FUZZY LOGIC-BASED FAULT DIAGNOSIS FOR MINING EQUIPMENT FAILURES

by

TAPAS RANJAN KAR

THESIS submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

MINING ENGINEERING

APPROVED:

DR. E. TOPUZ, CHAIRMAN

DR. J. R. LUCAS

DR. G. ADEL

December 11, 1989

Blacksburg, Virginia

FUZZY LOGIC-BASED FAULT DIAGNOSIS FOR MINING EQUIPMENT FAILURES

by

TAPAS RANJAN KAR

DR. E. TOPUZ, CHAIRMAN

DEPARTMENT OF MINING & MINERALS ENGINEERING

(ABSTRACT)

Equipment availability is the most significant factor in the productivity of many mines and processing plants. Machine breakdowns are not only expensive in terms of production losses but also important in meeting production schedules. In a complex piece of machinery like a shearer or a powered support system in a highly automated longwall face, such breakdowns can be due to one of the large number of possible faults. A large proportion, up to 80%, of the down time is spent in locating the fault. For this reason, a need for an automated diagnostic method to assist the operator in the diagnosis process is felt. In this study, a diagnostic system is developed by modeling the partially known or imprecise relations and poorly defined variables found in a diagnostic environment. Logic of fuzzy sets and systems theory finds an interesting application in this area. This study presents a diagnostic algorithm, which relates the possible causes of failure to their respective symptoms through fuzzy logic paths. Applications of the diagnostic method are illustrated through examples of a compressor and a shearer.

Acknowledgements

This thesis is dedicated to my parents whose constant support and encouragement have led to the completion of this degree. They taught me to strive for excellence and to reach ever higher goals.

I would like to express my sincere gratitude to Dr. Topuz for his continual guidance, constructive criticism, encouragement and not to mention financial support throughout the preparation of this research. Thanks are also due to Dr. Adel and Dr. Lucas for being in the committee and the valuable suggestions.

I would also like to express my deep appreciation to all my friends. Without their help and companionship it would have been a lot harder.

Table of Contents

List of Tables

Chapter 1

INTRODUCTION

With the increasing trend towards automation in different areas of mining, it has become necessary that the downtime of any system be reduced to minimum. In a complex piece of machinery, the exact diagnosis of the cause of a failure or early detection of an abnormal state is necessary in order to implement corrective action and to meet the increased safety requirements of modern mines. If the mode of equipment failure could be known and all remedial actions could be considered, an automated safety and a fault diagnostic system might be constructed. However, the complexity of modern equipment has rendered understanding of its functional deviations more difficult. In troubleshooting for such a system, human operators' perception such as smell, noise or vibration are very important. Such sensory measures, as well as signals from the mechanical sensors, may be available for the early detection of an abnormal state. Therefore, a diagnostic system must be a man-machine system, in which the operator gives sensory perceptions to the computer and is also responsible for the final judgement, while the computer processes the signal from the mechanical sensors together with the perception of the human operator.

The conventional quantitative techniques of system analysis are intrinsically unsuitable for dealing with humanistic systems or, for that matter, any system whose complexity is comparable to that of humanistic systems. The basis for this contention rests on what might be called the principle of incompatibility : "Stated informally, the essence of this principle is that as the complexity of the system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics" (Zadeh, 1973, pp. 28). Thus, with the growing complexity of the modern equipment used in mining, the diagnosis of such a system at an early stage has become extremely difficult in the presence of a huge mass of inexact operational data. The methods for obtaining a precise solution in this type of situation are too complex. A less precise but still acceptable solution can be achieved by developing a diagnostic algorithm which is essentially fuzzy in nature. It is understood that safety and availability of equipment are not mutually exclusive objectives, and one might expect a satisfactory diagnostic technique to benefit both.

Diagnosis is one of the many areas to which the mathematics of fuzzy set theory has been applied successfully. This is due to the fact that in cases of equipment trouble, human perception and decision, although imperfect, play important roles in the diagnosis of the fault. Because of the subjective and fuzzy nature of the information used, fuzzy set theory has found a useful application in this area. Fuzzy set theory as a tool for working with generalities and imprecise data was first introduced by Lofty A. Zadeh. As opposed to ordinary set theory, a fuzzy set is "a class of objects with a continuum of grades of membership," (Zadeh, 1965). These are the classes of objects which do not have precisely defined criteria. An example for this can be "a set of integer numbers near twenty," i. e., a set with such integer numbers as 18, 19, 20, 21, 22 etc. depending on the definition of nearness. Alternatively, the set of integer numbers approximately equal to fifty may comprise numbers like 48, 49, 51 and 52, which are defined to be approximately equal to fifty. Fuzzy membership allows one to express mathematically this type of less-than-precise statement. The word fuzzy is synonymous with inexact, vague, or imprecise, and in this context one may refer to "degree of fuzziness." Establishing a membership or characteristic function to represent a fuzzy expression is a subjective task.

Since its inception in the early sixties, the theory of fuzzy sets has evolved in a wide variety of fields. The concept of a linguistic variable and its application in fuzzy logic or approximate reasoning, as introduced by Zadeh (1975), evolved into an important field in the early seventies. The approximate reasoning was found to simulate human reasoning more closely than classical binary logic and began to be applied to philosophy, psychology, management science, decision analysis, and diagnosis.

In addition to Zadeh, Kaufmann (4), Ragade and Gupta (9), King and Mamdani (5), and several other authors have made a significant contributions to the field of fuzzy set theory and its application in various areas. At the beginning, control systems represented the area in which fuzzy logic found most of its applications. An overview of fuzzy control systems has been written by Mamdani (7). As a logical extension of control systems, diagnosis has been another area of interest. In this area the first work was probably done by Sanchez, who presented a paper on the use of inverse relationships for medical diagnosis at an IEEE conference in 1977 (11). After this Tsukamoto and Terano came out with new findings which established upper and lower bounds on the conditioned inverse fuzzy relation (15). This problem was further pursued by Shahinpoor and Wells (12).

A significant amount of work in the area of fuzzy logic and its application has been done in the eighties. Most of this work has originated in Japan. In the early eighties several Japanese firms plunged enthusiastically into the area of fuzzy research. By 1985, a computer based on fuzzy logic was developed. The fuzzy concept has been successfully applied in different areas, such as controls, forecasting, diagnosis, etc. It is believed that approximate reasoning and fuzzy logic have much to contribute in the area of diagnosis in the coming years.

In this study, the mathematics of fuzzy sets and systems relevant to the diagnosis environment is reviewed first. A diagnostic method is developed and its possible applications in trouble-shooting mining equipment are suggested. Next, development of the diagnostic algorithm is discussed in detail. A complete flow-chart of the algorithm is given for ease in understanding the procedure. A working computer code is also provided, along with the solution of the example problems. Several examples of diagnostic problems which may be found in mining are discussed in order to illustrate the usefulness of the algorithm.

Chapter 2

MATHEMATICS OF FUZZY SETS AND SYSTEMS

2.1 Fuzzy Set Theory and Fuzzy Operations

Fuzzy Set Theory

Imprecision in characterizing the presence or absence of properties in a system leaded to the concept of fuzzy set. A fuzzy set can be defined as a collection of numbers, objects, etc. Although, these elements belong to a clearly defined set, their memberships in the set are poorly defined or subject to interpretation. The concept of fuzzy set arose to cope with the mathematical imprecisions or generalities through the membership function. The elements of a set are said to have strong memberships if the membership function values are close to one or weak memberships if they are close to zero. This is a departure from binary logic, which only takes a value of 0 or 1. A good example of the "fuzzy concept" would be students' grading. The grade assigned to a student in a course represents the degree to which his/her teacher measures his/her performance in that subject. These grades may range from A to

F. Thus, not a binary but a "fuzzy concept" is used in judging the overall performance of the students. Other examples of fuzzy sets may be:

1. A set of beautiful girls.

2. A set of classical songs.

3. A set of odd integers near five.

Sets of girls, songs and odd integers may be well defined; but "beautiful," "classical" and "near five" are fuzzy expressions making the above sets subject to judgement. As an example, the last item might be expressed as follows: Let A be a set of odd integers near five.

$A = \{1, 3, 5, 7, \ldots, 11, \ldots\}$

The membership function values for the elements of the set may be given as:

$U_A = \{0.5, 0.8, 1.0, \ldots\ldots, 0.1, \ldots\ldots\}$

where U_A is the membership function for the elements. Thus, the degree to which the odd integer 1 is compatible with the set of odd integers near 5 is 0.5, while the compatibilities of 3 and 11 are 0.8 and 0.1 respectively. Here the fuzzy restriction, i. e., the fuzzy relation "near", which acts as an elastic constraint on the values, is assigned to the odd integers.

Now we will discuss briefly what exactly we mean by fuzzy restriction. As an illustration, consider the proposition "Soup is hot." In this case, the implied attribute is temperature. On denoting the restriction on the temperature of the soup by R (Temperature(Soup)), the proposition may be expressed as:

$$R(Temperature(Soup)) = hot$$

with hot being a subset of the interval $[0^0F, 212^0F]$ defined by, say, a membership function in the form of:

$$U_{hot}(t) = S(u^0F, 32^0F, 100^0F, 200^0F)$$

Thus, if the temperature of the soup is $u = 100^0F$, then the degree to which it is compatible with the fuzzy restriction "hot" is 0.5. This temperature is called the crossover point of "hot." The compatibility of 200^0F with "hot," however, is unity. Thus, R (Temperature(Soup)) plays the role of a fuzzy restriction on the soup temperature which is assigned the value "hot," with the compatibility function of "hot" serving to define the compatibilities of the numerical values of temperature with the fuzzy restriction "hot."

Fuzzy Operations

Consider a set X which represents a group of five automobiles. Further assume that the automobiles in this set can be grouped under two fuzzy subsets according to their efficiencies (subset A) and prices (subset B). Prices are qualified as being "inexpensive."

X = { a, b, c, d, e }

A = { (a, 0.8), (b, 0.5), (c, 0.9), (d, 0.6), (e, 1.0) }

B = { (a, 0.5), (b, 0.7), (c, 0.4), (d, 0.8), (e, 0.6) }

In the above subsets one can see that automobiles a, c and e have a greater membership values in A, where automobiles b and d have a greater membership values in B.

Fuzzy Complement

As ordinary set complement, the fuzzy complement is also defined as:

$$\overline{U}_A = 1 - U_A \tag{2.1}$$

The complement of A is stated as "not A" or in the example, "not efficient." The complements of A and B are given as:

$\overline{A} = \{(a, 0.2),(b, 0.5),(c, 0.1),(d, 0.4),(e, 0.0)\}$

$\overline{B} = \{(a, 0.5),(b, 0.3),(c, 0.6),(d, 0.2),(e, 0.4)\}$

Fuzzy Intersection

The characteristic function, U_c, for the intersection of two fuzzy sets A and B is given as:

$$U_C = U_{A \cap B} = \min(U_A, U_B) \qquad (2.2)$$

where the operation 'min' assigns the lowest values within the brackets to U_c. The intersection of fuzzy sets A and B can be expressed as:

$U_{\hat{C}} = \{0.5, 0.5, 0.4, 0.6, 0.6\}$

or the subset obtained can be given as:

$C = \{(a, 0.5),(b, 0.5),(c, 0.4),(d, 0.6),(e, 0.6)\}$

It may be said that C is included in A and B, because for all x, C is a subset of A as well as B. The intersection represents cars being both "efficient" and "inexpensive." In the example, the elements d and e have the greatest membership value ($U_c(d) = U_c(e) = 0.6$) , therefore car types d and e are equally "efficient" and "inexpénsive."

Fuzzy Union

The characteristic function for the union of the fuzzy sets A and B is given by the following expression:

$$U_C = U_{A \cup B} = \max(U_A, U_B) \qquad (2.3)$$

In this expression, the operation 'max' assigns the greatest value within the brackets to U_C. The union is stated as A and/or B, i. e., "efficient" and/or "inexpensive." For the example, the membership function values of the union can be expressed as:

$$U_C = \{0.8, 0.7, 0.9, 0.8, 1.0\}$$

or

$$C = \{(a, 0.8),(b, 0.7),(c, 0.9),(d, 0.8),(e, 1.0)\}$$

The union or intersection of a fuzzy set and its fuzzy complement has an interesting difference from those for an ordinary set. For the above example the union and the intersection of A and $\bar{A}$ are:

$$A \bigcup \bar{A} = \{(a, 0.8),(b, 0.5),(c, 0.9),(d, 0.6),(e, 1.0)\}$$

$$A \bigcap \bar{A} = \{(a, 0.2),(b, 0.5),(c, 0.1),(d, 0.4),(e, 0.0)\}$$

On the other hand, the union of an ordinary set and its complement always yield the universe set, and the intersection always results in an empty set (ϕ).

2.2 Linguistic and Fuzzy Variable

The traditional techniques of system analysis are not well suited to humanistic systems because they fail to model the fuzziness of human behavior. Thus, to deal with such systems realistically, an approach which does not require precision in its operation but instead employs a methodological framework which is tolerant of imprecision and partial truth is needed. The ability to summarize information plays an essential role in the characterization of complex phenomena. In the case of humans, the ability to summarize information finds its most pronounced manifestation in the use of natural languages. Thus, each word x in a natural language L may be viewed as a summarized description of a fuzzy subset M(x) of a universe of discourse U, with M(x) representing the meaning of x. In this sense, the language as a whole may be regarded as a system for assigning atomic or composite labels (i. e., words, phrases and sentences) to the fuzzy subsets of U. For example, if the meaning of the noun *flower* is fuzzy subset M(flower) and the meaning of the adjective *red* is fuzzy subset M(red), then the meaning of the noun phrase *red flower* is given by the intersection of M(flower) and M(red).

If the color of the an object is considered as a variable, then its values red, blue, yellow, green, etc., may be interpreted as labels of fuzzy subsets of the universe of objects. In this sense, the attribute color is a fuzzy variable, i. e., a variable whose values are labels of fuzzy sets. It is important to note that the characterization of a value for the color by a natural label such as *red* is much less precise than the numerical value of the wavelength of the particular color.

In the preceding example, the values of the variable color are red, blue, yellow, etc. More generally, the values may be sentences in a specified language, in which one can say that the variable is linguistic. To illustrate, the values of the fuzzy variable "tallness" might be expressed as tall, not tall, somewhat tall, very tall, not very tall, very very tall, tall but not very tall, quite tall, more-or-less tall, etc. Thus values in question are sentences formed from the label *tall*, the negation *not*, the connectives *and* and *but* and the hedges *very, somewhat, quite* and *more or less.* Therefore, the variable height as defined above is a linguistic variable.

The function of linguistic variables is to provide systematic means for approximate characterization of complex or ill-defined phenomena. In essence, by moving away from the use of quantified variables towards the use of the type of linguistic descriptions used by humans, one can acquire a capability for dealing with systems which are much too complex to be susceptible to analysis in conventional mathematical terms.

2.3 Characteristic Function

The key elements in thinking are not numbers but labels of fuzzy sets, i. e., classes of objects in which the transition from membership to nonmembership is gradual rather than abrupt. Indeed, the pervasiveness of fuzziness in human thought processes suggests that much of the logic behind human reasoning in not the traditional two-valued or even multivalued logic, but a logic that plays a basic role in one of the most important facets of human thinking, namely, the ability to summarize information, that is, to retrieve information from the collection of masses of data stored in the human brain those and only those subcollections which are relevant to the performance of the task in hand. Therefore, for this purpose, a very approximate characterization of a collection of data is sufficient because most of the basic tasks performed by humans do not require high degree of precision in their execution. Thus, the stream of information reaching the brain via the visual, auditory, tactile and other senses is eventually reduced to some less precise values. A characteristic function helps to summarize this stream of information through the definitions of certain fuzzy sets. The characteristic function of a particular fuzzy set must be chosen to best represent the understood relationship of the elements to the set, which is not clearly defined.

