates. A few more types of gates are defined exclusively for fault tree application in process industries, e.g. inhibit gate, priority gate.

There may be only two states of the basic event: either true or false, which implies two possible states for the undesired event, its occurrence (true) and its non-occurrence (false), respectively. The two states are associated with certain probabilities; in case of technical components the same are generally obtained by evaluating the operating behaviour of a great number of similar components. The probability of the undesired event is calculated using the probabilities of the basic events.

4. Application of FTA in assessing risk of accidents in chemical process industries

In order to apply FTA to process industries, all possible initiating events that are capable of bringing out the pre-identified undesired events are identified. Subsequently,

a hierarchical tree is developed, taking undesired event as a top event and initiating causes as base events. For a typical chemical plant, base event may be a component malfunction or loss of integrity of a component.

Once fault tree has been developed for any undesired event in the plant or unit, it can be evaluated (solved) to identify the pathways (minimum combination of events) that would lead to the undesired event. Subsequently, using the failure frequency and reliability data of the base events, these pathways can be further evaluated to estimate the frequency of occurrence of top event. Evaluation can also be done to identify the vulnerable pathways and the vulnerable basic components. Thus, it is appropriate to say that application of FTA to process plant gives an idea about the vulnerability of each component, its contribution to cause the undesired event in relation to malfunctioning of other components, the top undesired event, and the frequency of occurrence of the undesired event [25–29].

4.1. Articulation of a fault tree

At first glance, articulation of a fault tree may appear a relatively simple exercise, but it is not so. Guidelines have been developed to avoid ''short-circuiting'' and other pitfalls in developing a fault tree [4,6,11,30]. An essential preliminary to construction of the fault tree is the definition and understanding of the system.

Fault tree for process plants falls into two main groups, as distinguished by the top event. The first group comprises those trees where the top event is a fault within the plant, including faults which can result in a toxic/flammable release or an internal explosion. In the second group, the top event is a hazardous event outside the plant: external fires, explosions, floods, wind, and earthquake. Both types of fault trees are developed considering the undesired event as top event and the initiating causes as base events. To control the undesired event there will be some protective or safety system. This will be considered as a barrier to the undesired event. The barrier will be distributed in different stages (steps of the process) and the incident will occur only if the barrier fails. The barrier may be having different levels of dependency; two or three barriers may simultaneously fail to cross one stage, or failure of one barrier may be enough to cross a particular stage. In fault tree, these barriers are represented by using logical gates, and are dependent on the basic events. The number of barriers may not be the same for all the process variables, as some variables may have more protective control system and some others lesser. The probability of occurrence of each event shall depend on the reliability data and number of barriers [14,15,22,24,25,31]. It may be that an event having more barriers with low reliability yields same occurrence probability as an event having one barrier with high reliability. The fault tree for a process is developed, taking these barriers into account.

4.2. Fault tree evaluation

After drawing up the fault tree, the cutsets and the probability of occurrence of the top event is to be estimated. A cutset is a collection of events that lead to the occurrence of the top event. When the collection of basic events (cutset) cannot be further reduced,

it is known as a minimal cutset for the fault tree. Another important property of a fault tree is the path set defined as sequence of basic events whose non-occurrence ensures the non-occurrence of the top event. A path set is minimal if the events in the paths cannot be further reduced. It is interesting to note that the minimal path-sets for a given tree can be obtained by employing a minimal cutset algorithm on the fault tree [20].

The key to the problems of high costs and lack of reliability of FTA lie in the method of evaluation of the fault tree. Available techniques for fault tree evaluation include Monte-Carlo simulation and analytical methods [18,19,24,25,32–35]. A brief description of these methods is presented below.

The evaluation of fault tree by Monte-Carlo and direct simulation methods is carried out by simulating the behaviour of components (base events) in accordance with the distribution of their lifetimes. This is done by generating random numbers with a uniform distribution over the interval of 0 to 1. Subsequently, the uniform random numbers are converted to exponential distribution as:

$$r_{ij} = -T_j^* \ln(1 - z_{ij}),$$

where z_{ij} is random number for component i for trial j and r_{ij} is the lifetime of component i in trial j.

The process of trials may be regarded as an inversion of the lifetime measurement for the component, which have led to the mean time to failure T_j. The life time r_{ij} is compared with the mission time for which the failure frequency is to be calculated. The component i is considered failed if $r_{ij} <$ mission time and function otherwise. For an undesired event to occur, this fact is recorded and the next trial, $j + 1$, is initiated. This process is continued for all the components as per the logical function of the fault tree. Since the process described is stochastic, the failure probability can only be indicated within certain confidence bounds, which can be narrowed by increasing the number of trials. For a predetermined degree of precision, the number of trials required rises with the inverse of the system unreliability, and may hence become prohibitive for very reliable system [14,18,20,36].

In analytical methods for evaluating fault trees, Boolean algebric operations are used in order to transform the tree in such a way that it is expressed in terms of its minimal cutsets by performing certain set of operations. In contrast with the preceding approach, this procedure does not require reliability data for obtaining the minimal cutsets of the tree, but only for calculating the probability of the undesired event. Hence, the process of obtaining the minimal cut analysis is not affected by the possible flaws in the data, as may be the case if the Monte-Carlo method is used for this purpose. However, sophisticated maintenance, repair, accident sequences and other strategies are difficult to implement here, as it requires cumbersome mathematics to represent these concepts. Quite often, evaluation of a fault tree having a large number of basic events by analytical method may lead to problems of memory allocation in computers [14,19,36].

These techniques have been successful in certain types of applications such as reliability studies. However, their applicability to risk assessment in chemical process industry is limited due to (i) requirement of precise reliability data, (ii) comparatively large computation-times, (iii) requirement of upmarket computing machines with high processing abilities, and (iv) comparatively less reliable results as computation is

strongly dependent on the accuracy of input data. Rauzy [23] has proposed advancement in Monte-Carlo simulation technique to reduce the computation time, while all other limitations remain as such.

Building upon the studies of Camarinpoulous and Yllera [19], Lai et al. [20], Bossche [27], Rauzy [23], Yllera [37] and Hauptamanns [38], we have proposed improvements in conventional analytical method by using concepts of latest Monte-Carlo simulation. The constraint of dependency on accurate reliability data has been overcome by incorporating fuzzy probability space theory.

5. A new algorithm for FTA: AS-II

The algorithm starts with the representation of fault tree in terms of Boolean matrix using Boolean algebra. Later this matrix is solved for minimal cutsets by using standard analytical procedure [4,38]. Generally, for a real-life industrial problem the Boolean matrix may become very large and thus matrix computation may exceed the computer memory. For this reason, we recommend that the whole problem should be divided into different modules. Among various moduling techniques, we find the one based on structure moduling as the most appropriate in the present context [17]. The modules are solved either simultaneously or one by one depending upon the requirement of the problem and the type of dependency among them. The algorithm of FTA using AS-II is shown in Fig. 1.

5.1. Boolean matrix formulation and its solution

The fault tree having basic events in series, parallel and/or a combination can be represented in a binary function as:

$$i = \begin{matrix} 1, & \text{if basic event is true} \\ 0, & \text{if basic event is not true} \end{matrix}$$

In a similar way, fault tree for the complete system (process unit) can be represented as a combination of these basic events in terms of Boolean matrix function:

$$Fk = \text{matrix}[\, jbi \,] \begin{vmatrix} 1b1 & 1b2 & 1b3 & \dots & 1bn \\ 2b1 & 2b2 & 2b3 & \dots & 2bn \\ . & . & . & \dots & . \\ mb1 & mb2 & mb3 & \dots & mbn \end{vmatrix}$$

where *jbi* represent the element of Boolean matrix, *j* represents the rows, and *i* represents column.

The system function ''*Fk*'' is defined as:

$$Fk = \begin{matrix} 1, & \text{system fails (undesired event occurs)} \\ 0, & \text{system is working (undesired event does not occur)} \end{matrix}$$

where *k* represents the number of times system function *F* is true or, in other words, the cutsets of the fault tree.