This section reflects the attempts made by Zadeh and others to develop a conceptual framework for analyzing systems which are too complex or ill-defined to perform a precise quantitative analysis.

An example of such a fuzzy set in diagnosis of a compressor could be air discharge temperatures above a normal of 160° F. Let us say that the characterization function which represents the above relationship is:

$$U(t) = 0 , \qquad 0 \leq t \leq L$$

$= 1 - e^{-K(t-L)}, \quad t > L,$

where L = normal air discharge temperature of 160°

t = actual air discharge temperature

K = 0.05

Now when the air discharge temperature is 210° F, the membership function takes a value of 0.92, while a temperature of 170° F has a value of 0.40. Similarly, a temperature of 190° F assumes a membership value equal to 0.78. These membership values are found by considering the associated membership or characteristic function.

There are infinite possibilities for selecting a mathematical expression to represent the characteristic function of a fuzzy expression. Several examples for the fuzzy expressions like "x is small" and "x is large" are given in Kaufmann (4) and Ragade and Gupta (9).

The characteristic function for a fuzzy set may be subject to various operations that modify the degree of fuzziness.

Contrasting

Contrast intensification is the artificial reduction of fuzziness to approximate ordinary sets. This intensification is done through the following operations:

INT(A) = CON(A) = $U_A^2(x)$ for all x such that $U_A(x) < 0.5$

INT(A) = DIL(A) = $U_A^{0.5}(x)$ for all x such that $U_A(x) \geq 0.5$

where $CON(A) \equiv$ *Concentration* of A and $DIL(A) \equiv$ *Dilation* of A

The use of exponent increases membership function value if it is greater than 0.5 and decreases it if less than 0.5.

Fuzzification (or blurring)

It is an arbitrary method of accounting for reduced accuracy, in which membership values in the open interval (0, 1), that is, values other than values 0 or 1, are converged to 0.5 with functions such as:

BLUR(A) = DIL(A) = $U_A^{0.5}(x)$ for all x such that $U_A(x) < 0.5$

BLUR(A) = CON(A) = $U_A^2(x)$ for all x such that $U_A(x) \geq 0.5$

This has the property of decreasing membership function values if they are greater than 0.5 and increasing them if less than 0.5. The level 0.5 is chosen arbitrarily.

In the context of diagnosis, the linguistic variables such as noisy, hot, etc., as proposed by Zadeh (16) are very helpful. Hedges like very, very very, and not very are used to modify the characteristic function to represent a modified linguistic phrase. These help in modeling human observations mathematically. The results of the diagnosis also may be better stated linguistically for operator's information., i. e., "piston ring may be worn or broken" instead of a numerical value. As an example, RPM of a motor may be considered in an expression as being "near 60 RPM." The characteristic function for this expression, then, may be chosen as:

$$A = e^{-|C(r-x)|}$$

where r is the reference speed, x is the actual speed and C = 0.05. Table 1 shows the membership function value for different values of x, together with their values modified for different linguistic expressions. The modification formula for different expressions suggested by Zadeh (17), have been used to obtain the values.

Table 1. Example for Characteristic Function Values and their Linguistic Modifications

		Membership values for x =				
Expression	Function	3	25	45	55	58
Near r	U_A	0.05	0.17	0.47	0.78	0.91
Very near r	U_A^2	0.003	0.03	0.22	0.61	0.83
Very very near r	U_A^4	0.00	0.00	0.05	0.37	0.69
Not near r	$1 - U_A$	0.95	0.83	0.53	0.22	0.09
Not very near r	$1 - U_A^2$	0.997	0.97	0.78	0.39	0.17

2.4 Fuzzy Relations

Let us consider the following fuzzy restriction:

"John and Tom are approximately equal in height."

In this proposition we have two variable to consider, namely Height(John) and Height(Tom). Thus in this case the assignment equation takes the form:

R(Height(John), Height(Tom)) = approximately equal

in which "approximately equal" is a binary fuzzy relation which may be characterized by a compatibility matrix $U_{approximately\ equal}(u,v)$ shown in Table 2. For example, in the table it is observed that if John's height is 5'10 and Tom's height is 5'8, then the degree to which they are approximately equal is 0.9.

Two different fuzzy restrictions may be related or unrelated. For example, "Kim is young" and "the color of Ted's hair is black" are two propositions in which the fuzzy propositions involved are unrelated in the sense that the restriction on Kim's age has no bearing on the color of

Table 2. Compatibility Matrix of the Fuzzy Relation Approximately Equal

u \ v	5′6	5′8	5′10	6′0	6′2	6′4
5′6	1	0.8	0.6	0.2	0	0
5′8	0.8	1	0.9	0.7	0.3	0
5′10	0.6	0.9	1	0.9	0.7	0
6′0	0.2	0.7	0.9	1	0.9	0.8
6′2	0	0.3	0.7	0.9	1	0.9
6′4	0	0	0	0.8	0.9	1

Ted′s hair. More generally, however, the restrictions may be interrelated, as in the following example:

"X_1 is small."

"X_1 and X_2 are approximately equal."

In terms of fuzzy restrictions on X_1 and X_1, X_2, the two propositions translate into the following assignment equations:

$R(X_1)$ = small

$R(X_1, X_2)$ = approximately equal

where $R(X_1)$ and $R(X_1, X_2)$ denote the restrictions on X_1 and (X_1, X_2) respectively.

From the knowledge of the fuzzy restriction on X_1 and (X_1, X_2), one can deduce a fuzzy restriction on X_2. Thus, for the example discussed above it is possible to assert that

$R(X_2) = R(X_1) \bullet R(X_1, X_2)$ = small • approximately equal

where • denotes the composition of fuzzy relations. The above relation for the given example is inferred from the compositional rule of inference.

The compositional rule of inference may be best expressed by a simple example. Assume

X_1 = { (1,1.0), (2,0.6), (3,0.2), (4,0.0) }

i. e., $U_{small}(X_1)$ = { 1.0, 0.6, 0.2, 0.0 }

and

X_1 , X_2 = [{((1,1), (2,2), (3,3), (4,4)), 1.0}, {((1,2), (2,1), (2,3), (3,2), (3,4), (4,3)), 0.5}]

In this case, the operation small•approximately equal, may be expressed as the max-min product of the relation matrices of small and approximately equal. This is denoted as $R(X_1) \bullet R(X_1,X_2)$ and defined as:

$$U_{R(X_1)\bullet R(X_1,X_2)}(X_2) = \max_{X_1}[\, \min\{U_R(X_1),\, U_R(X_1,X_2)\}] \qquad (2.4)$$

The operation "min" is similar to intersection and yields the least value found within the parenthesis. The operation "max" is the same as union and yields the greatest value for any value of X_1. Thus small • approximately equal = [1 0.6 0.2 0] •

$$\begin{bmatrix} 1 & 0.5 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0 \\ 0 & 0.5 & 1 & 0.5 \\ 0 & 0 & 0.5 & 1 \end{bmatrix}$$

= [1 0.6 0.5 1]

The fuzzy restriction on X_2 is obtained through the max-min composition, which is illustrated in Table 3. The result is given as:

C = { (1,1.0), (2,0.6), (3,0.5), (4,0.2) }

When there exists a relation between two sets A and B, and the values of set B are dependent upon the values for set A, then B is called a conditioned set. Here A may be considered as an array of causes and C as an array of effects connected through B.

Table 3. Example for Finding Max-Min Composition

Min operation	Min of the values within the parentheses				Max of min values
(1,1) (0.6,0.5) (0.2,0) (0,0)	1	0.5	0	0	1
(1,0.5) (0.6,1) (0.2,0.5) (0,0)	0.5	0.6	0.2	0	0.6
(1,0) (0.6,0.5) (0.2,1) (0,0.5)	0	0.5	0.2	0	0.5
(1,0) (0.6,0) (0.2,0.5) (0,1)	0	0	0.2	0	0.2

Chapter 3

POTENTIAL USE OF FUZZY LOGIC-BASED DIAGNOSIS IN MINING

It is only by reducing the cost of production that coal industries will be able to compete in the energy market. In the interests of their greater efficiency and continued existence, mining operations worldwide are becoming more and more automated. The philosophy behind automating the mining industry, as all other industries, is to improve overall productivity and to reduce cost of production. These goals can only be achieved through higher utilization of mining equipment. This, in turn, requires that machine run-time or availability of the system be improved. In fact, low availability of equipment is the most significant cause of low productivity in a highly automated mining operation. Therefore, it is important that the key factors inhibiting the greater realization of the potential performance of mining equipment be identified.

It is a recognized fact that a good maintenance program is a prerequisite for achieving minimum delay costs and maximum availability of equipment. Moreover, in recent years, the shift in emphasis in the mining industry towards consolidation of plant investments in larger and more productive mining sections (i. e., longwall operations), and the control of unit costs, have generated much interest in the field of equipment availability and reliability. It is important that any potential problem is identified at an early stage and the necessary corrective or preventive actions taken. Machine breakdowns are extremely expensive not only in terms of the production that is immediately lost but also in terms of disruptions to future production schedules. In a complex piece of machinery like a shearer or powered support system in a highly automated longwall face, such breakdowns can be due to one of the large number of possible faults. In the event of a failure, the cause the the failure needs to be located as quickly as possible. A microcomputer-based logical fault diagnosis technique would be of great help to an operator in detecting operational deviation of a piece of equipment at an early stage. Failure diagnosis based on the statistical analysis of available information about the system's operations and environment prior to the time of failure has been one of the classical systems engineering approaches. In engineering applications, automated failure diagnosis (for a given device) can be carried out by using a special-purpose processor with a stored, fixed program or an arithmetical unit running in the time-sharing mode. In automated statistical diagnosis, possible fault causes are recognized from observed symptoms and previous operating history. In the terminology of pattern recognition these symptoms are called pattern. Automated statistical diagnosis is suitable for complex failure situations, involving wear, multiple failures, and the consequences of external working conditions. This approach is useful when direct measurements and logical fault diagnosis prove inadequate. Logical fault finding is based on symptom analysis combined with the inspection of the equipment. The implementation of this procedure depends upon the development of a suitable fault free, which is a very difficult task as the equipment being used in mining is becoming increasingly complex. This statistical diagnosis also demands numerous preliminary equipment tests to obtain learning data about all possible equipment conditions. Zadeh's principle of incompatibility becomes relevant as the complexity increases and statistical diagnosis becomes time consuming.

Besides logical and statistical diagnosis there is a third approach to the problem. This approach involves developing an analytical and physical understanding of the relationships among the conditions of a piece of equipment, the parameter variations governing it and the degradation process. Knowing the relationships between symptoms and failures, it is possible to anticipate breakdowns or to locate the cause, in case of a fault. This approach is not contradictory to statistical diagnosis; on the contrary, it can be considered an indispensable tool for improvement of the learning data, through a reduction in the number of preliminary trials required and a selection of only those condition variables which are closely linked to actual equipment degradation.

It is therefore considered that a fuzzy approach to such complex system problems as those found in mining may have certain advantages, such as filtering out insignificant data. In general, fuzzy mathematics is deemed to be a useful tool for any engineering problem which involves a large body of data and a general understanding of the system, but which requires a prompt solution. Also, the concept of linguistic variables and hedges used in fuzzy logic is very effective in modeling various human sensory measures mathematically. These human perceptions or observations are among the most important sources of information on equipment degradation processes. Success of this fuzzy logic-based diagnostic algorithm will depend upon up-to-dateness of the data used and on defining the symptom-failure relations accurately.

Chapter 4

FAULT DIAGNOSIS

4.1 Introduction

In case of conventional techniques for fault diagnosis, the analysis proceeds through various types of logic diagrams or cause and effect matrices starting from the undesirable event (operational deviation) and continues until the cause of the failure is determined. A complete understanding of the system with regard to how the effects of different faults propagate and affect the operation, is required for this type of analysis. Fault trees are constructed which focus on undesirable events and their possible causes. A complex piece of equipment like a longwall shearer would conceivably need several hundred such logic diagrams developed, each for a particular undesirable event. But the major disadvantages found with the fault tree analysis are as follows:

1. Events can be overlooked and omitted.

2. Binary logic cannot describe a partial fault accurately.

3. There is often a lack of accurate failure rate data for the different functional units.

4. Human elements like sensuous perceptions, which is very significant in diagnosis, cannot be included in the analysis.

Therefore, the conventional method of fault-tree analysis is not well suited to real-time operational usage, specially in case of complex system diagnosis. The diagnostic procedure based on the relationship between possible fault causes and their respective symptoms, established through fuzzy logic paths, seems to be more suitable for the analysis of this kind of system.

Predicting the occurrence of symptoms is a directly solvable problem in the sense of fuzzy relations and conditioned fuzzy sets when the causes and their relationships to the symptoms are known. This forward problem will always yield one solution or a symptom array for a given cause. But, solving the inverse relation, i.e., determining the causes or faults based on symptoms observed, is somewhat complex. For example, one can say that in an electric circuit, if the switch is open then there will be no current in the circuit. However, the inverse of the statement "if there is no current in the circuit, then the switch is open" cannot be assumed. There may be other causes responsible for the circuit to have no current. Tsukamoto and Terano solved this problem by placing upper and lower bounds on the possible fault array (Tsukamoto and Terano, 1977). Their solution procedure demands that the symptoms are known and the cause-consequence relations are understood. These bounds are analogous to the natural expressions, i. e., the upper bound represents the possibility of occurrence of a fault and the lower bound the necessity of existence for that fault. It is obvious that the necessity of existence of a fault should not exceed the possibility of its existence. In order to perform diagnosis, it is necessary that the symptoms are observed and the observation matrix between the faults and the symptoms is established. The observation of the symptoms is a fuzzy membership function U_b, where $U_b(j)$ represents the relative degree of truth to which the jth symptom is observed. The components t_{ij}'s of the observation matrix indicate the actual degree of certainty for the observation of a particular symptom b_j when fault a_i occurs. The diagnosis places bounds on the fault array and thereby reduces the existence band upon them.

4.2 Upper Bound or Possibility of Fault Existence

The expression for the upper bound of fault possibility is derived from the logic of implication or inference. The statement "a implies b" may have different truth value for different combinations of values of existence for a and b. The truth value is an indication that the implication expression is satisfied. In ordinary set theory, a truth table for this implication expression for the various combinations of existence for elements a and b is as follows:

	$U_b = 0$	$U_b = 1$
$U_a = 0$	1	1
$U_a = 1$	0	1

Ordinary Truth Table for Implication

In the above table it is seen that when "a" does not exist, i. e., ($U_a=0$), then "b" may or may not exist, i. e., (U_b is equal to 1 or 0, respectively). But when "a" exists, "b" has to exist. The only combination which does not satisfy the truth of implication is when U_a and U_b are equal to one and zero, respectively. In binary logic different combinations may take same truth value, but in multi-valued logic or fuzzy sets the results will be different for different values of existence for a and b.

Kaufmann suggested the truth value for the implication statement "a implies b", in case of a fuzzy set, as:

$$U_t = \overline{U}_a \bigcup U_b \tag{4.1}$$

where $\overline{U}_a$ is the complement of the existence value for a. A more widely used expression for implication, derived from Lukasiewics multi-valued logic, as follows:

$$U_t = \min(1, 1 - U_a + U_b) \tag{4.2}$$

These two approaches yield identical results when either element is non- fuzzy. What makes Lukasiewics formula more acceptable in case of a fuzzy implication is that the truth value of the implication is one whenever a supports b. The relation 4.2 indicates that the occurrence of "a" implies the occurrence of "b" if a cause-consequence relation U_t exists. Now the propositions "a" and "b" can be deemed as fault vector a_i and symptom vector b_j, and U_t is the truth or observation relation t_{ij}. Thus for the complete fault and symptom array, a truth or observation matrix, T, can be formed. The components of the matrix indicate the actual degree of certainty for the observation of a particular symptom b_j when fault a_i occurs. The aspect of observation is significant because of the possibility of insufficient instrumentation, misread data or instrument malfunction. The T matrix is therefore called the observation matrix. Lukasiewics formula is chosen to represent this implication because of the partial relations of the symptoms to different faults. From the relation 4.2 the implication expression can be written as:

$$U_t(i,j) = \min\{1,(1 - U_a(i) + U_b(j))\} \tag{4.3}$$

For diagnosis, the symptoms $U_b(j)$'s are known, the observation relations $U_t(i,j)$'s are understood and the causes $U_a(i)$'s are to be determined. The fault variable may be separated by evaluating two cases.

Case I : When $U_a(i) \geq U_b(j)$

$U_t(i,j) = \min\{1,(1 - U_a(i) + U_b(j))\} = 1 - U_a(i) + U_b(j) \leq 1.0$

$U_a(i) = 1 - U_t(i,j) + U_b(j)$

This is further equivalent to

$U_a(i) = \min\{1,(1 - U_t(i,j) + U_b(j))\}$

since the "min" operation assigns the value as $(1 - U_t(i,j) + U_b(j))$ when $U_a(i) \geq U_b(j)$

Case II : When $U_a(i) < U_b(j)$

$U_t(i,j) = \min\{1,(1 - U_a(i) + U_b(j)\} = 1.0$

$U_t(i,j) < (1 - U_a(i) + U_b(j)) \leq 2.0$

$U_a(i) < (1 - U_t(i,j) + U_b(j)) \leq 1.0$

$U_a(i) < \min\{1,(1 - U_t(i,j) + U_b(j))\}$

The expression is reached following the same arguments as discussed in Case I. These two possibilities are combined and satisfied by the inequality

$$U_a(i) \leq \min\{1,(1 - U_t(i,j) + U_b(j))\} \tag{4.4}$$

The inequality in the expression may also be interpreted from the fact that, truth value that a particular symptom for a particular fault has been observed cannot be greater than the truth value that the fault can exist. Another fault(s) must be responsible for for this value of $U_b(j)$ of the symptom. Therefore, the above expression puts an upper bound on the possibility that a fault a_i exists based on the degree to which a symptom b_j is observed. In order to obtain an upper limit for the ith fault, the expression above must be satisfied for each symptom. This is given as:

$$U_a(i) \leq \min_j [\, \min\{1, (1 - U_t(i,j) + U_b(j)\}] \qquad (4.5)$$

Therefore, given the observation matrix T and an array B, an upper limit, i.e., the possibility of existence for the fault vector A, can be found.

4.3 Lower Bound or Necessity of Fault Existence

A lower bound on the fault vector representing the necessity of a fault's existence can also be established. This results from considering the symptoms as a conditioned fuzzy set (B), because their occurrence is contingent upon the relative existence of faults (A). A fault-symptom relationship matrix, R, can be defined whose components r_{ij}s indicate whether or not a certain fault, a_i, is related to a certain symptom, b_j. When the relationship exists, r_{ij} takes a value of one; otherwise it is equal to zero. Therefore, the components of this matrix are either zero or one. The values of the conditioned fuzzy set of symptoms are found from the following fuzzy relational equation, which is known as the "maxmin compositional rule of inference" (Zadeh, 1974).

$$b = a \bullet R, \quad U_b(j) = \max_i [\, \min\{U_a(i), U_r(i,j)\}] \qquad 1 \leq j \leq N \qquad (4.6)$$

The above relationship can be interpreted from the viewpoint of causal relations between faults and symptoms, which are as follows:

1. No symptoms are detected in the absence of any fault.

2. A fault will cause all the symptoms related to it.

3. A symptom is observed only when one of the faults related to it occurs.

4. When two or more faults occur simultaneously, then the intensity or exactitude of the symptom observed is determined by the maximum of the values caused by each fault.

From this discussion, it is clear that fuzzy relational equation 4.6 is a fuzzy model for representing the causal relationship between faults and symptoms. However, this conditioned relation equation is modified into an inequality, as the existence of a symptom cannot exceed the greatest existence of a related fault. The related inequality becomes:

$$U_b(i) \leq \max_{i}[\ \min\{U_a(i),\ U_r(i,j)\}] \tag{4.7}$$

A more detailed explanation may be found in Tsukamoto and Terano, 1977. In the above relation as forward equation, upper limits are assigned to the symptoms when the fault is known. However, solving the inverse relation, i.e., determining the causes or faults based on the symptoms, is somewhat complex because of the possibility of multiple configurations of the solution. The inverse relation as a tool for diagnosis establishes one or more configurations of minimum fault conditions that would support the symptoms. Two simple cases are considered to illustrate how the multiple solutions may occur:

Case I :

Assume one symptom, b_1, and three possible faults, a_1, a_2 and a_3. A fault-symptom relationship matrix indicates that this particular symptom may be caused by fault $a_1, a_2,$ or both. Further assume $U_b(1) = 0.8$. This simple case may be expressed as:

$$R: \begin{array}{c|c|} & b_1 \\ \hline a_1 & 1 \\ a_2 & 1 \\ a_3 & 0 \\ \hline \end{array}$$

$$\text{and } U_b(1) = 0.8$$

For fault vector A to satisfy the lower bound (necessity of existence) ineqality, satisfactory solutions are found to be

$$a(1) \geq 0.8$$

or:

$$a(2) \geq 0.8$$

Case II :

In this case assume there exist two symptoms and three possible faults. The relationship matrix between them is given as below. Assume the symptoms' observed values are $U_b(j) = 0.8, 0.4$.

$$R = \begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \end{array} \begin{array}{c} \begin{array}{cc} b_1 & b_2 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \end{array}$$

$$\text{and } U_b(j) = 0.8, 0.4$$

Again, A has several configurations satisfying the inequality:

$U_a(1) \geq 0.8$ and $U_a(2) \geq 0.4$;

$U_a(1) \geq 0.8$ and $U_a(3) \geq 0.4$; or

$U_a(2) \geq 0.8$,

Thus, the fault vector is completely bounded with the implication and the relational inequalities. The upper bound is determined by the possibility of existence inequality. The lower bound, found with the necessity of existence inequality, will have one or more configurations. It is obvious that the necessity of existence of a fault should not exceed the possibility of its existence. Thus any configuration in which the lower bound exceeds the upper bound is not accepted as a satisfactory solution.

4.4 Application Procedure

Applying the diagnostic technique to a system would, foremost, require a complete understanding of the system and its behavior. Development of the model discussed above may be summarized through the following basic steps:

1. All possible faults and related symptoms are identified. Operator's experience is of great help in this matter. These faults and the symptoms are those which have been historically observed.

2. The relationship between the various faults and symptoms is identified and a relationship matrix is formed. The elements of R take a value of one or zero depending upon whether or not a relation exists, respectively.

3. In this step the truth or observation matrix between different faults and symptoms is established. The elements of this matrix take a value anywhere in the interval [0,1]. Assigning the value of t_{ij}, which is the value of the implication statement "the fault i implies the symptom j," is a subjective step based on understanding of the system and operational experience. For a particular fault, Tsukamoto and Terano have suggested t_{ij} as 0.9, 0.6, 0.3 and 0.0 when the symptom is observed certainly, usually, sometimes, or seldom, respectively.

4. Membership or characteristic functions for symptoms observed and faults occurred are determined to represent the system best. Zadeh's linguistic variables like "very hot", "very noisy," etc. are helpful in quantifying these expressions

5. The model is tested for reasonable result using sample values, preferably drawn from experience.

A flow-chart detailing the above basic steps is given in Appendix A.

When the above results are not satisfactory, or a solution cannot be found, the following steps are considered:

1. Verify whether the symptoms have been accurately observed, particularly those that cannot be supported by any fault pattern.

2. The upper and lower bounds for $U_a(i)$ should be reviewed as a list of possible faults.

3. Ensure that no faults or symptoms have been omitted from the model.

4. The R and T matrices should be reviewed carefully for errors in logic.

A final comment on application concerns the assembly of the observation matrix, T. As the values $U_t(i,j)$ are generally reduced, a greater number of solutions for A will be found and each configuration will have a broader range between limits. Generally if U_t (i,j) is increased, then the solutions will be more specific and less frequent. Therefore, it is advised that T(i,j) be adjusted with experience.

4.5 Diagnostic Algorithm

The diagnostic algorithm consists of six specific steps. The function of each step is briefly discussed below.

Step 1: Determine the fault upper bound ($ub_a(i)$)

First, the R matrix is formed from the T matrix input. An element of the R matrix takes a value of zero when the corresponding element in the T matrix is equal to zero, otherwise it assumes a value of one. Now with the T matrix and the observed symptom array (B(j)) input, the upper limit or possibility of fault existence is found directly from the inequality:

$$ub_a(i) \leq \min_j[\ \min\{1 - U_t(i,j) + U_b(j)\}]$$

Step 2: Find the greatest lower bound ($\mathrm{glb}_a(i)$)

The greatest lower bound is a value above which no lower bound will be found. This indicates the search for various lower bound configurations, and is obtained from the expression:

$$\mathrm{glb}_a(i) = \max_j[\ \min\{U_r(i,j), U_b(j)\}]$$

Step 3: Reduce the glb to meet the upper limit

Each greatest lower bound is output with any necessary reduction flagged. The reduction may be expressed as:

$$\mathrm{glb}_a(i) = \min\{\ \mathrm{glb}_a(i), ub_a(i)\}$$

Step 4: Determine whether a solution exists

In Step 3 the greatest lower bound was adjusted to meet the upper limit. It must then be verified that the lower limit expression is still satisfied. With glb(i) substituted for $U_a(i)$, a solution exists if the expression

$$U_b(j) \leq \max_i[\ \min\{U_r(i,j),\ glb_a(i)\}]$$

holds for any i and j. When the inequality can not be satisfied, the unsupported symptoms are flagged and the rest of the program is bypassed.

Step 5: Determine the existence of multiple solutions

When it is verified that a solution exists, the elements of glb are minimized recursively M times to a lower bound. The first reduction minimized $glb_a(1)$, then $glb_a(2)$, and proceeds through $glb_a(M)$. The next reduction commences with glb(2) and continues. Each reduction generates a lower bound configuration. In M reductions each fault is minimized first once, i. e., each reduction starts with separate fault. The results are compared in order to identify the lower bound configurations. Multiple configurations exist when two or more lower bounds vary.

Step 6: Obtain the various lower bound configurations

In order to ensure that all the solutions are identified, faults with varying lower bounds must be in every sequence of order. If, for example, there exist three faults (a_2, a_4, a_6) with varying lower limits, six possible minimization orders exist:

$$a_2, a_4, a_6; a_4, a_6, a_2; a_6, a_2, a_6;$$

$$a_2, a_6, a_4; a_4, a_2, a_6, a_6, a_4, a_2.$$

Subroutine PERMUT generates the permutations of order for minimization. When L lower bounds vary, L factorial permutations of order should exist. Each permutation generates a solution which is compared with existing solutions, and included if unique. Each unique result is a satisfactory lower bound.

Chapter 5

PROGRAM DESCRIPTION

The program is developed following the algorithm discussed in Chapter 4. The microcomputer version of the algorithm is written in FORTRAN and it runs on the IBM-PC and PCXT. Minimum system configuration requirements are a RAM (capacity depending on the number of faults and symptoms), monitor (color or monochrome), keyboard and printer. A system having 30 different modes of failure and 30 different kinds of possible symptom, would require about 10KB RAM. The working computer code consists of a main program and a subroutine.

5.1 Main Program

The main program executes six different steps as discussed in Chapter 4 and finds the possible fault array and establishes a lower and an upper bound probabilities for them. The important variables used in the program are as follows:

M = numbers of faults

N = numbers of symptoms

T(I,J) = element of the observation matrix associated with ith faults and jth symptom

R(I,J) = element of the relationship matrix associated with ith fault and jth symptom

UL(I) = upper limit for ith fault

X(I) = GLB(I) = greatest lower bound for ith fault

MODE = number of symptoms for which the conditions cannot be met

L = number of faults with varying lower bounds

NUM(L) = fault number of the Lth component of the array of faults which have varying lower bound

Z(K,I) = lower bound for ith fault in the kth recursive reduction

IZ = counter for the number of permutations passed by the subroutine PERMUT

NC = counter for number of unique lower bound configurations for the faults

F(IK,(NUM(NL))) = array for already existing solutions which are compared with a new solution for uniqueness

5.2 Program Execution

The program is executed by entering FDSYS. The following will appear on the screen.

```
*********************************************************
*                  FDSYS
*           FAULT DIAGNOSIS SYSTEM
*                Tapas R. Kar
*  Department of Mining & Minerals Engineering
*         Virginia Polytechnic Institute &
*               State University
*                December 1989
*********************************************************
```

Now to run the program follow the instructions

PREPARATION OF DATA, ENTER OPTION 1: CREATE NEW FILE
2: CHANGE EXISTING FILE
3: USE EXISTING FILE

--enter option 3 if the numbers of faults and symptoms and the observation matrix already exist in the file. If some data are to be changed, enter option 2.

If the option is 1, then the following will appear:

ENTER NAME OF INPUT FILE

--enter the name of the data file that wiil be created when the data is input interactively. <RETURN> (e. g. PAPI.DAT).

ENTER THE OUTPUT OPTION 1: PRINTER
2: FILE

--enter option <RETURN>. If the choice is 2, the program will prompt:

ENTER OPTION, 1: CREATE NEW FILE
2: USE EXISTING FILE NAME

If the option is 1,

ENTER THE NAME OF THE OUTPUT FILE

--enter the output file name <RETURN>. (e. g. SONA.OUT)

Now if the option in the data preparation is 3 then the program starts executing with the existing file. Otherwise, it will ask:

ENTER THE NOS. OF FAULTS AND THE NOS OF SYMPTOMS

--enter the numbers of faults and symptoms <RETURN>. (e. g. 9, 15)

ENTER THE NAME OF FAULT 1

--enter the first fault name. (e. g. System demand exceeds rating.

Thus all the fault names are input.
ENTER THE NAME OF THE SYMPTOM 1

--enter the name of the first symptom. (e. g.Compressor noisy or knocks)
Thus all the symptom names are entered.

Now the program will prompt:

TO FORM THE OBSERVATION MATRIX SELECT THE RELATIONSHIP BETWEEN THE AND THE SYMPTOMS, BASED ON THE GRADE OF OCCURRENCE OF THE SYMPTOMS CAUSED BY THE FAULTS.

1: VERY OFTEN OCCURS

2: OFTEN OCCURS

3: QUITE OFTEN OCCURS

4: SOMETIMES OCCURS

5: SCARCELY OCCURS

6: VERY SCARCELY OCCURS

7: DOES NOT OCCUR

ENTER THE RELATIONSHIP FROM THE ABOVE BETWEEN FAULT 1 AND SYMPTOM 1

--enter the degree to which the symptom 1 is observed when the fault 1 occurs. (e. g. 7)

Thus all the elements of the observation matrix are entered row-wise interactively.