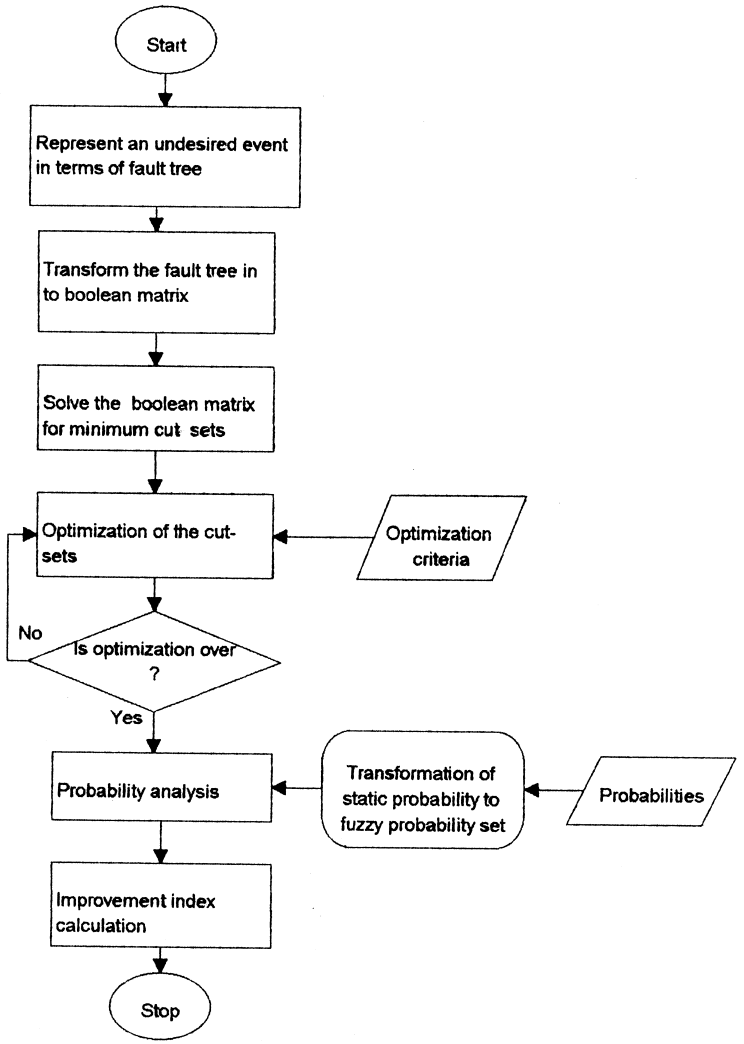


Fig. 1. Algorithm of analytical simulation methodology.

Once the complete system is represented in terms of *jbi* (basic events) using Boolean algebra, it is further evaluated using analytical method to identify the dependency of *F* on basic element *jbi*. The evaluation of dependency gives combination of basic events that lead the system function *F* to fail condition (undesired event to occur). These combinations of basic elements, also known as minimal cutsets, give insight to the system, as represented by the basic events that alone or in conjunctions with others can bring out the undesired event — through one or other path. For a real-life industrial problem, the number of these minimal cutsets may be very large. Hence we have introduced the concept of ‘‘optimal minimal cutsets’’, which represents the cutsets

having direct dependency on the top event (having frequency/probability of occurrence higher than a minimal value). The minimal cutsets can be optimised by using any standard optimisation procedure. However, we recommend the use of modified Fibonacci search method as the best tool to optimise the minimal cutsets. The optimal minimal cutsets can be represented as;

$$G(x^i) = \text{minimum}\left[U_j = 1{,}1\left[jg^i\right]\right] \text{ for } jg^i \geq \text{minimal criteria}$$

$$jg^i = PI_i = 1,\ m\left[p(x^i)\right], \text{where } m\left[p(x^i)\right]$$

$$= n(x^i) * \text{base duration} * \text{boundary limitations}$$

where i represents the event, j the number of cutsets, x^i state of variable, $P(x^i)$ probability of occurrences of an event, jg^i probability of a cutset, $n(x^i)$ frequency of failure rate of an event i, and $g(x^i)$ optimised minimal event set.

The optimised minimal cutsets are exceedingly important as they represent the core combination of events susceptible to cause undesirable event. For a typical fault tree that consists of a large number of basic events and gates, the optimal minimal cutsets of each module are linked with other modules according to their control barrier dependency. This step is repeated till all the modules of the problem are combined. This combination finally gives the optimal minimal cutsets for the complete system. If required, the final set of optimal minimal cutsets can be further optimised. However, this is an optional choice and need not be implemented for each and every problem. The algorithm for getting optimal minimal cutsets is presented in Fig. 2.

5.2. Analysis of optimal minimal cutsets using fuzzy set theory

In simple set theory, the probability of occurrence of top event, through optimal minimal cutsets, $P^{\text{Top}n}$ is described by a function of the basic events.

$$P^{\text{Top}} = h\left[P(x^1), P(x^2), \ldots, P(x^{i-1}) \ldots P(x^n)\right]$$

The probability of occurrence of top event when one event x^1 has been eliminated or made not to fail can be represented as

$$P^{\text{Top1}} = h\left[0, P(x^2), \ldots P(x^{i-1}) \ldots P(x^n)\right]$$

While considering these probabilities, an improvement factor has been defined as a factor representing an event's contribution to the undesired event. As per definition, improvement factor signifies the improvement in the probability of occurrence of the top event (undesired event). The higher the improvement factor for an event, the more likely it is to cause the undesired event.

Mathematically, improvement factor for an event is represented as

$$(P^{\text{Top}} - P^{\text{Top1}}) > 0 = \text{improvement factor}$$

The simple set theory requires exact values of probabilities of each event described by optimal minimal cutsets to estimate the probability of undesired event, and the improvement factor. Even small deviations (uncertainty) in these values (probability data of basic events) get accumulated and thus lead to a high deviation in the result.

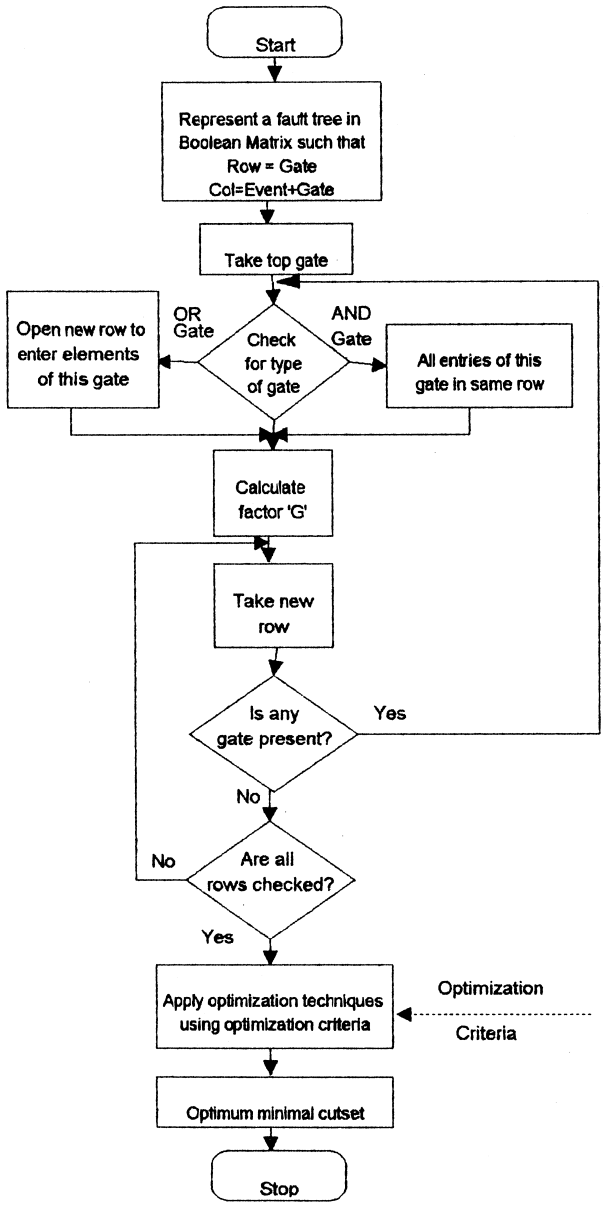


Fig. 2. Algorithm for optimal minimal cutsets.

As discussed earlier, getting exact values of failure data is very difficult. To counter this, we have used fuzzy probability space concept, which dilutes the dependency of analysis on reliability data. In the past, some authors used probability density function

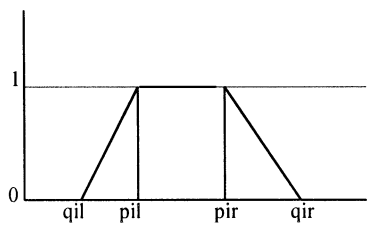
instead of single probability value; we believe this is not appropriate in the present context because:

(i) a density function has a distribution over longer dimension, which in many times causes error accumulation and thus less reliable results;
(ii) the tail of the distribution function acts a source for higher deviation in the result;
(iii) the final output also comes in the form of density function, which restricts its direct use, and requires conversion to a single value that further reduces the reliability of the result;
(iv) evaluation of fault tree for density function is tedious.