The program will screen all the system fault and symptoms types. For example, in case of compressor diagnosis it will show:

FAILURES

1: SYSTEM DEMAND EXCEEDS RATING

2: ELECTRICAL CONDITIONS WRONG

3: BELT SLIPPING

4: BEARINGS NEED ADJUSTMENT OR RENEWAL

5: OIL LEVEL TOO HIGH

6: AIR FLOW TO FAN BLOCKED

7: OIL LEVEL TOO LOW

8: RESONANT PULSATION

9: INTAKE FILTER CLOGGED

SYMPTOMS

1: COMPRESSOR NOISY OR KNOCKS

2: OPERATING CYCLE ABNORMALLY LONG

3: DELIVERY AT LESS-THAN-RATED CAPACITY

4: DISCHARGE PRESSURE BELOW NORMAL

5: INTERCOOLER PRESSURE NOT NORMAL

6: AIR DISCHARGE TEMPERATURE ABOVE NORMAL

7: CARBONACEOUS DEPOSITS ABNORMAL

8: MOTOR OVERHEATING

9: COMPRESSOR PARTS OVERHEATING

10: CRANKCASE OIL PRESSURE LOW

11: OIL PUMPING EXCESSIVE

12: PISTON RING, PISTON CYLINDER WEAR EXCESSIVE

13: COMPRESSOR FAILS TO START

14: INTERCOOLER SAFETY VALVE POPS

15: VALVE WEAR AND BREAKAGE EXCESSIVE

Finally, the program will prompt:

ENTER THE OBSERVED SYMPTOMS VALUES, B(J), BASED ON THE DEGREE TO WHICH IT IS OBSERVED

1: FOUND WITH ALMOST CERTAINTY

2: FOUND WITHH LESS CERTAINTY

3: RECOGNIZED STRONGLY

4: RECOGNIZED LESS STRONGLY

5: FELT STRONGLY

6: FELT LESS STRONGLY

7: SUSPECTED

8: NOT OBSERVED

ENTER THE DEGREE TO WHICH THHE SYMPTOM 1 IS OBSERVED

--enter the intensity of occurrence of the symptom 1. (e. g. 1)

Thus the exactitude or intensity of all the symptoms observed are entered. For the compressor example in case I these are:

(1, 8, 8, 8, 8, 7, 6, 1, 7, 5, 8, 8, 8, 8, 8)

EXECUTION STARTED, PLEASE WAIT

If the output option selected is PRINTER then after a few seconds theprogram will print the solutions. If the output option selected is FILE, then the output file will be created on the disk. This file will bear the name given to it at the beginning of the program. The program can be run several times with different sets of adjusted observed symptom values, obtained through repeated modifications, in order to obtain a satisfactory solution.

Chapter 6

EXAMPLES OF FAULT DIAGNOSIS

The application of the fuzzy logic-based diagnostic method is illustrated by two examples, one involving a compressor failure and the other, faults in a longwall shearer.

6.1 Compressor

There may be many kinds of compressor failure and many symptoms which may be observed, but in this example only nine types of abnormal state will be identified as the causes of compressor trouble. The fault and the symptom arrays are shown in Table 4. The relationship matrix R and the observation matrix T are set up based on whether or not a particular symptom is related to a fault and the degree to which the symptom is observed when the fault occurs, respectively. This information is obtained from the Ingersoll Rand handbook on compressed air and gas data (2). From the R matrix, it is evident that the fault, belt slipping, a_3, is related to symptoms such as compressor noise, b_1; delivery of less-than-rated capacity,

b_3; and low discharge pressure, b_4. The observation matrix illustrates the degree of occurrence of these symptoms; in the case of belt slipping, the compressor noise or knocks ($t_{31}=0.9$) are predominantly observed, while delivery volume ($t_{33}=0.5$) and discharge pressure ($t_{34}=0.3$) are observed to a lesser degree.

Case I

A particular case will now be discussed in which the operator perceives quite a high rate of compressor knocks ($b_1=0.8$) and high motor temperature ($b_8=0.8$), feels abnormal carbonaceous deposits ($b_7=0.3$) and a low crankcase oil pressure ($b_{10}=0.4$), and suspects an air discharge temperature slightly higher than normal ($b_6=0.2$) and slightly overheated compressor parts ($b_9=0.2$). The values of the exactitudes of the symptoms are, therefore, as follows:

$$b_j\text{:}[0.8,0.0,0.0,0.0,0.0,0.2,0.3,0.8,0.2,0.4,0.0,0.0, 0.0,0.0,0.0]$$

The solutions to this problem given by the implication expression equation 4.5 and the inverse operation of equation 4.7, are as follows:

$$upperbound\text{:}[0.3,0.1,0.5,0.9,0.1,0.5,0.5,0.3,0.3]$$

$$lowerbound_1\text{:}[0.0,0.0,0.0,0.8,0.0,0.3,0.0,0.0,0.0]$$

$$lowerbound_2\text{:}[0.0,0.0,0.0,0.8,0.0,0.0,0.0,0.0,0.3]$$

Appendix C lists the computer output for the problem. The upperbound (ub) solution tells us about the possibilities of the occurrences to a great extent: of "worn or misadjusted bearings," a_4 ($ub_4 = 0.9$); "belt slipping," a_3($ub_3 = 0.5$); "blocked air flow to the fan," a_6 ($ub_6 = 0.5$); "oil level too low," a_7 ($ub_7 = 0.5$); and, to a lesser extent, "high system demand," a_1; "resonant pulsation," a_8 ; and "clogged intake filter," a_9. The lowerbound solutions (lb) indicate that there is a "worn or misadjusted bearing" ($lowerbound_4 = 0.8$), but other kinds of faults may or may not be present. Therefore it can be ascertained that a "worn or misadjusted bearing" is the cause of the failure. Since, however, the lowerbound solutions represent the least necessary abnormal state to bring about the given symptoms, a Fault 6 or 9 may be found to exist. This may, however, be due to weak values assigned to some of the symptoms observed. Therefore, although some of the symptoms observed are vague, the solution obtained is clear enough to allow the diagnosis to be carried out.

Case II

In this example the operator finds that the delivery is less than the rated capacity and the air discharge temperature is quite high ($b_3 = 0.8$ and $b_6 = 0.7$), recognizes that the operating cycle is longer than normal and feels that excessive oil pumping is occurring ($b_2 = 0.5$ and $b_{11} = 0.4$), and, he also feels abnormal carbonaceous deposits ($b_7 = 0.3$) and suspects low discharge pressure ($b_4 = 0.2$). The values of the exactitudes of the symptoms are given as follows:

$$b_j\text{:}[0.0, 0.5, 0.8, 0.2, 0.0, 0.7, 0.3, 0.0, 0.0, 0.0, 0.4, 0.0, 0.0, 0.0, 0.0]$$

The solution (Appendix C) to this problem is found to be as follows:

$$\textit{upperbound}\text{:}[0.7, 0.1, 0.1, 0.3, 0.5, 0.5, 0.3, 0.3, 0.8]$$

$$\textit{lowerbound}\text{:}[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.8]$$

The upperbound solution provides information about the possibilities of existences of the faults and shows a "clogged intake filter" to be the most probable. This problem has only one lowerbound solution, thereby reducing the ambiguity of the diagnosis; the most probable fault is also confirmed by the lowerbound solution with $\textit{lowerbound}_9 = 0.8$ and the other fault types being equal to 0.0.

Table 4. Relationship and Observation Matrix between Failures and Symptoms in a Compressor

Relationship matrix

a_i \ b_j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0
5	1	0	0	0	0	1	1	0	1	0	1	0	0	0	0
6	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0
8	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0
9	0	1	1	1	1	1	0	1	1	0	1	0	0	0	0

Observation matrix

a_i \ b_j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.0	0.7	0.5	0.5	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.9	0.0	0.0
3	0.9	0.0	0.5	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.3	0.5	0.0	0.0	0.0	0.0	0.0
5	0.5	0.0	0.0	0.0	0.0	0.3	0.7	0.0	0.3	0.0	0.9	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.7	0.5	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.7	0.0	0.5	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.7	0.0
9	0.0	0.5	0.7	0.3	0.2	0.7	0.5	0.2	0.2	0.0	0.5	0.0	0.0	0.0	0.0

Failures

1: System demand exceeds rating
2: Electrical conditions wrong
3: Belt slipping
4: Bearings need adjustment or renewal
5: Oil level too high
6: Air flow to fan blocked
7: Oil level too low
8: Resonant pulsation
9: Intake filter clogged

Symptoms

1: Compressor noisy or knocks
2: Operating cycle abnormally long
3: Delivery at less than rated capacity
4: Discharge pressure below normal
5: Intercooler pressure not normal
6: Air discharge temperature above normal
7: Carbonaceous deposits abnormal
8: Motor overheating
9: Compressor parts overheating
10: Crankcase oil pressure low
11: Oil pumping excessive (single-acting compressor)
12: Piston ring, piston cylinder wear excessive
13: Compressor fails to start
14: Intercooler safety valve pops
15: Valve wear and breakage excessive

6.2 Longwall Shearer

In this example seven different types of fault and eleven symptoms are identified, although there may be many causes of failure and associated symptoms. The diagnosis of the faults is performed based on the relationship and observation matrix and on the symptoms observed. The list of faults and symptoms, and the relationship and the observation matrix between them, are shown in Table 5.

Suppose that an operator recognizes that the shearer unit does not haul smoothly, finds that a high priming pressure is required for the power pack to pressure up to the stall/relief valve setting, and that the auxiliary circuit is not operating satisfactorily. The operator also suspects the hydraulic fluid level to be slightly below the minimum mark and the push-button pressure to be a little below the setting pressure. The symptom array may be established subjectively as:

$$b_j\text{:}[0.6,0.0,0.0,0.0,0.7,0.2,0.8,0.0,0.2,0.7,0.0]$$

The solution (Appendix C) to the problem is given as follows:

$$upperbound\text{:}[0.5,0.5,0.3,0.5,0.8,0.1,0.5]$$

$$lowerbound_1\text{:}[0.0,0.0,0.0,0.2,0.8,0.0,0.0]$$

$$lowerbound_2\text{:}[0.0,0.0,0.0,0.0,0.8,0.0,0.2]$$

The upperbound solution shows that the occurrence of "faulty auxiliary pump suction or delivery connections" is the most likely cause of failure. This is confirmed by the lowerbound solutions. The small probability for necessity of existence, i.e., lowerbound for Faults 4 or 6 as given by the lowerbound solutions, may be due to some vagueness in the observed input data. The diagnosis becomes clearer as the symptom data become more accurate, i.e., when b_i's are either equal to zero or close to one.

Table 5. Relationship and Observation Matrix between Failures and Symptoms in a Longwall Shearer

Relationship matrix

a_i \ b_i	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	0	1	0	0	0	0	0	0
2	1	0	0	0	1	0	1	0	0	0	1
3	1	0	0	1	1	0	1	0	0	0	1
4	1	0	0	1	1	1	1	0	1	1	1
5	1	0	0	0	1	1	1	0	0	1	0
6	0	0	0	0	1	1	1	1	1	1	0
7	0	0	0	0	1	1	1	1	1	1	0

Observation matrix

a_i \ b_i	1	2	3	4	5	6	7	8	9	10	11
1	0.7	0.5	0.3	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0
2	0.9	0.0	0.0	0.0	0.5	0.0	0.7	0.0	0.0	0.0	0.5
3	0.5	0.0	0.0	0.7	0.3	0.0	0.5	0.0	0.0	0.0	0.2
4	0.3	0.0	0.0	0.3	0.5	0.7	0.7	0.0	0.5	0.2	0.3
5	0.5	0.0	0.0	0.0	0.7	0.3	0.7	0.0	0.0	0.9	0.0
6	0.0	0.0	0.0	0.0	0.5	0.3	0.5	0.9	0.7	0.7	0.0
7	0.0	0.0	0.0	0.0	0.3	0.7	0.7	0.3	0.3	0.5	0.0

Failures

1: Bearing failure
2: Electric motor retaining-relay contacts stuck open
3: Low-pressure micro-switch setting is not correct
4: Faulty low-pressure piston
5: Faulty auxiliary-pump suction or delivery connections
6: Damaged supply hoses
7: Leakage from pump casing

Symptoms

1: Shearer does not haul in either direction
2: Electric motor current above full load
3: With shearer free of cut, motor current still above full load
4: Shearer travels at minimum speed
5: Above normal priming pressure
6: Hydraulic fluid level below minimum mark
7: Power pack does not pressure up to the stall/relief valve setting
8: Leaks in push-button start lines
9: Push-button pressure below setting pressure
10: Auxiliary circuit does not operate satisfactorily
11: Power pack control handle in neutral and electric motor cuts within 4 - 5 seconds

Chapter 7

CONCLUSIONS & RECOMMENDATIONS FOR FUTURE RESEARCH

7.1 Conclusions

The effects of the proposed algorithm on the availability of equipment through early detection of abnormal states are demonstrated. It would help the mine operator to be able to detect and diagnose faults through suggestive information when certain abnormal states are perceived at an early stage.

In this diagnostic method, the upperbound solution of a, provides information regarding possible faults. The lowerbound solutions will designate the least necessary state among possible states of failure that will bring about the same exactitudes of the symptoms. As a result, when the lowerbound and the upperbound solutions take values of zero and one, respectively, they will provide no information. The existence band of the faults can be reduced by setting

r_{ij} or t_{ij} values correctly and when the symptoms are clear and more accurately noted. The process of selecting the values of r_{ij}'s or t_{ij}'s is not necessarily easy. The values need to be modified and updated through experience and more detailed analysis of the system. The smaller the value of t_{ij}, the more cases will exist in which conditions for the observed symptoms are satisfied. In turn, this situation will lead to an increased number of solutions. However, there will be less information on diagnosis as the existence band for different faults will be broader.

The lack of a satisfactory solution may be due to any one of several factors, such as mistaken observation or omission of possible faults or symptoms. The occurrence of a single configuration, i.e., one lowerbound solution, is preferred to multiple lowerbound solutions. In cases of multiple solutions, the operator is advised to make a further evaluation of the symptoms.

A computer program has been developed following the diagnosis algorithm. It identifies all possible configurations and compares them for uniqueness. The logic is streamlined in order to reduce computation time, and the memory requirement for running the program is significantly low. It is believed that, through large-scale modeling on a microcomputer, real-time diagnosis may be accomplished for the complex equipment used in mining.

7.2 Recommendations

Additional techniques may exist for placing upper limits on the failure vector a_i^*. This can be deemed as an upperbound solution on the relative likelihood that a_i is not expected. This is important because all detrimental factors responsible for a failure in a system can not be recorded. To cope with this, a confidence array, C_i, can be developed reflecting that a fault a_i will result from known causes, with a confidence, varying from 0 to 1. Comparing such a solution with a diagnostic solution is probably best accomplished with a relative Hamming distance, which is defined as the measure of difference between two sets from the same universe set. A history vector developed from the historical information can be used as a universe set. Such a comparison will favor the possibility of existence of one fault array over the rest. This analysis will be very useful in prioritizing the fault possibilities when one is confronted with multiple configurations for a fault vector. In addition, this will lend insight when no solution is given and may lead to reinforcement of the single solution.

BIBLIOGRAPHY

1. Bandopadhyay, S. and Venkatasubramanian, P., "A Fault-Diagnostic Expert System for Longwall-Shearer," Proceedings, 21st APCOM Symposium, 1989, pp. 936 -948.

2. Ingersoll Rand Co., Compressed Air and Gas Data, Ingersoll Rand Co., 1982.

3. Himmelblau, D. M., Fault Detection and Diagnosis in Chemical and Petrochemical Processes, Elsevier North-Holland Publication, New York, 1978.

4. Kaufmann, A., Introduction to the Theory of Fuzzy Subsets, Vol. I, Academic Press, New York, (translated) 1975.

5. King, P. J., and Mamdani, E. H., "The Application of Fuzzy Control Systems to Industrial Processes," in Fuzzy Automata and Decision Processes, Edited by Gupta et al., North-Holland, 1977, pp. 321 - 330.

6. Lukasiewics, J., Aristotle's Syllogistic from the Standpoint of Modern Formal Logic, Oxford University Press, New York, 1951.

7. Mamdani, E. H., "Application of Fuzzy Set Theory to Control Systems," in Fuzzy Automata and Decision Processes, Edited by Gupta et al., North-Holland Publication, 1977, pp. 77 - 88.

8. Pau, L. F., Failure Diagnosis and Performance Monitoring, Marcel Dekker Inc., 1981.

9. Ragade, R. K. and Gupta, M. M., "Fuzzy Set Theory : Introduction," in Fuzzy Automata and Decision Processes, Edited by Gupta et al., North-Holland Publication, 1977, pp. 105 - 131."

10. Sanchez, E., "Resolution of Composite Fuzzy Relation Equations, Information and Control, Vol. 30, 1976, pp. 38 - 48.

11. Sanchez, E., "Inverse of Fuzzy Relations, Applications to Possibility Distribution and Medical Diagnosis," IEEE Proceedings, Decision and Control, Vol. II, 1977, pp. 1384 - 1389.

12. Shahinpoor, M. and Wells, D. J., "Engineering Diagnostics and Troubleshooting: A New Use for Fuzzy Logic," Proceedings, Symposium on Failure Prevention and Reliability, 1981, pp. 167 - 171.

13. Terano, T. et al., "Diagnosis of Engine Trouble by Fuzzy Logic, Proceedings, IFAC Symposium, 1978, pp. 1621 - 1628.

14. Terano, T. et al., "Trouble Diagnosis System of Marine Engine, Proceedings, IFAC Symposium on Ship Operation Automation, Washington D. C., 1976, pp. 335 - 344.

15. Tsukamoto, Y. and Terano, T., "Failure Diagnosis by Using Fuzzy Logic," IEEE Proceedings, Decision & Control, Vol. II, 1977, pp. 1390 - 1395.

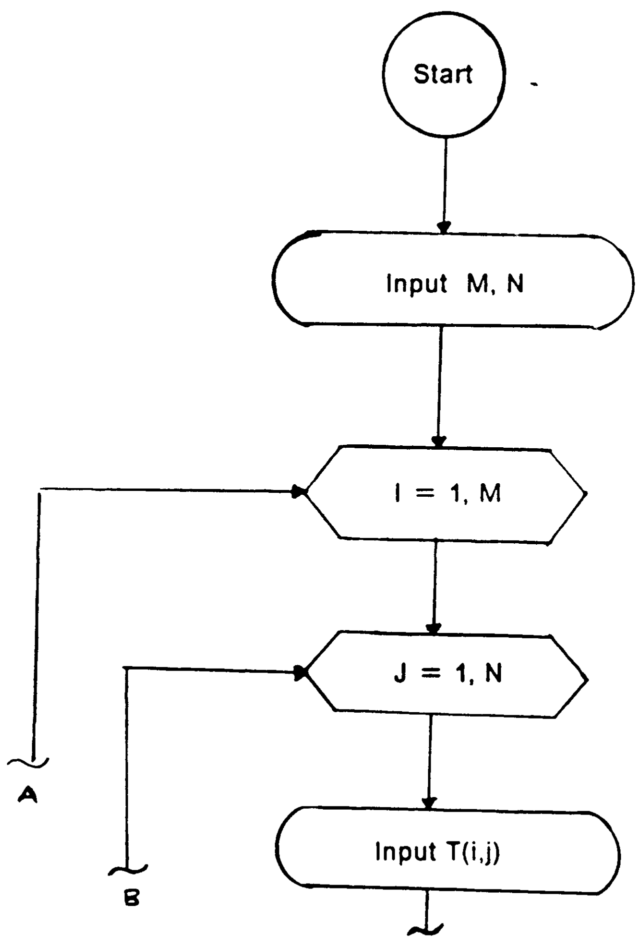
16. Zadeh, L. A., "Fuzzy Sets," Information and Control, Vol. VIII, pp. 338 - 353.

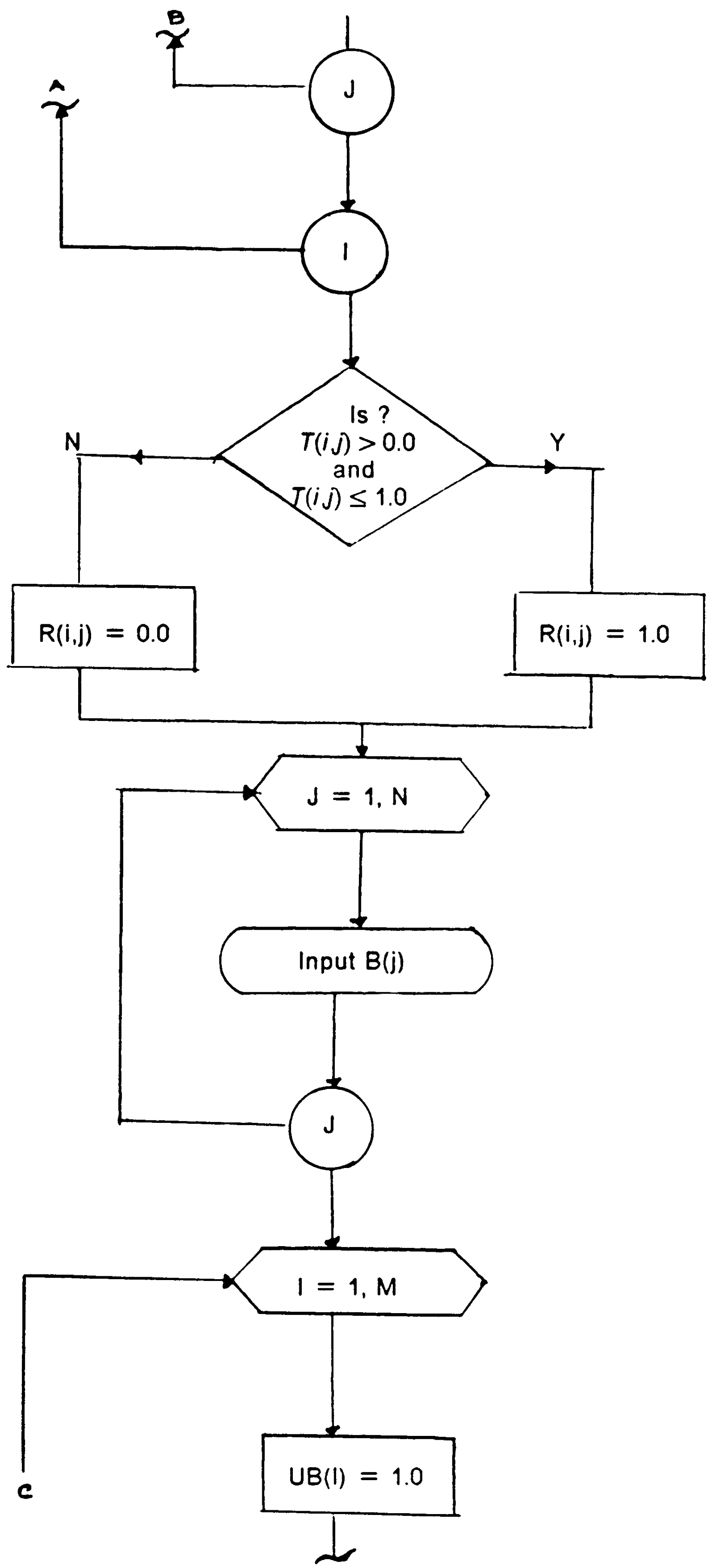
17. Zadeh, L. A., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," IEEE Transactions, on Systems, Man and Cybernetics, Vol. III, 1973, pp. 28 - 44.

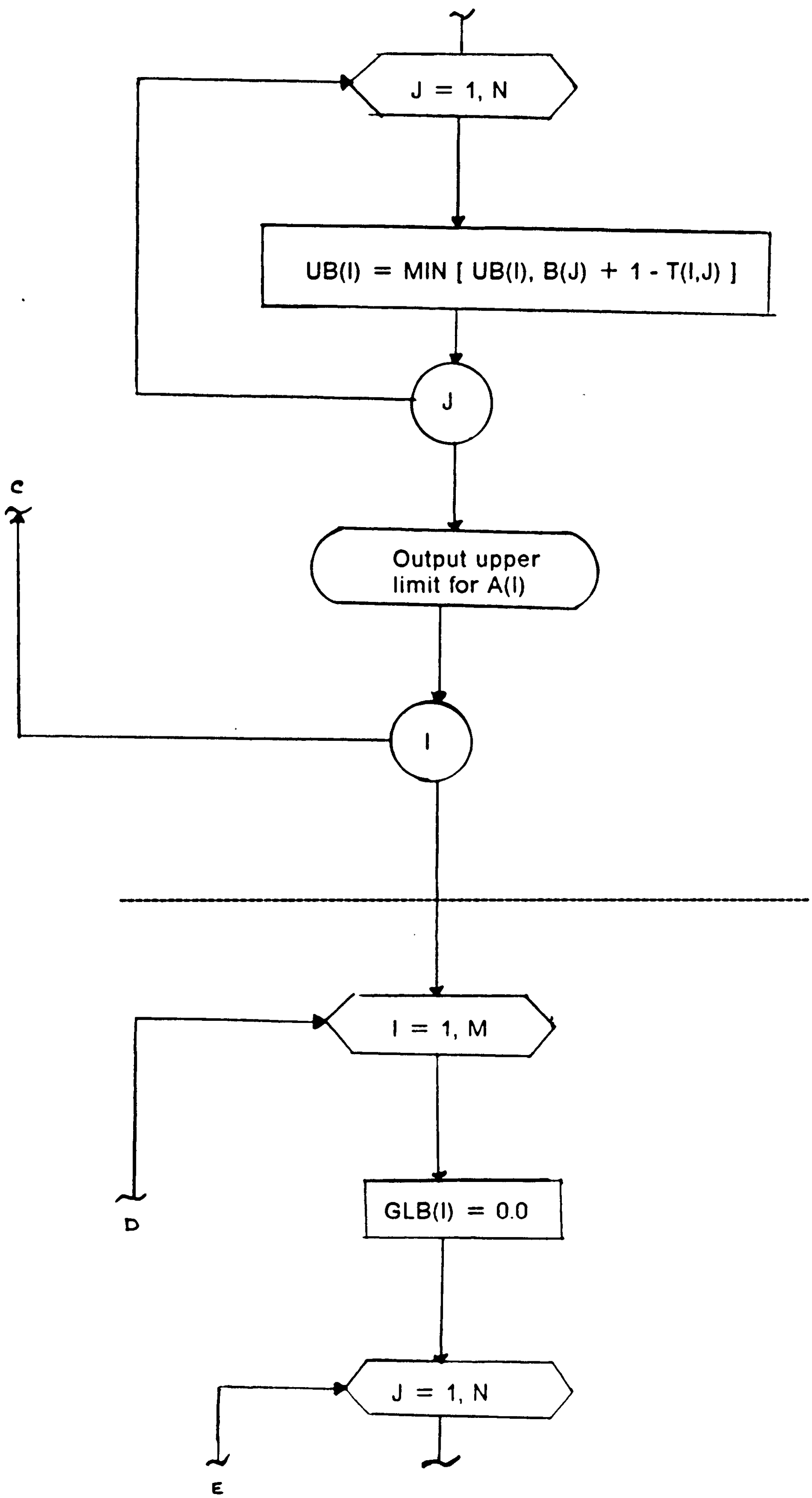
18. Zadeh, L. A., Fu, K. S., Tanaka, K. and Shimura, M., Fuzzy Sets and their Application to Cognitive and Decision Processes, Academic Press, New York, 1975.