In the present context, fuzzy probability space concept means the probability of an event is expressed in terms of a fuzzy set [20,28,39,40]. Among various forms of fuzzy probability set representations, we recommend the use of trapezoidal representation. For example, the probability of occurrence of an event x^i is expressed as

$$P(x^i)\Delta = (\mathrm{qil}, \mathrm{pil}, \mathrm{pir}, \mathrm{qir})$$

Such that



where $fP(x^i)$ represents fuzzy probability

Using the same procedure as discussed in the simple set theory, the probability of occurrence of top event can be expressed as

$$P^{\mathrm{Top}} = h\left[P(x^1), P(x^2), \ldots P(x^i) \ldots + P(x^n)\right]$$

$$P^{\mathrm{Top}}\Delta = (\mathrm{qtl}, \mathrm{ptl}, \mathrm{ptr}, \mathrm{qtr})$$

The probability of occurrence after eliminating element x^i can be represented as:

$$P^{\mathrm{Top1}} = h\left[(0, P(x^2), P(x^3) \ldots P(x^i), \ldots P(x^n)\right]$$

$$P^{\mathrm{Top1}}\Delta = (\mathrm{qt1l}, \mathrm{pt1l}, \mathrm{pt1r}, \mathrm{qt1r})$$

and finally the improvement factor can be calculated as:

$$P^{\mathrm{Top}} - P^{\mathrm{Top1}} = \mathrm{improvement}$$

All computations are carried out in a fuzzy probability space. The final probability of occurrence and the improvement factor is also calculated in terms of fuzzy probability set. Later the fuzzy probability is transferred to normal probability using the trapezoidal average function.

The results obtained using this concept are more reliable compared to the results obtained by other methods with the same level of uncertainty in the input data. It is mainly because the single probability values are transformed in a well-defined space and all calculations are done on the same space. Doing so, the error in the data is also distributed to wider space and computation in this space causes lesser error accumulation. Eventually, the fuzzy probability can be transformed to normal probability as desired, using average function.

Further, the improvement factor has been used to formulate an improvement index. This index gives a direct measure of the sensitivity of the top event to any preceding event. The higher the index, the more sensitive is the system to that particular event. Using the index, one can identify the base events, which need greater attention if the probability of the top event (accident) has to be reduced.

To test the applicability and effectiveness of the proposed technique, we have solved a few real-life problems. The resultant case studies are summarised below.

6. Features of the software package PROFAT-II

We have recently enhanced the capabilities of PROFAT in several ways, including (a) embedding a new algorithm for FTA, (b) faster processing, (c) more efficient error handling and (d) more visual appeal. This has been done by integrating the AS-II technique described above with advanced concepts of software engineering (event-based design, object-oriented database, etc.). The features of the resulting computer automated tool, PROFAT-II, are summarised below.

PROFAT-II has been coded in Visual C + to run under WINDOWS environment. It consists of four main modules: Information handling (IHAN), cutset minimisation and optimisation (CUMO), probability analysis (PRAN), and improvement factor analysis (IMAN).

Each module performs a specific task, and is linked with the other modules. For example, the minimum cutset analysis module uses information provided in the form of Boolean relation (fault tree relation) by the IHAN module, to generate minimum cutsets. The architecture and message flow sequence of PROFAT II is given in Fig. 3.

6.1. IHAN module

In IHAN module, all relevant basic data needed for the use in other modules is assembled. Two submodules are associated with this task: (i) general information (GI), and (ii) specific information (SI). The information handled by the GI submodule serves all other modules. The information in SI submodule serves one or another specific module. Based on the information provided by the user to this module, a Boolean matrix is developed and is subsequently available to other modules. This module also deals with handling of data files, output files, and general flow of information. In a word, this module serves as an ‘‘information manager’’ that provides the necessary information to each module and submodule to carry out desired operations, and stores the results in

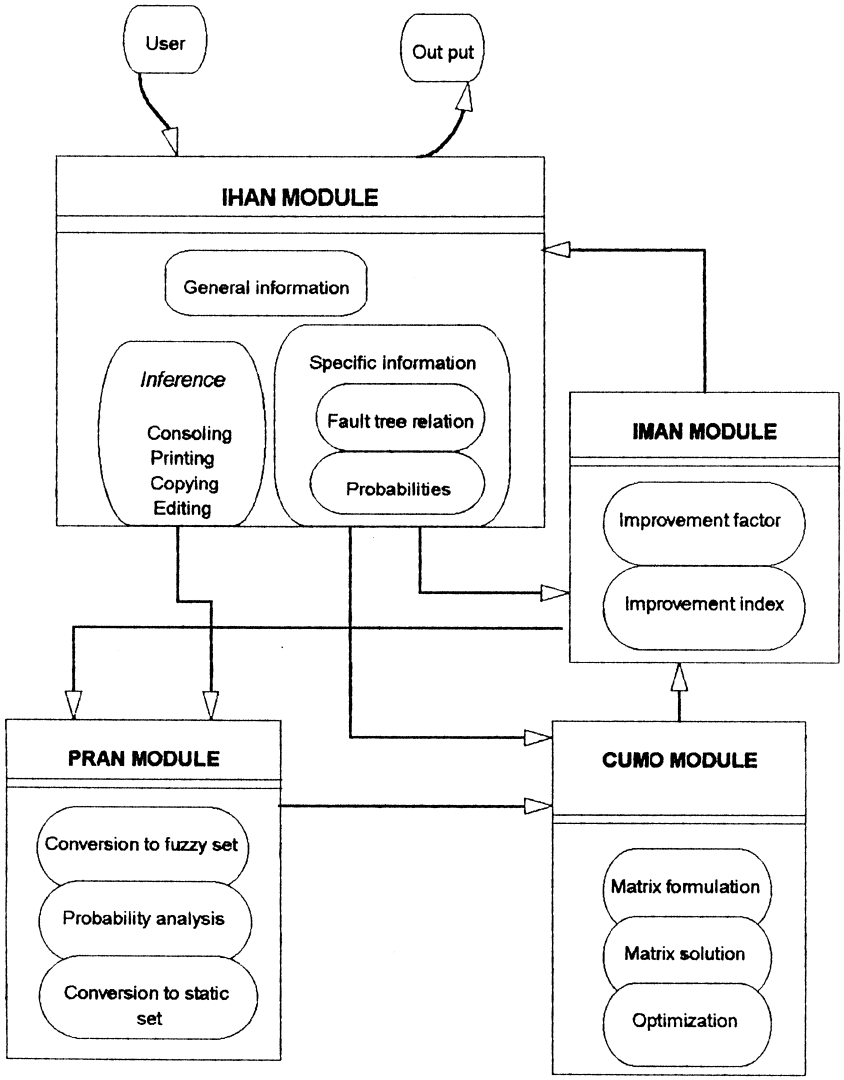


Fig. 3. Architecture and message flow sequence of PROFAT-II.

different files. It also serves up all commonly used file operations such as copying, deleting, consoling and printing.

6.2. CUMO module

CUMO consists of three fragments: (i) matrix formulation submodule, (ii) matrix solution submodule, and (iii) cutset optimisation submodule. The first submodule transforms the fault tree into a Boolean matrix, the second submodule solves the

Boolean matrix for minimum cutsets, and third submodule optimises the minimum cutsets.

Boolean matrix being the starting point of FTA, it deserves utmost care because the efficacy of further operations is strongly influenced by the matrix. The matrix is solved for minimum cutsets, using top-to-down algorithm. The cutsets are then passed on to the optimisation submodule. The resulting cutsets represent paths, which directly cause the top event. The optimisation submodule is embedded with modified Fibonaci technique. It has been given an optimisation criterion and has the facility to modify the criteria if the user so desires.

6.3. PRAN module

The probability analysis of the already developed and solved fault tree is done in PRAN. It uses optimum cutsets, the probability of occurrence of each basic event to estimate the probability of each path and, finally, the probability of the top event. PRAN consists of three submodules: (i) fuzzy set conversion submodule, which transforms the static probabilities to fuzzy probability sets; (ii) probability analysis submodule: it estimates the probability of each path and the probability of top event; and (iii) static submodule: it transforms the fuzzy probability to static probability.