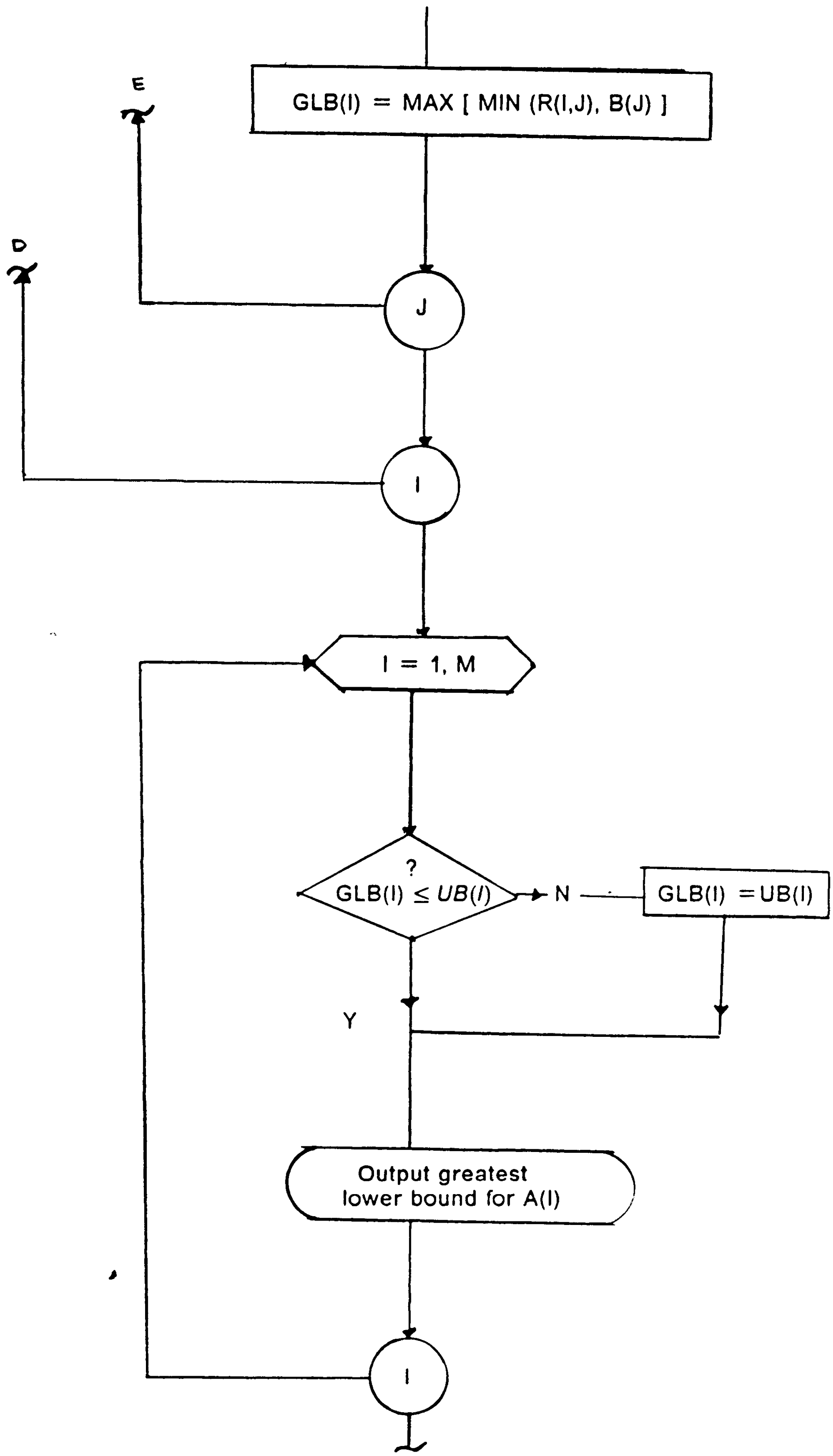
APPENDICES

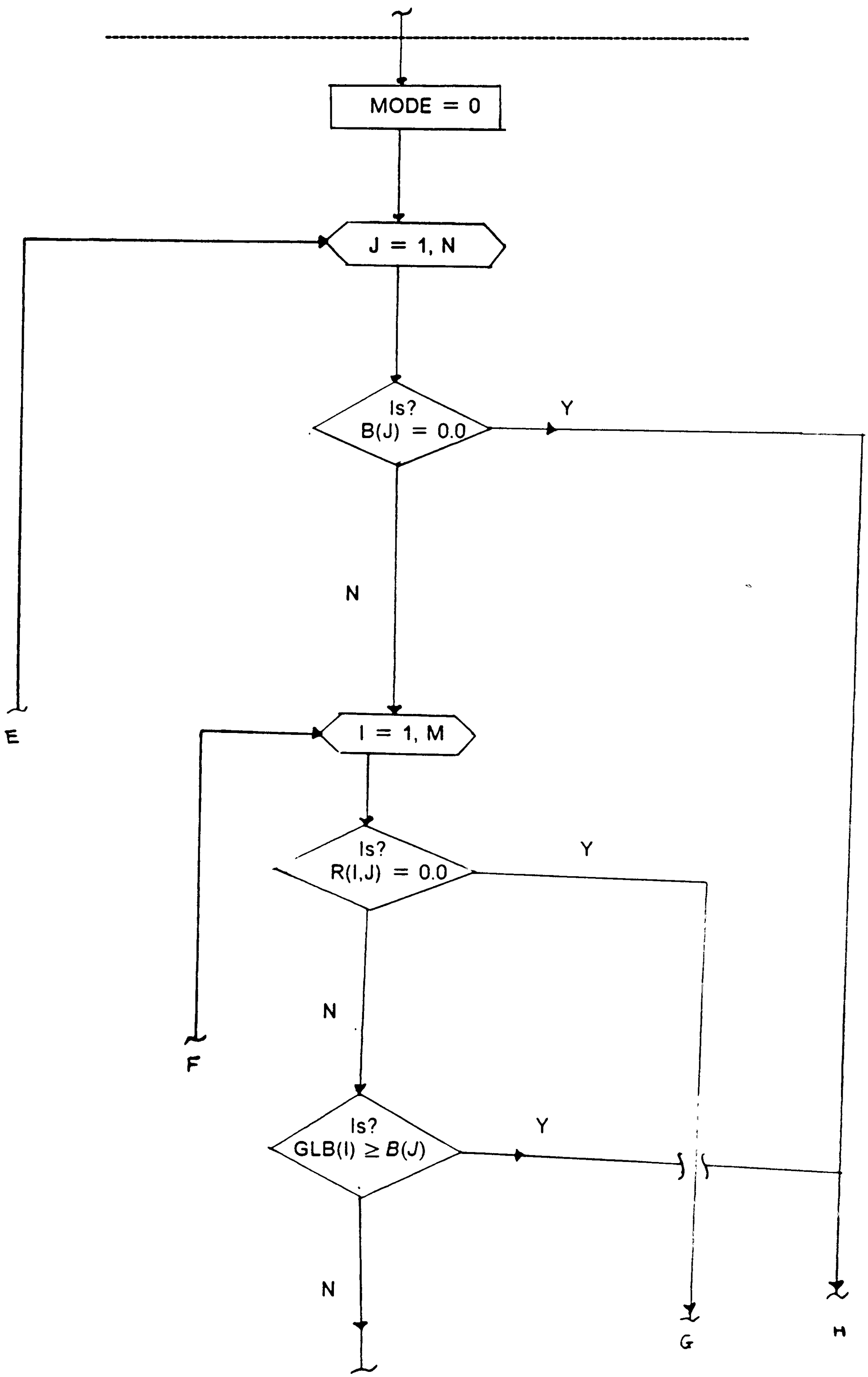
Appendix A. Fault Diagnostic Algorithm

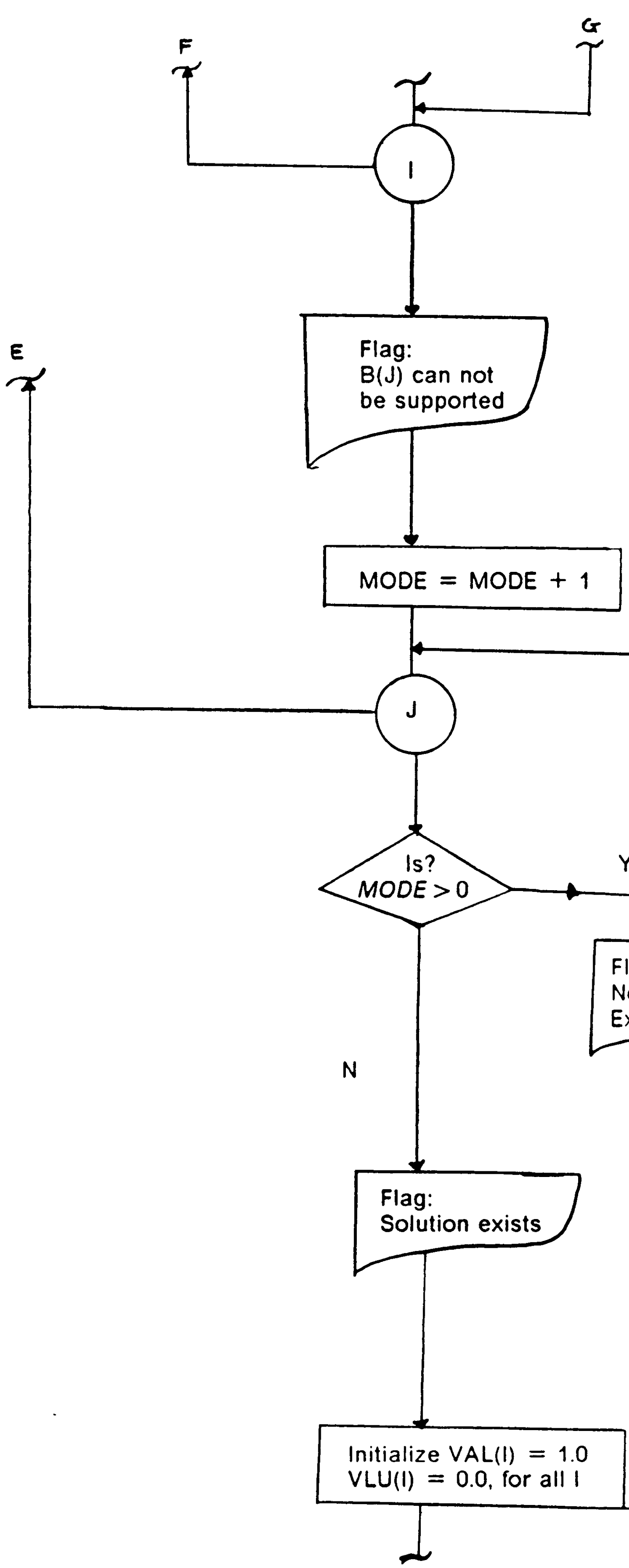


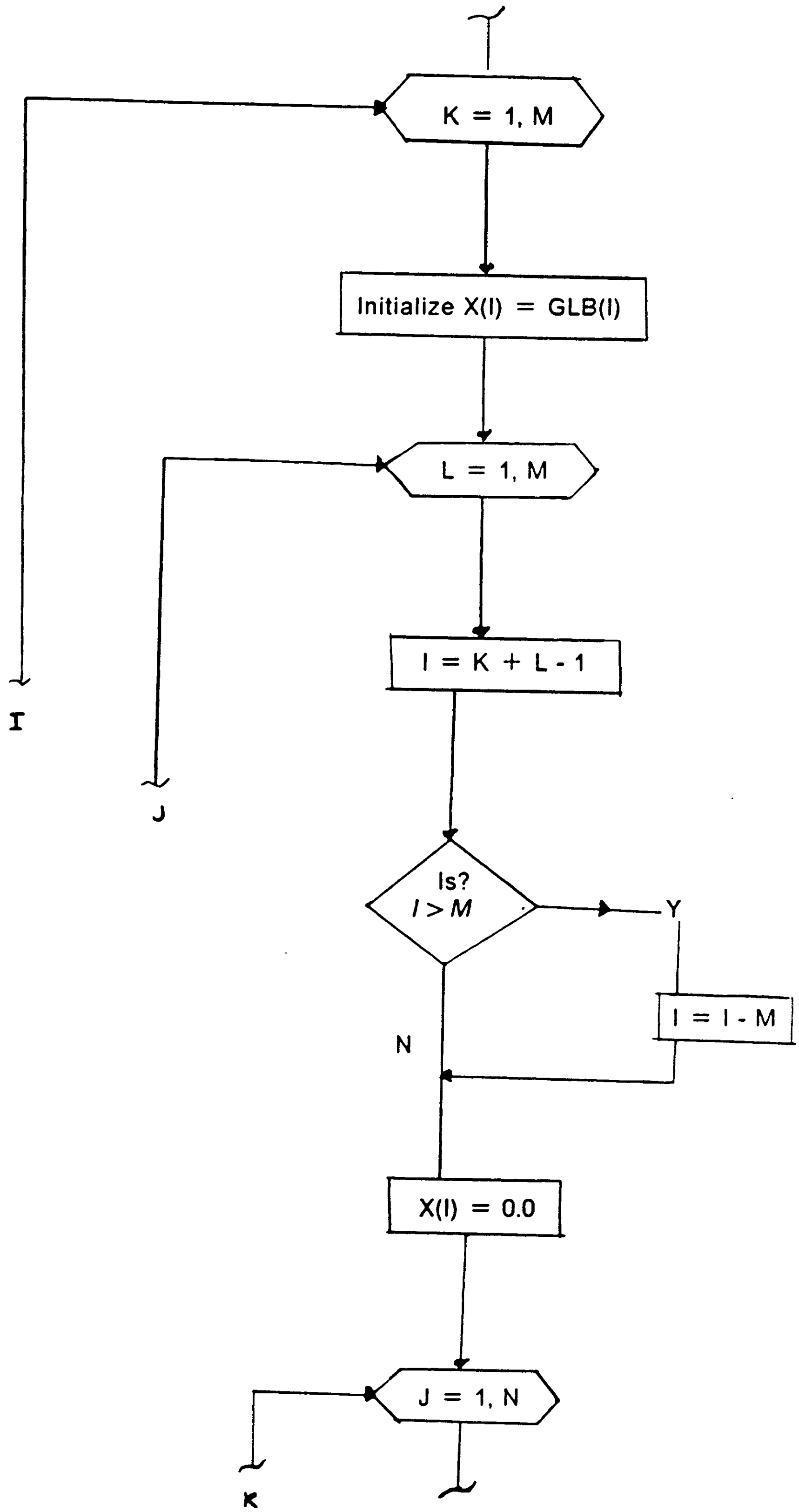

B
A
J
I
Is ?
T(i,j) > 0.0
and
T(i,j) ≤ 1.0
N
Y
R(i,j) = 0.0
R(i,j) = 1.0
J = 1, N
Input B(j)
J
I = 1, M
C
UB(I) = 1.0


J = 1, N
UB(I) = MIN [UB(I), B(J) + 1 - T(I,J)]
J
C
Output upper limit for A(I)
I
I = 1, M
D
GLB(I) = 0.0
J = 1, N
E

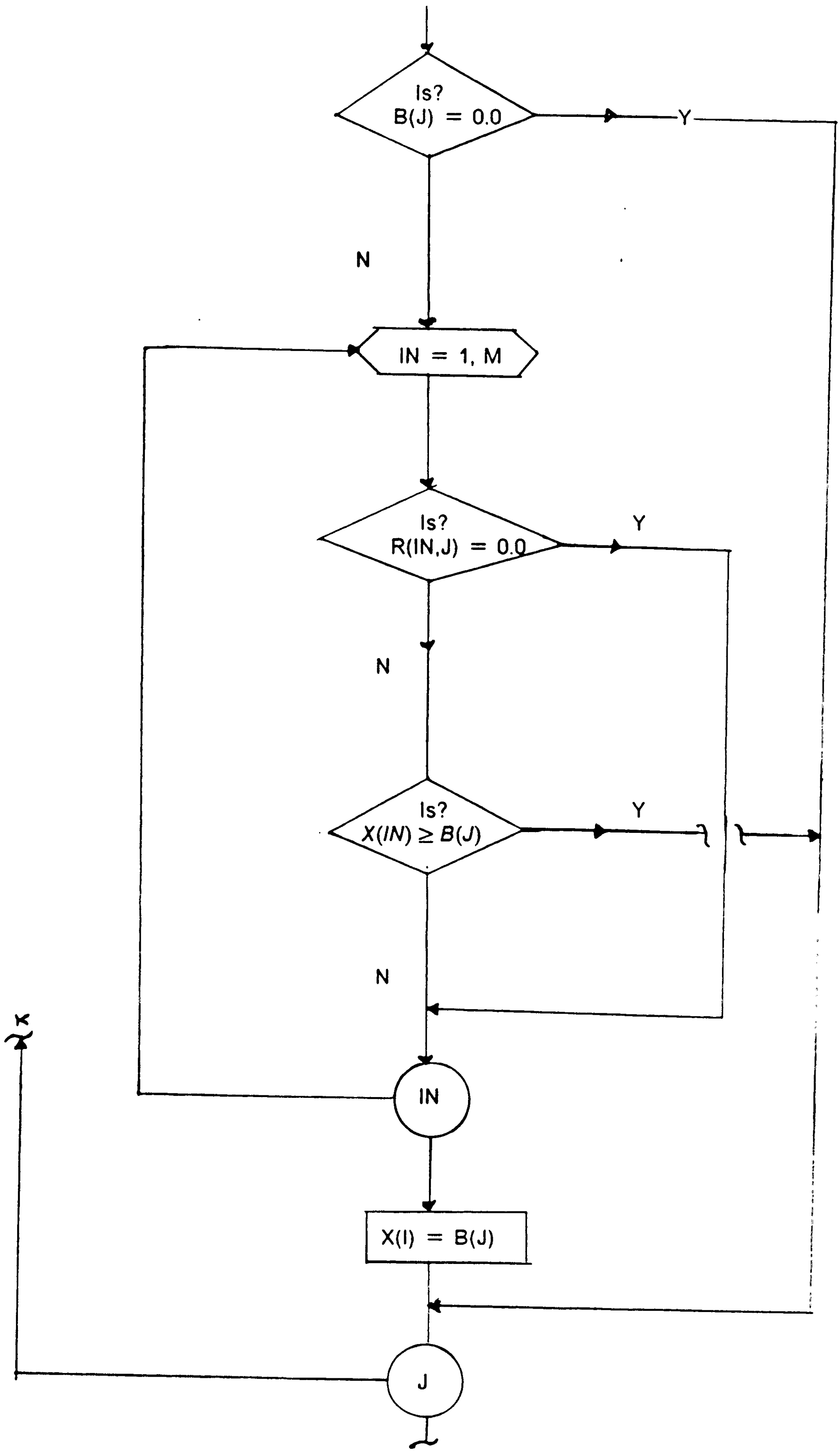

E
D
GLB(I) = MAX [MIN (R(I,J), B(J)]
J
I
I = 1, M
?
GLB(I) ≤ UB(I)
N
GLB(I) =UB(I)
Y
Output greatest
lower bound for A(I)
I

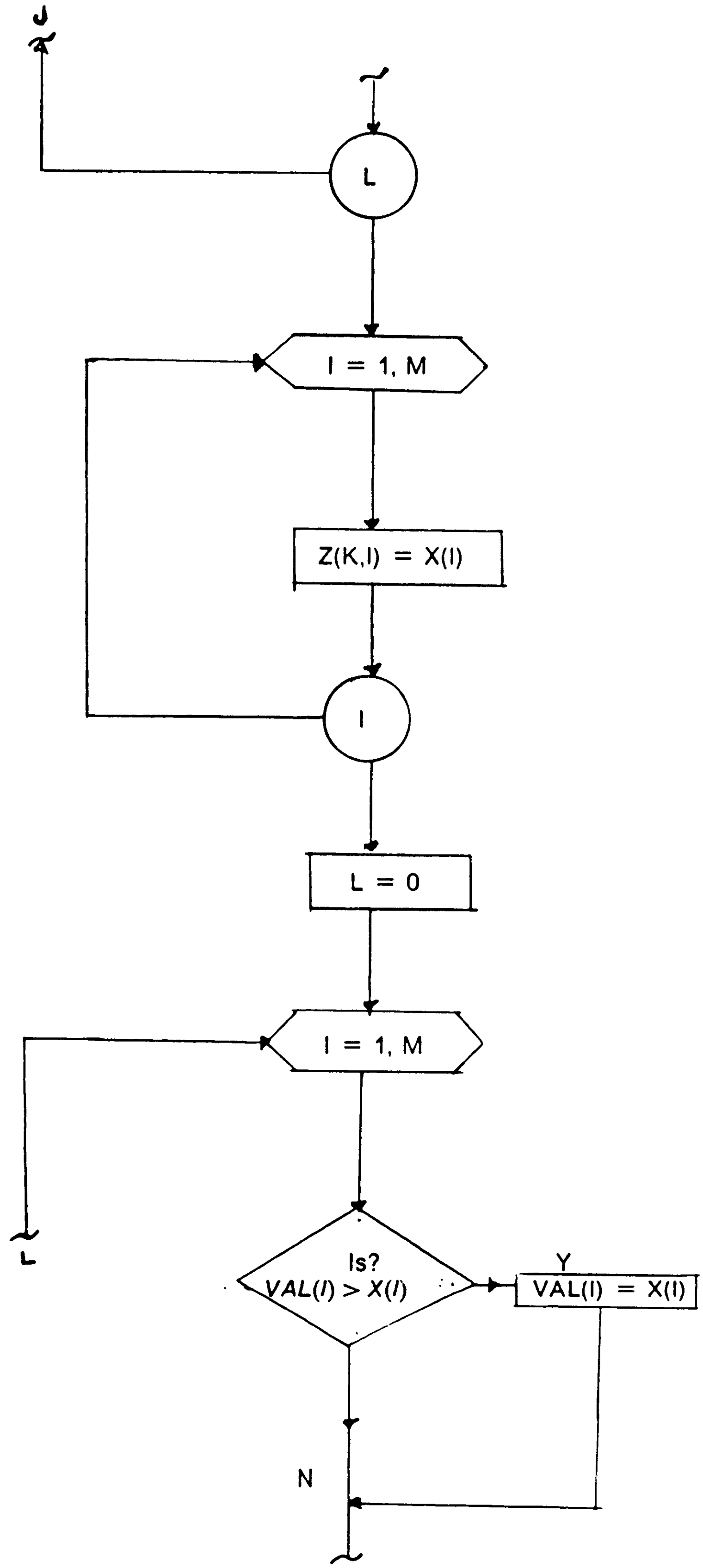

MODE = 0
J = 1, N
Is?
B(J) = 0.0
Y
N
E
I = 1, M
Is?
R(I,J) = 0.0
Y
N
F
Is?
GLB(I) ≥ B(J)
Y
N
G
H

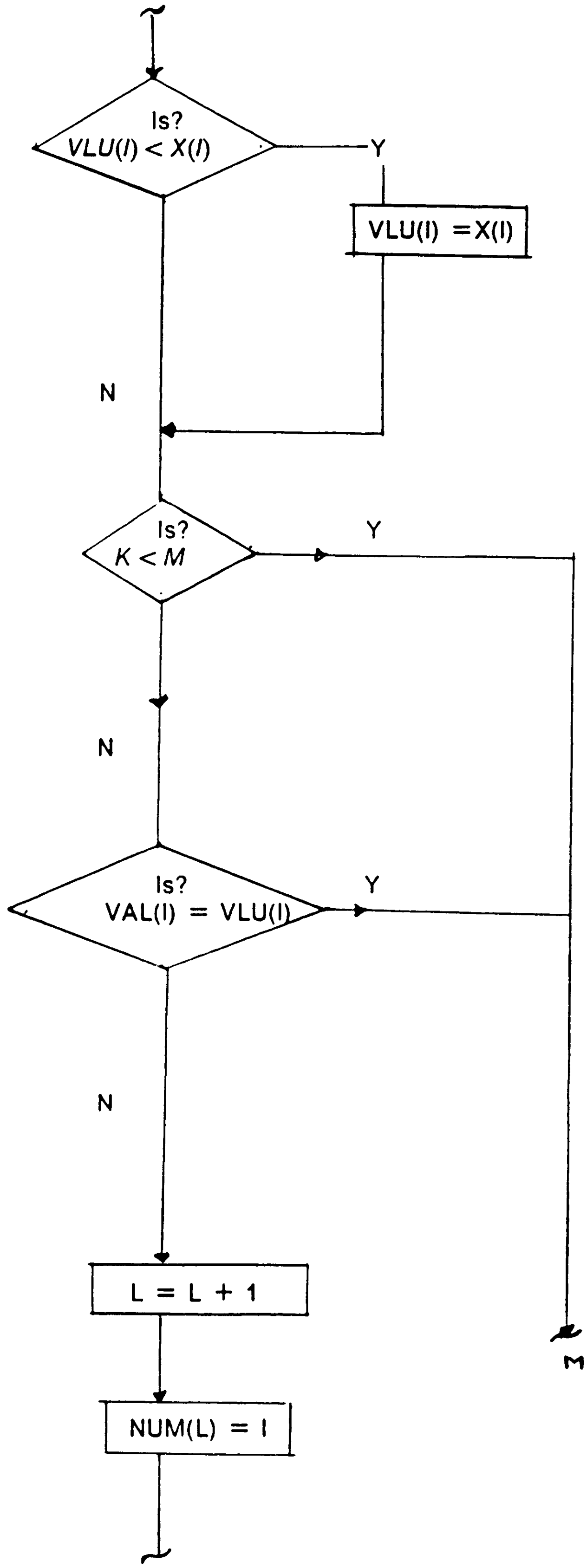

F
G
H
I
Flag:
B(J) can not
be supported
E
MODE = MODE + 1
J
Is?
MODE > 0
Y
Flag:
No solution
Exists
N
Flag:
Solution exists
Initialize VAL(I) = 1.0
VLU(I) = 0.0, for all I



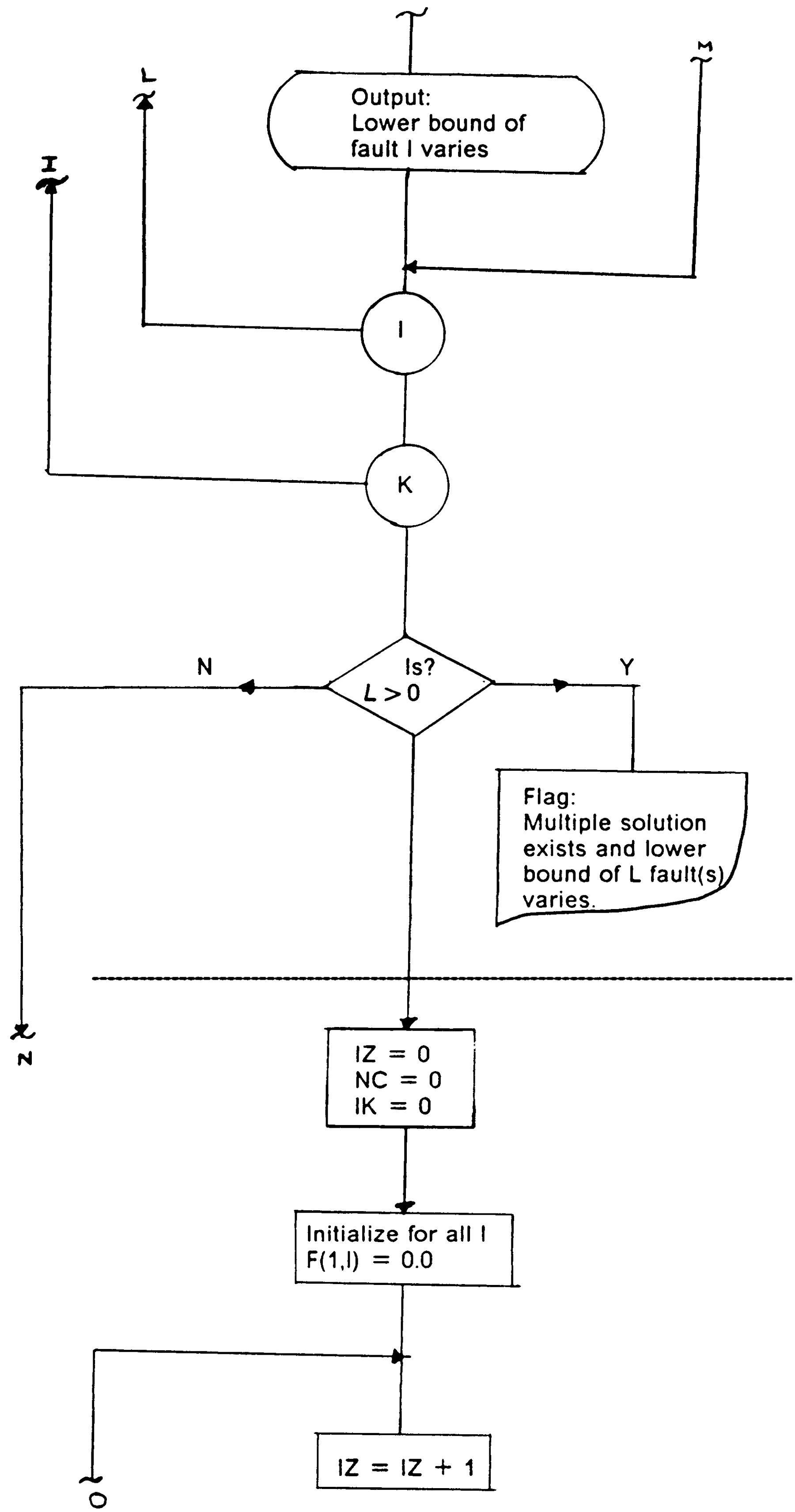
K = 1, M
Initialize X(I) = GLB(I)
L = 1, M
I = K + L - 1
I
J
Is?
I > M
Y
N
I = I - M
X(I) = 0.0
J = 1, N
K