6.4. IMAN module

IMAN conducts analysis of fault tree for each base event, assuming that only this base event shall not occur whereas other base events shall occur. Subsequently, It compares this probability of failure with the probability of failure when all base events were to function properly. Later, the improvement factors are computed on the basis of the difference. Finally, the improvement factors are transformed into the ''improvement index''. This index represents direct contribution (in percentage) of an event in leading the system to the undesired condition (top event to occur). The higher the improvement index, the more potent the initiating event vis a vis its role in the eventual accident.

6.5. Hardware and software requirements of PROFAT-II

PROFAT-II is a PC-based system. It needs about 16 MB RAM and ~100 MB ROM. It is operable in WINDOWS environments. It does not need any other hardware or software to input the data or to prepare final reports based on the output. The output is so formatted that it can be directly used in filling reports. Further, PROFAT-II is easy to operate and has on-line help available for each step. Even those not well versed with the AS-II techniques employed in PROFAT-II can gainfully use the software.

7. Case study 1

This case study pertains to the nitration unit of a Hexagon industry (SH process). The unit was identified for detailed FTA after all the units where screened using indices and the nitration unit was found to be potentially most hazardous [1].

7.1. Process summary of the nitration unit

The unit handles the reactant nitric acid and hexamine in 8:1 molar ratio, with the ideal temperature being 10°C. Any positive deviation in temperature or reactant proportion may cause a runaway reaction. The reactor is cooled by passing a mixture of water and methanol at a temperature of 5°C through the cooling coil. The coolant flow rate is controlled by pneumatic valve in order to maintain a reaction temperature of around 10°C. A slow moving stirrer is provided in the reactor to avoid local heating and hot spot formation. In case of an emergency, the contents of the reactor may be discharged to an emergency tank. The discharge from the reactor is activated either by pulling the electric chain, by using automatic button, or by opening manually operated manhole valve. The simplified process flow diagram of the unit is shown in Fig. 4.

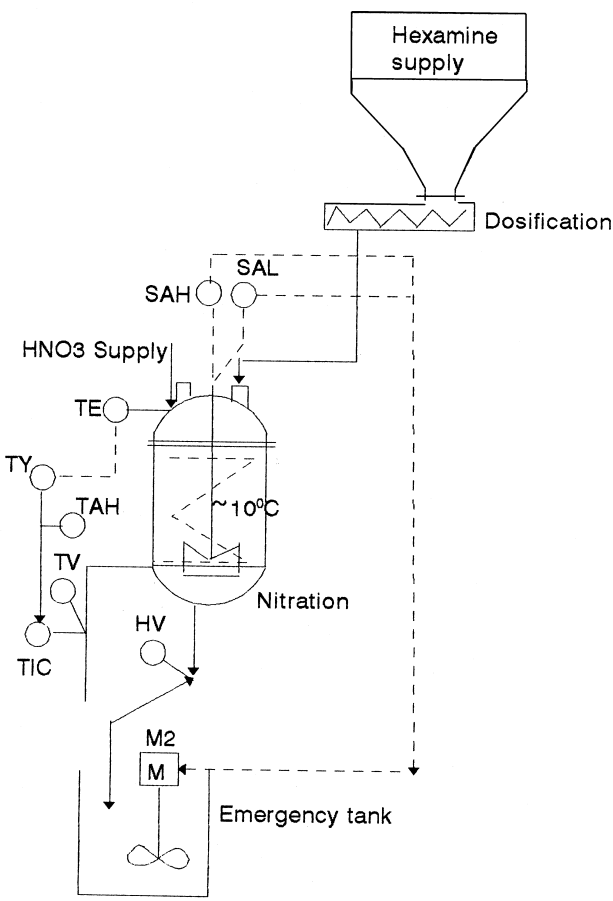


Fig. 4. Piping and instrumentation diagram.

A detailed study of the unit reveals that to control an explosion in the reactor the following precautions are necessary:

1. The reactant proportion must be controlled; especially, the proportion of HNO_3 must not be allowed to fall below eight times the proportion of hexamine.
2. The temperature in the unit must be maintained close to 10°C.
3. Local heating must to be avoided.
4. Proper working of the emergency system must be ensured.

7.2. Fault tree development and analysis

Detailed study of the process and safety measures yielded 29 basic events, which have direct and indirect dependency on the top event, explosion in the reactor. These

Table 1
Basic events and their probabilities of occurrence [38]

Basic events	Event number	Probability (failure rate/year)
Temperature alarm fails	1	0.0219
Operator ignores sounding of alarm	2	0.0001
Failure of control valve (TV)	3	0.1830
Failure of temperature controller	4	0.1270
Operator fails to open bypass on sounding alarm	5	0.0003
Bypass valve gets struck	6	0.0182
Coolant supply is inadequate	7	0.0520
Temperature indicator of HNO_3 fails	8	0.0219
No attention paid on stirrer alarm (SAH)	9	0.0001
Temperature sensor fails (TE)	10	0.1438
Transducer fails (TY)	11	0.0657
Ratio control fails	12	0.0045
Not enough HNO_3 available	13	0.1850
Stirrer shaft fails	14	0.0007
Alarm fails (SAL)	15	0.0087
Hydraulic stirrer motor fails	16	0.1277
Hydraulic supply fails	17	0.1825
Coolant ingress into the reactor	18	0.0087
Stirrer motor M2 does not start on demand	19	0.3060
On requirement hexamine supply fails to cut	20	0.0550
Solenoid valve (SV) does not work	21	1.2775
Discharge valve (HV) gets stuck	22	0.0127
Temperature switch fails (TSH)	23	0.1275
Resistance thermometer fails (TE)	24	0.0365
Sensing of higher temperature fails	25	0.1277
Operator fails to activate manual discharge	26	0.0255
Manually discharge valve gets stuck	27	0.0214
Operator fails to activate automatic discharge	28	0.0128
Automatic discharge valve gets stuck	29	0.0255

basic events include HNO_3 concentration going below permissible value, coolant leaks into the reactor, ratio control fails, transmission error, thermostats malfunction, trans-

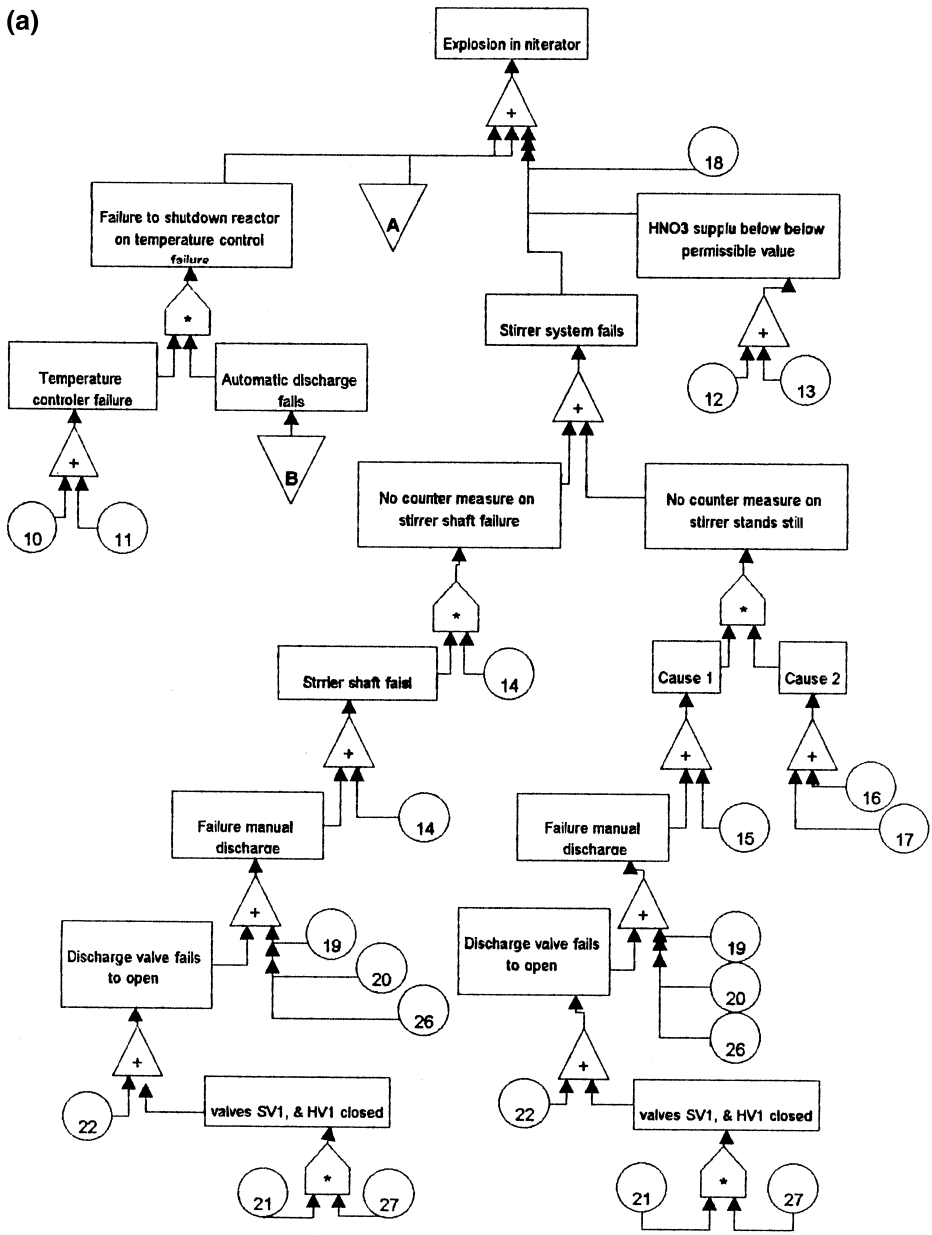


Fig. 5. Fault tree for explosion in nitration unit.

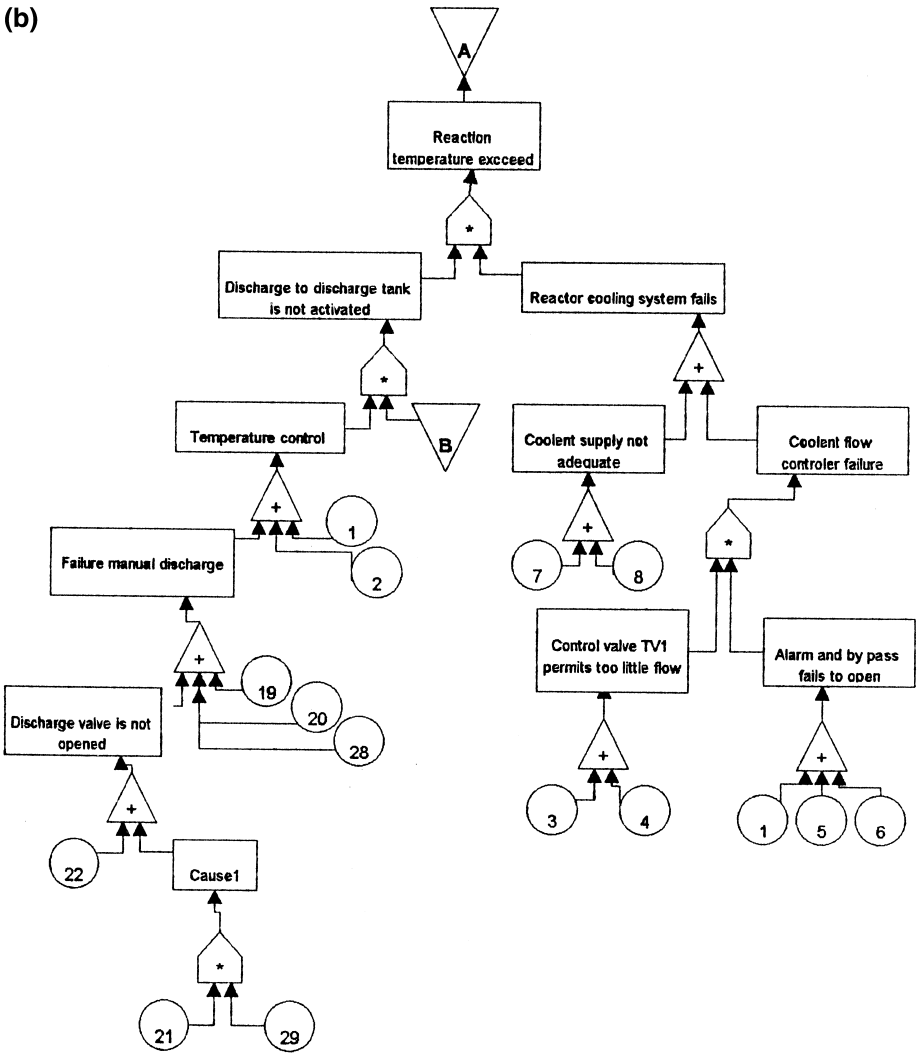


Fig. 5 (*continued*).

ducer gives erroneous signals, etc. A list of basic events with their probability of failure is given in Table 1. The probability data has been adapted from Hauptamanns [38]. The complete fault tree for the unit is shown in Fig. 5.

The fault tree was analysed using AS-II technique detailed earlier in this paper. A value of $1 * 10^{-6}$/year was considered as a limiting condition to optimise the minimal cutsets. We obtained 163 optimum minimal cutsets. They were than analysed using the fuzzy probability concept to estimate the improvement index.

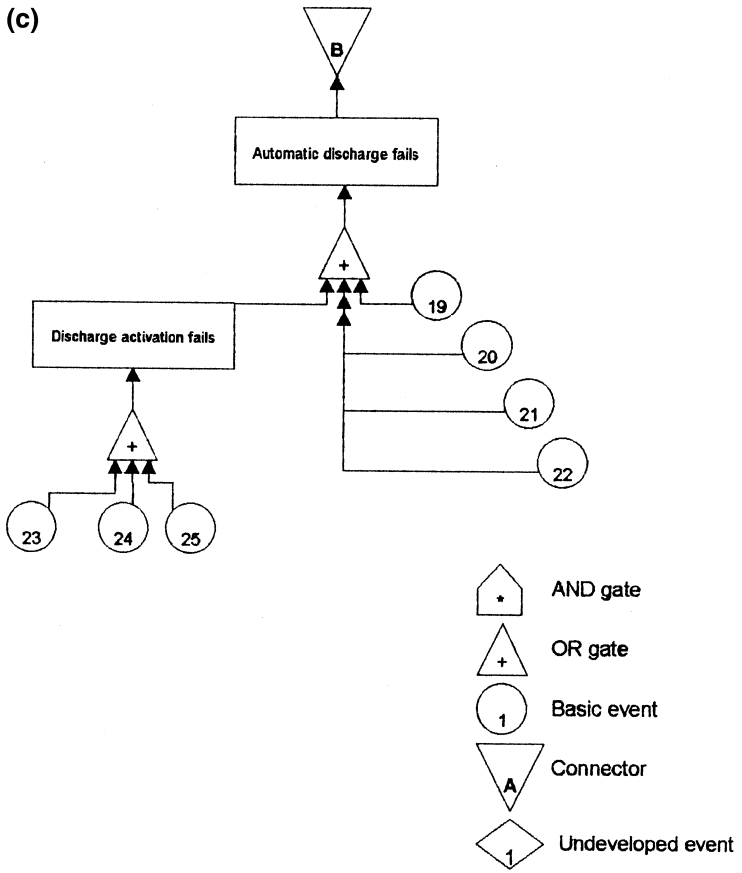


Fig. 5 (*continued*).

Table 2 presents the improvement index which reveals that HNO_3 supply below permissible limit has the highest value of the improvement index, whereas the error of reading low temperature was the lowest improvement value of index.

The study reveals (Table 2) that control of the following shall reduce the total probability by 75%:

(i) Failure of valve TV1.
(ii) Failure of temperature controller TIC1.
(iii) Ratio controller fails.
(iv) Not enough HNO_3 available.
(v) Motor M2 does not start.
(vi) When necessary, the hexamine supply is not cut off.
(vii) Solenoid valve SV1 fails.
(viii) Discharge valve HV2 gets stuck.

8. Case study 2

We have studied here the fault tree for the pressure tank system described by Vesely et al. [34], as shown in Fig. 6. The basic events and their probability of occurrence to cause pressure vessel rupture are presented in Table 3. The function of the control system is to regulate the operation of the pump. The pressure switch contacts are closed when the tank is empty. When the threshold pressure has been reached in the tank, the pressure switch opens up, cutting off power from the pump, and causing the pump motor to cease operation. The tank is fitted with an outlet valve that quickly drains off the entire tank; the outlet valve, however, is not a pressure relief valve. When the tank is empty, the pressure switch contacts close down, and the cycle is repeated.

Table 2
Probability of occurrence of the top event, and the improvement index for various basic events

Event number	Fuzzy probability × 1000 of occurrence of top event (q^a, p^a, p^b, q^b)	Improvement index
–	(5.18734e + 01, 7.26557e + 01, 1.05537e + 02, 1.18232e + 02	
1	(5.14903e + 01, 7.21792e + 01, 1.05026e + 02, 1.17758e + 02	0.750
2	(5.18715e + 01, 7.26449e + 01, 1.05521e + 02, 1.18221e + 02	0.002
3	(4.85569e + 01, 6.84708e + 01, 1.00790e + 02, 1.13641e + 02	6.964
4	(4.95714e + 01, 6.97484e + 01, 1.02238e + 02, 1.15043e + 02	4.833
5	(5.18672e + 01, 7.26397e + 01, 1.05516e + 02, 1.18217e + 02	0.010
6	(5.17811e + 01, 7.25317e + 01, 1.05394e + 02, 1.18098e + 02	0.190
7	(5.14242e + 01, 7.20850e + 01, 1.04897e + 02, 1.17624e + 02	0.925
8	(5.16755e + 01, 7.23993e + 01, 1.05247e + 02, 1.17959e + 02	0.407
9	(5.18722e + 01, 7.26457e + 01, 1.05522e + 02, 1.18222e + 02	0.000
10	(5.10460e + 01, 7.15839e + 01, 1.04269e + 02, 1.16985e + 02	1.815
11	(5.17727e + 01, 7.25176e + 01, 1.05370e + 02, 1.18072e + 02	0.220
12	(4.88162e + 01, 6.88740e + 01, 1.01434e + 02, 1.14364e + 02	6.123
13	(4.06471e + 01, 5.82504e + 01, 8.85330e + 01, 1.01408e + 02	24.62
14	(5.18712e + 01, 7.26444e + 01, 1.05521e + 02, 1.18221e + 02	0.002
15	(5.16069e + 01, 7.22951e + 01, 1.05077e + 02, 1.17761e + 02	0.631
16	(5.18168e + 01, 7.25686e + 01, 1.05413e + 02, 1.18102e + 02	0.150
17	(5.16673e + 01, 7.23833e + 01, 1.05214e + 02, 1.17918e + 02	0.448
18	(5.13883e + 01, 7.20554e + 01, 1.04898e + 02, 1.17640e + 02	0.945
19	(4.64013e + 01, 6.54634e + 01, 9.65086e + 01, 1.08928e + 02	12.83
20	(4.18825e + 01, 5.94136e + 01, 8.85752e + 01, 1.00543e + 02	23.97
21	(4.85941e + 01, 6.84451e + 01, 1.00526e + 02, 1.13222e + 02	7.243
22	(4.93394e + 01, 6.93507e + 01, 1.01466e + 02, 1.14082e + 02	5.812
23	(5.18722e + 01, 7.26457e + 01, 1.05522e + 02, 1.18222e + 02	0.000
24	(5.18722e + 01, 7.26457e + 01, 1.05522e + 02, 1.18222e + 02	0.000
25	(5.18722e + 01, 7.26457e + 01, 1.05522e + 02, 1.18222e + 02	0.000
26	(5.18354e + 01, 7.25992e + 01, 1.05469e + 02, 1.18170e + 02	0.078
27	(5.18628e + 01, 7.26337e + 01, 1.05508e + 02, 1.18208e + 02	0.022
28	(5.16164e + 01, 7.23252e + 01, 1.05162e + 02, 1.17875e + 02	0.532
29	(5.16457e + 01, 7.23685e + 01, 1.05225e + 02, 1.17941e + 02	0.449

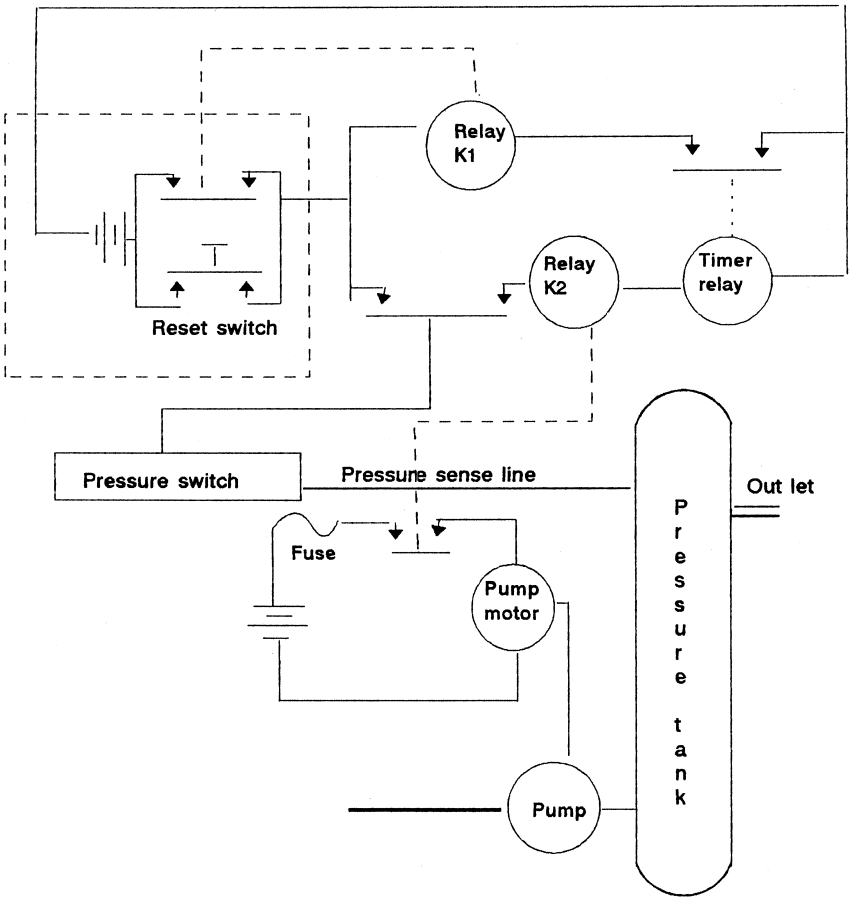


Fig. 6. Pressure tank system; system diagram [34].

The top event considered in this illustration is the rupture of the pressure tank. The fault tree developed for this event is shown in Fig. 7.

Table 3
Basic events and their probabilities of occurrence [4]

Basic events	Event number	Probability (failure rate/year)
Time relay fails to function on demand	1	5.0e−04
Time relay closed	2	1.0e−04
K1 relay fails to contact	3	3.0e−05
Reset switch fails	4	3.0e−05
Pressure switch fails to perform on demand	5	4.0e−04
Pressure switch fails	6	1.0e−04
K2 relay fails contact	7	3.0e−05
Mechanical failure of the tank due fabrication fault	8	7.5e−04

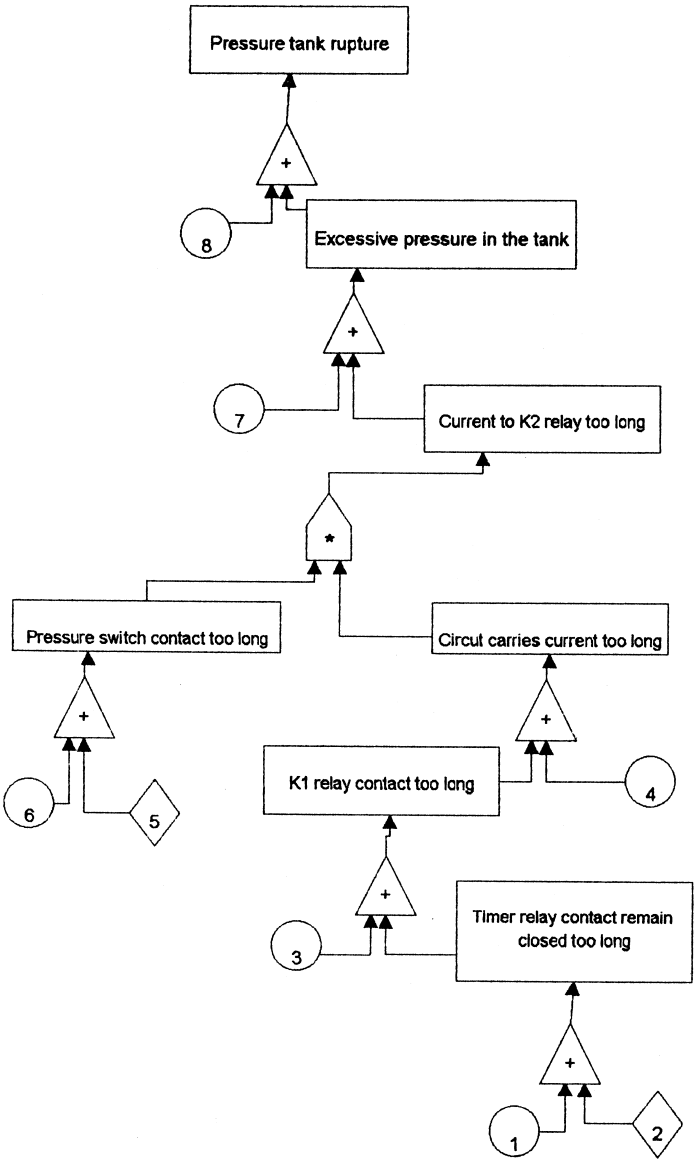


Fig. 7. Fault tree for pressure tank system [4,34].

Our analysis of the fault tree revealed that the following events would contribute to about 75% probability of the top event (accident):

(i) Failure of the timer relay (event 1 in Fig. 7).
(ii) Failure of the pressure switch on demand (event 5 in Fig. 7).

(iii) Failure of the pressure switch (event 6 in Fig. 7).
(iv) Failure of the K_2 relay contact (event 7 in Fig. 7).

9. Discussion

The performance of AS-II has been compared with other methods for case studies 1 and 2. The main parameters studied are duration of analysis (computation time) and uncertainty in the results. The uncertainty in the results have been computed using difference in the results obtained using accurate and altered (by a known amount) input data.

Evaluation of the first case study (considered to be a moderately complex fault tree) using the conventional analytical method took 46% more time than the AS-II technique we discussed, while the solution of the same problem by Monte-Carlo method took about 52% extra time. Monte-Carlo results deviate more than 11% from the single exact value, while for the same uncertainty in input data, analytic method gives 21% uncertainty in the results, and the AS-II technique only 7%.

Evaluation of the second case study (a simpler problem than the first case study) using Monte-Carlo simulation technique took only 15% more time than AS-II. In this case study, analytical method took 7% less time than AS-II technique; but for a 5% deviation (uncertainty) in the input data, the deviation in the results is maximum for analytical method ($\sim 10\%$) and least in the AS-II technique ($\sim 5.5\%$).

We have conducted a study of a more complex system consisting of 55 basic events (but not discussed in the present paper due to limitation of space). We have used modularization concepts to evaluate the fault tree by all the three methods discussed above (Monte-Carlo, analytical method, AS-II). The study using Monte-Carlo simulation took more than 2145 iterations, and required massive computational time. Moreover, the results obtained are very sensitive towards accuracy of input data; even a slight change in the data causes a large deviation in the result. Solution of this problem using analytical method generated even bigger problem of memory requirement. The Boolean matrix dimension goes up to $177 * 212$, which on subsequent processing goes beyond the allocated memory of a PCAT 586 (16 MB RAM, running under WINDOWS environment). Finally, we got a total of 1245 minimal cutsets. The results obtained by this method are also more sensitive to deviations in input than the ones obtained by Monte-Carlo simulation technique.

On the other hand, application of AS-II gave only 445 optimal minimal cutsets. Using the same computational machine, it took 53% less time than Monte-Carlo, and 48% less time than analytical method. Reliability of the results is also comparatively better. To have swifter visualisation of the comparison, we have plotted various parameters for different complexity of the problem. On plotting the duration (time required) of the fault tree evaluation (without considering the duration of fault tree construction) by various methods (Fig. 8), it has been observed that as complexity increases, the duration of study by Monte-Carlo method rises very sharply, while this rising trend is flatter for AS-II method. Similar trend has been observed for memory requirement; it is the largest for

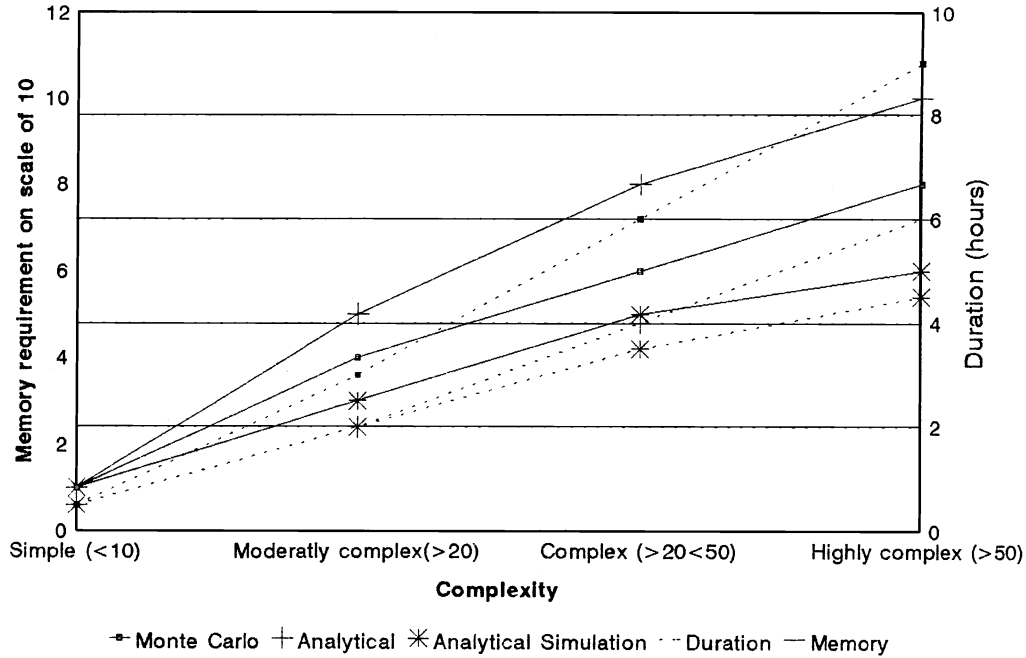


Fig. 8. Comparison of memory and time required to carry out FTA by different algorithms and for different levels of complexity of the problems; (complexity is characterised by number of gates, shown in brackets).

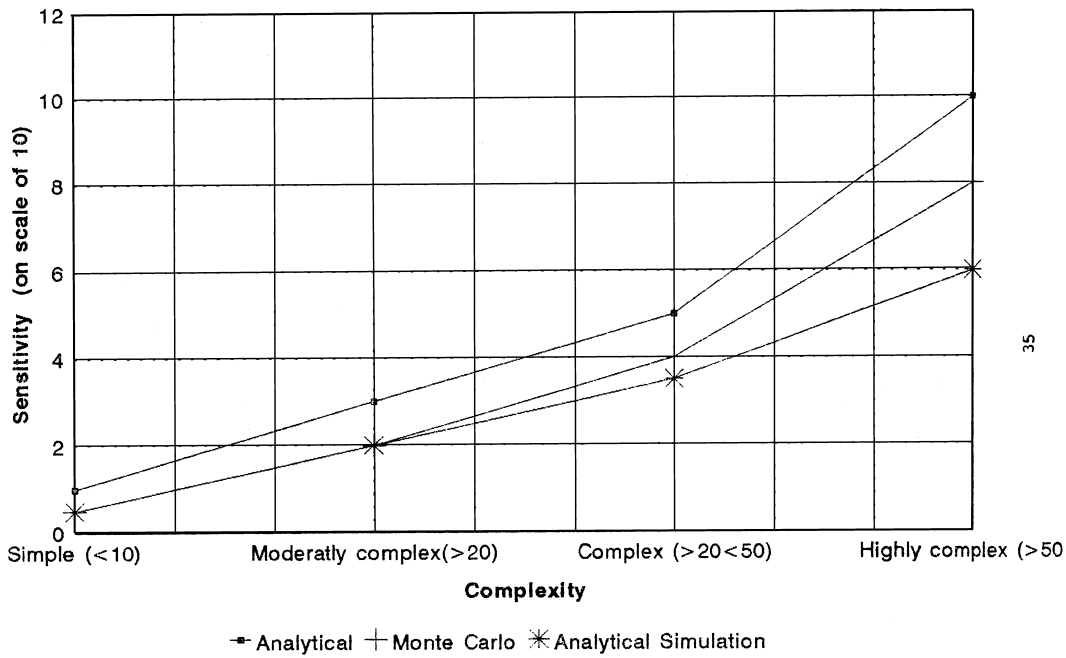


Fig. 9. Comparison of sensitivity of the results towards errors in the input data by different algorithm of FTA (complexity is characterised by number of gates, shown in brackets).

analytical method and the least for AS-II method (Fig. 8). It is because AS-II method optimises the number of minimal cutsets (on-line optimisation) at the outset, and thus reduces the computational load and memory requirement.

A plot for sensitiveness of the result to errors in input data also reflects similar trends (Fig. 9). Results obtained by analytical method are more sensitive while the same obtained by AS-II method are relatively robust against the deviations (uncertainty) in the input data.

10. Summary and conclusions

A new methodology, AS-II technique, has been described for conducting FTA. The methodology has been developed with special reference to application in assessing risks in chemical process industries. The methodology incorporates a structure modularization concept to handle complexity and largeness of the fault tree, while fuzzy space concepts have been used to dilute the impact of uncertainty in the input reliability data over final results. Two case studies are described in which the performance of the proposed technique has been evaluated. It is seen that AS-II technique performs better than Monte-Carlo simulation and analytical methods in the following terms:

(a) it requires lesser computational memory spaces, and lesser processing time,
(b) it is more robust vis a vis errors in the input reliability data.

A computer automated tool PROFAT-II has been developed on the basis of the new methodology; its features are also described in the paper.

Acknowledgements

The authors wish to thank the All India Council for Technical Education, New Delhi, for instituting the CAEM unit which made this study possible.

References

[1] F.I. Khan, S.A. Abbasi, Techniques and methodologies for risk analysis in chemical process industries, Journal of Loss Prevention in Process Industries 11 (2) (1998a) 261–273.
[2] F.I. Khan, S.A. Abbasi, Risk Assessment in Chemical Process Industries: Advanced Techniques, Discovery Publishing House, New Delhi, 1998b, ix + 365 pp.
[3] F.I. Khan, S.A. Abbasi, PROFAT: a user-friendly system for probabilistic fault tree analysis, Process Safety Progress 18 (1) (1999) 42–49.
[4] F.P. Lees, Loss Prevention in Chemical Process Industries, Butterworths, London, 1996.
[5] F.I. Khan, S.A. Abbasi, Risk analysis: a systematic scheme for hazard assessment, Journal of Industrial Pollution Control 11 (2) (1995) 89–98.
[6] CCPS, Guidelines for technical management of chemical process safety, Centre for chemical process safety, AIChE office, New York. 1989.

[7] J.S. Arendt, Using quantitative risk assessment in the CPI, Reliability Engineering and System Safety 29 (1990) 133–149.
[8] G.R. Van Sciever, Quantitative risk analysis in the chemical process industries, Reliability Engineering and System Safety 29 (1990) 55–68.
[9] P. Guymer, G.D. Kaiser, T.C. Mckelvey, Probabilistic risk assessment in the CPI, Chemical Engineering Progress (1987) 37–45, January.
[10] S.A. Lapp, A risk evaluation system, Chem Tech 14 (1991) 700–704.
[11] IChemE, Risk analysis in the process industries, EFCE publication series No. 45, IChemE office, Rugby, England, 1988.
[12] I.A. Papazoglou, A.O. Nivoliantiou, M. Christou, Probabilistic safety analysis in chemical installation, Journal of Loss Prevention in Process Industries 5 (3) (1992) 181–191.
[13] E.S. Beckjord, M.A. Cunningham, J.A. Murphy, Probabilistic safety assessment development in the United States 1972–1990, Reliability Engineering and System Safety 39 (1993) 159–170.
[14] S.A. Lapp, G.J. Powers, Chemical Engineering Progress 72 (4) (1976) 89–94.
[15] S.A. Lapp, G.J. Powers, IEEE Transactions on Reliability R-28 (1979) 12–19.
[16] G.J. Powers, F.C. Tompkins, Fault tree synthesis for chemical process, AIChE Journal 20 (1974) 376–387.
[17] A. Shafaghi, Structure modelling of process systems for risk and reliability analysis, in: Kandel, Avni (Eds.), Engineering Risk and Hazard Assessment Vol. 2 CRC Press, Florida, 1988, pp. 45–64.
[18] U. Hauptmanns, J. Yllera, Fault-tree evaluation by Monte-Carlo simulation, Chemical Engineer (1983) 1991–1995, January.
[19] L. Camarinpoulous, J. Yllera, An improved top-down algorithm combined with modularization as a highly efficient method for fault tree analysis, Reliability Engineering 11 (2) (1985) 93–102.
[20] F.S. Lai, S. Shenoi, L.T. Fan, Fuzzy fault tree analysis theory and applications, in: Kandel, Avni (Eds.), Engineering Risk and Hazard Assessment Vol. 1 CRC Press, Florida, 1988, pp. 139–167.
[21] K. Thangamani, Reliability of FCC unit using Monte-Carlo and direct simulation technique, Reliability Engineering 68 (1992) 125–136.
[22] A. Bossche, Computer aided fault tree synthesis; system modelling and causal trees; fault tree construction; real time fault location-I, Reliability Engineering and System Safety 32 (1991) 217–241.
[23] A. Rauzy, New algorithms for fault tree analysis, Reliability Engineering and System Safety 40 (1993) 203–211.
[24] H.R. Greenberg, B.B. Slater, Fault Tree and Event Tree Analysis, Van Nostrand Reinhold, New York, 1991.
[25] E.J. Henevely, M. Kumanoto, Reliability Engineering and Risk Assessment, Prentice-Hall, Englewood Cliffs, NJ, 1981.
[26] E.J. Henley, M. Kumanoto, M.J. Cliffs, Designing for Reliability and Safety Control, Prentice-Hall, Englewood Cliffs, NJ, 1985.
[27] J.M. Taylor, An algorithm for fault tree construction, IEEE Transactions on Reliability R-31 (1982) 137–146.
[28] R.B. Worrel, D.W. Stack, A SETS user's manual for the fault tree analyst, SAND77-2051, Sandia Natl. Lab., Albuquerque, NM, 1990.
[29] U. Hauptmanns, W. Werner, Engineering Risks, Springer-Verlag, Berlin, 1990.
[30] G.A. Moetin-Solis, P.K. Ardow, F.P. Lees, Fault tree synthesis for design and real time applications, Transactions of the Institute of Chemical Engineering 60 (1) (1982) 14–25.
[31] AIChE. Guidelines for process equipment reliability data with data tables, AIChE office, New York. 1989.
[32] N. Ulerich, G.J. Powers, On-line hazard aversion and fault-diagnosis in chemical processes: the digraph + fault-tree method, IEEE Transactions on Reliability 37 (2) (1988) 171–178.
[33] S.A. Lapp, G.J. Powers, Computer aided synthesis of fault trees, IEEE Transactions on Reliability R-26 (1977) 2–21.
[34] W.E. Vesely, F.F. Goldberg, N.H. Roberts, D.F. Haas, Fault Tree Handbook, US Nuclear Regulatory Commission, Washington, DC, 1981.
[35] K. Noma, H. Tankara, K. Asai, Fault tree analysis with fuzzy probability, Journal of Ergonomics 17 (1981) 291–297.

[36] A. Kandel, E. Avni, Engineering Risk and Hazard Assessment Vols. 1 and 2 CRC Press, Florida, 1988.
[37] J. Yllera, Modularization methods for evaluating fault tree of complex technical system, in: Kandel, Avni (Eds.), Engineering Risk and Hazard Assessment Vol. 2 CRC Press, Florida, 1988, pp. 81–100.
[38] U. Hauptmanns, Fault tree analysis for process industries: engineering risk and hazard assessment, in: Kandel, Avani (Eds.), Engineering Risk and Hazard Assessment Vol. 1 CRC Press, Florida, 1988, pp. 21–61.
[39] D. Dubois, H. Prade, Fuzzy Sets and Systems: Theory and Applications, Academic Press, New York, 1980.
[40] H. Tanaka, L.T. Fan, F.S. Lai, K. Toguchi, Fault tree analysis by Fuzzy probability, IEEE Transactions on Reliability R-32 (1983) 453–456.