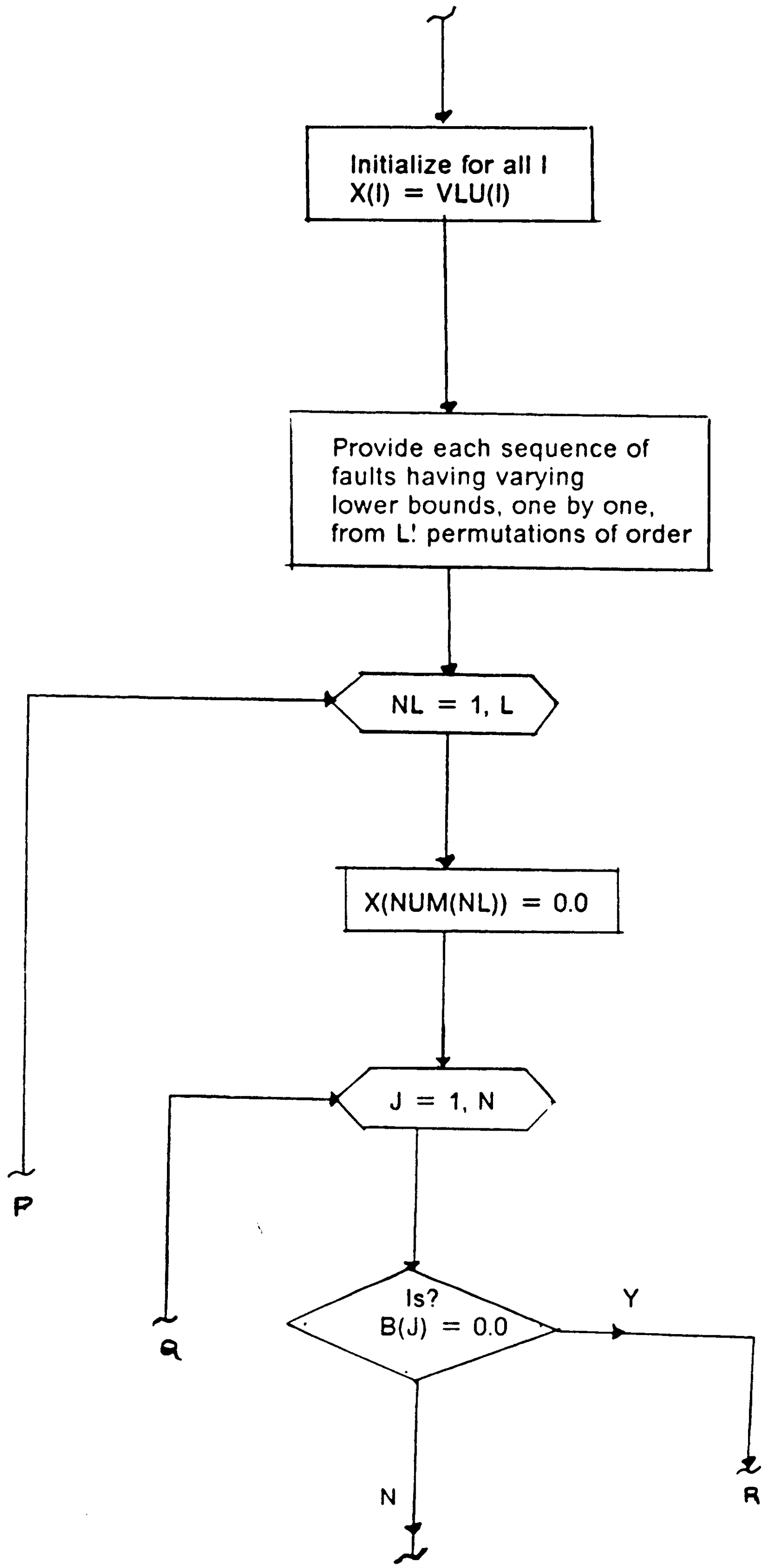

Is?
B(J) = 0.0
Y
N
IN = 1, M
Is?
R(IN,J) = 0.0
Y
N
Is?
X(IN) ≥ B(J)
Y
N
IN
X(I) = B(J)
J

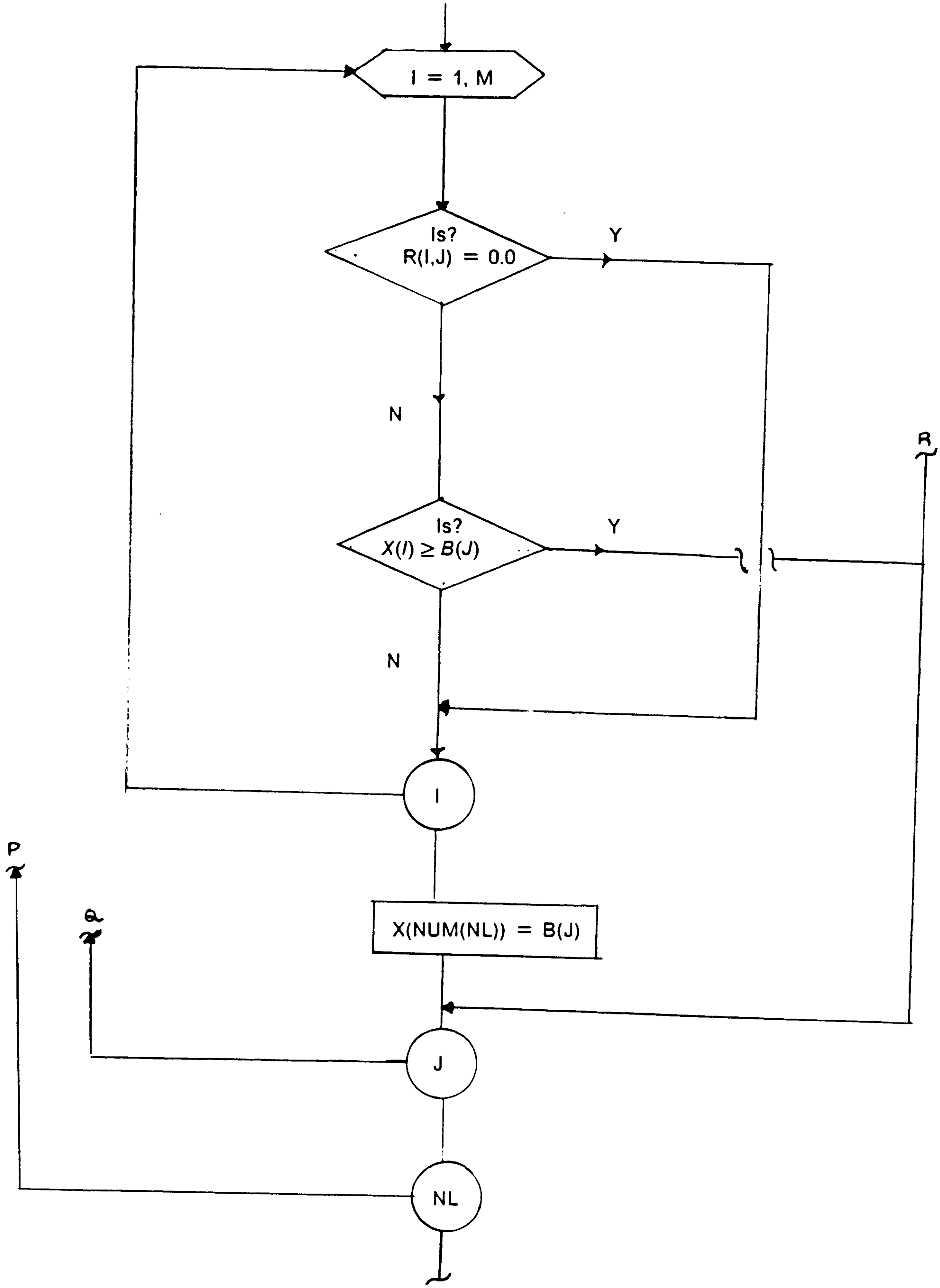

J
L
I = 1, M
Z(K,I) = X(I)
I
L = 0
I = 1, M
L
Is?
VAL(I) > X(I)
Y
VAL(I) = X(I)
N

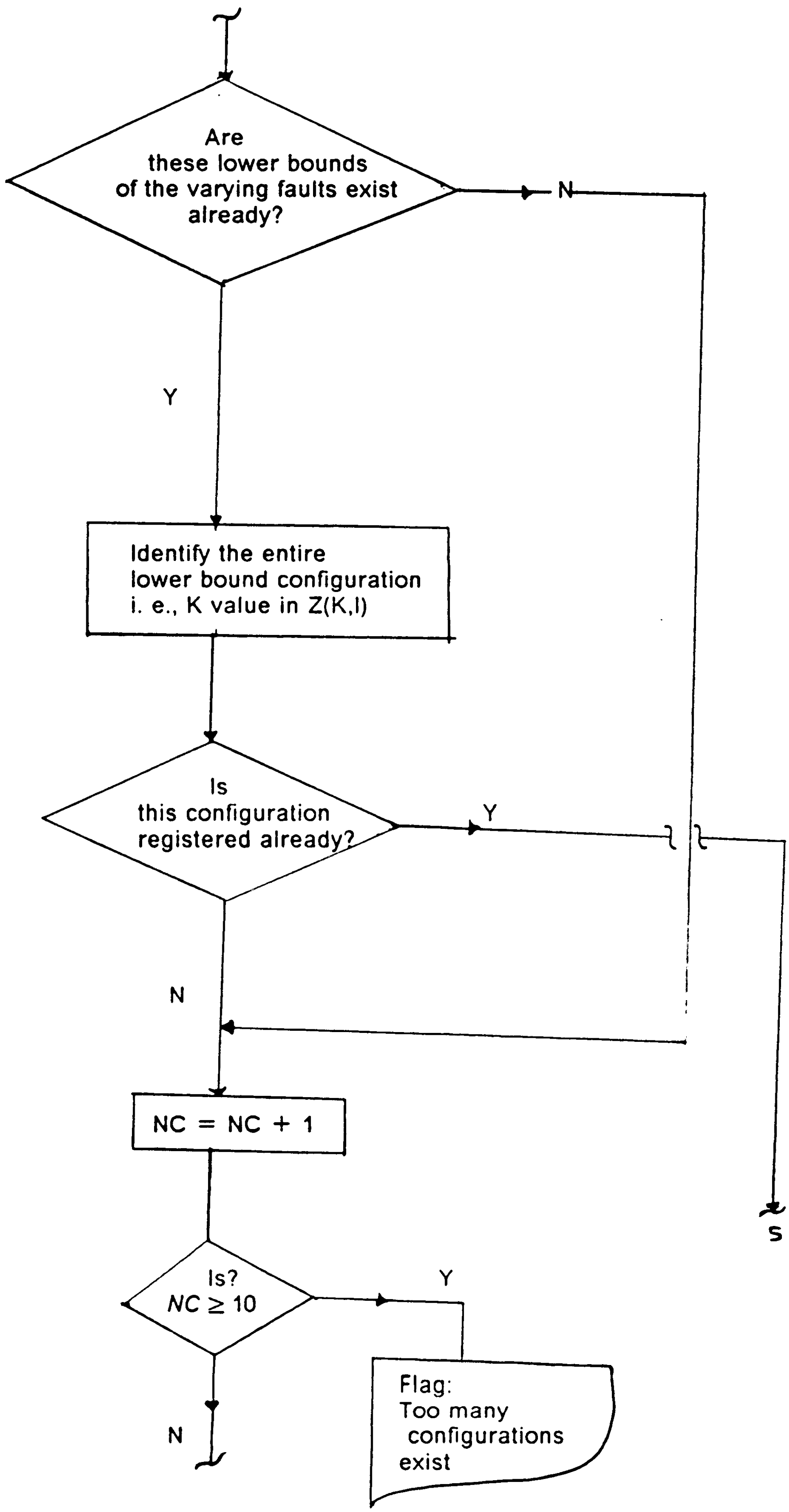


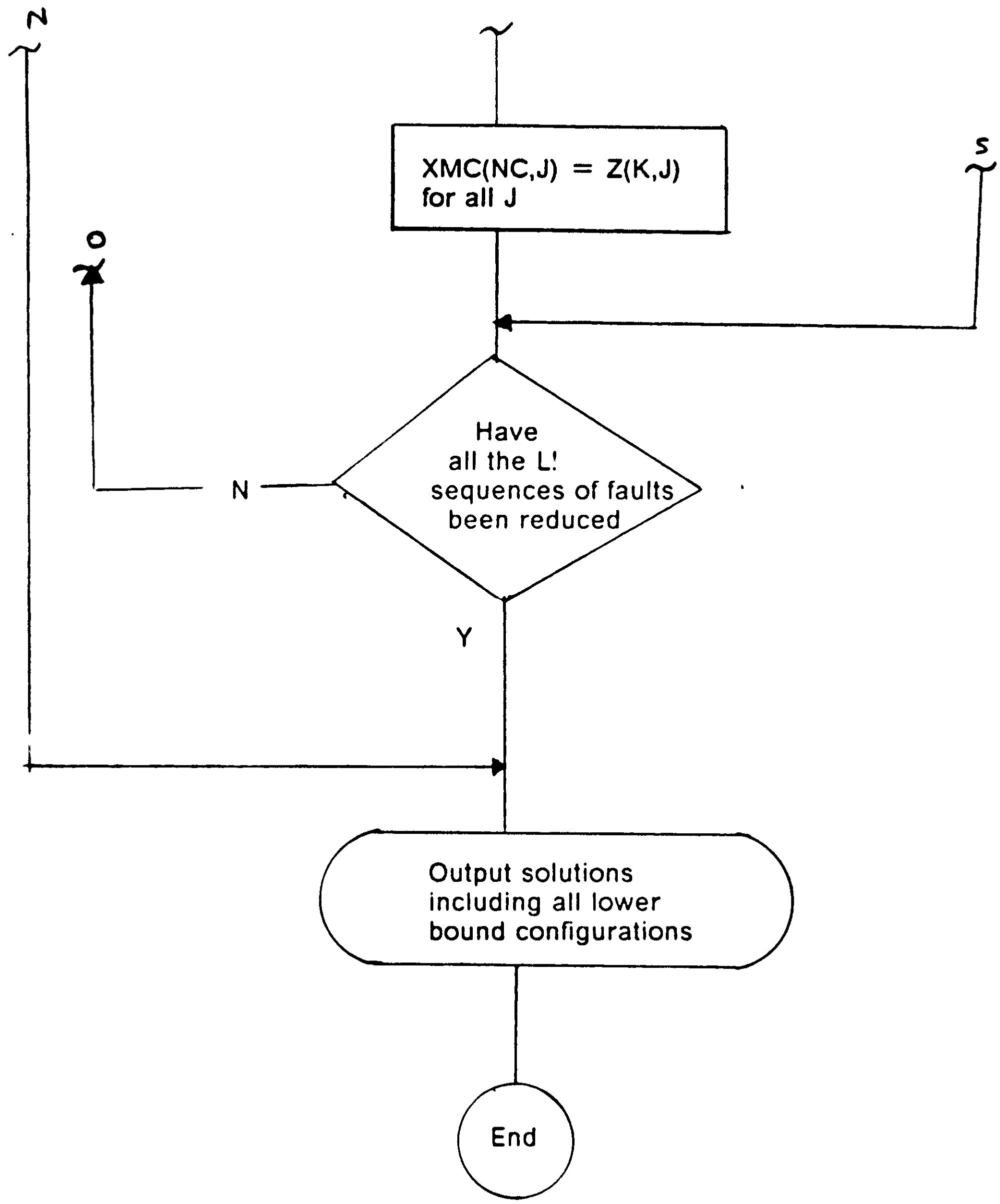
Is? VLU(I) < X(I)
Y
VLU(I) = X(I)
N
Is? K < M
Y
N
Is? VAL(I) = VLU(I)
Y
N
L = L + 1
M
NUM(L) = I


Output:
Lower bound of
fault I varies
I
K
Is?
$L>0$
N
Y
Flag:
Multiple solution
exists and lower
bound of L fault(s)
varies.
IZ = 0
NC = 0
IK = 0
Initialize for all I
F(1,I) = 0.0
IZ = IZ + 1


Initialize for all I
X(I) = VLU(I)
Provide each sequence of faults having varying lower bounds, one by one, from L! permutations of order
NL = 1, L
X(NUM(NL)) = 0.0
J = 1, N
P
Q
Is?
B(J) = 0.0
Y
N
R


I = 1, M
Is?
R(I,J) = 0.0
Y
N
R
Is?
X(I) ≥ B(J)
Y
N
I
P
Q
X(NUM(NL)) = B(J)
J
NL


Are
these lower bounds
of the varying faults exist
already?
N
Y
Identify the entire
lower bound configuration
i. e., K value in Z(K,I)
Is
this configuration
registered already?
Y
N
NC = NC + 1
S
Is?
NC ≥ 10
Y
Flag:
Too many
configurations
exist
N


Z
S
O
XMC(NC,J) = Z(K,J)
for all J
Have
all the L!
sequences of faults
been reduced
N
Y
Output solutions
including all lower
bound configurations
End

Appendix B. Computer Program

```
      DIMENSION R(30,30),T(30,30),B(30),UL(30),X(30),GLB(30)
      DIMENSION NUM(30),VAL(30),VLU(30),Z(10,30),F(11,30),XN(100,10)
      DIMENSION XMC(10,30),SYMP(30),FAULT(30)
      LOGICAL FIRST,A
      CHARACTER C,O
      CHARACTER *64 FNAME
      CHARACTER *64 FNAME1
      DATA C /'N'/
      WRITE (*,1)
1     FORMAT (10X, '**************************************************************'
     +      ,/,10X,'*                FDSYS
     +      ,/,10X,'*         FAULT DIAGNOSTIC SYSTEM
     +      ,/,10X,'*              Tapas R. Kar
     +      ,/,10X,'*  Department of Mining & Mining Engineering
     +      ,/,10X,'*       Virginia Polytechnic Institute &
     +      ,/,10X,'*              State University.
     +      ,/,10X,'*               December 1989
     +      ,/,10X,'**************************************************************'
      WRITE (*,3)
3     FORMAT (1X,'PREPARATION OF DATA, ENTER OPTION 1: CREATE NEW FILE
     + ',/,'                          2: CHANGE EXISTING FILE
     + ',/,'                          3: USE EXISTING FILE ',/)
      READ(*,*)NOPT1
      IF ( NOPT1 .EQ. 1 ) THEN
        WRITE (*,4)
4       FORMAT (1X,'ENTER THE NAME OF THE INPUT FILE',/)
        READ(*,'(A)')FNAME
        OPEN (7,FILE = FNAME )
      ENDIF
      WRITE(*,5)
5     FORMAT (1X,//,' ENTER THE OUTPUT OPTION  1: PRINTER ',/,
     +      '                            2: FILE ', // )
      READ (*,*)NOPT2
      IF ( NOPT2 .EQ. 1 ) THEN
        OPEN (8, FILE = 'PRN')
      ELSE
       WRITE(*,6)
6      FORMAT (1X, 'ENTER OPTION 1: CREATE NEW FILE',/
     +      '                 2: USE EXISTING FILE NAME', //)
       READ(*,*)NOPT3
       IF ( NOPT3 .EQ. 1 ) THEN
         WRITE (*,7)
7        FORMAT (1X,'ENTER THE NAME OF THE OUTPUT FILE',/ )
         READ (*,'(A)')FNAME1
         OPEN (8, FILE = FNAME1 )
      ENDIF
      ENDIF
      IF ( NOPT1 .EQ. 3 ) THEN
        GO TO 25
      ELSE
      WRITE (*,10)
10    FORMAT (1X,'ENTER THE NOS. OF FAULTS AND THE NOS OF SYMPTOMS',//)
```

```
      READ (*,*)M,N
      DO 11 I = 1,M
      WRITE (*,12)I
12     FORMAT (1X,'ENTER THE NAME OF FAULT',2X,I2,/)
      READ (*,'(A)') FAULT(I)
11     CONTINUE
      DO 13 I = 1,N
      WRITE (*,14)I
14     FORMAT (1X, 'ENTER THE NAME OF SYMPTOM',2X,I2,/)
      READ (*,'(A)')SYMP(I)
13     CONTONUE
      WRITE (*,15)
15     FORMAT (1X,'TO FORM THE OBSERVATION MATRIX SELECT THE RELATIONSHIP
     + BETWEEN FAULTS AND SYMPTOMS BASED ON THE GRADE OF OCCURRENCE OF
     + THE SYMPTOMS CAUSED BY THE FAULTS',/,
     + '          1: VERY OFTEN OCCURS',/,
     + '          2: OFTEN OCCURS',/,
     + '          3: QUITE OFTEN OCCURS',/,
     + '          4: SOMETIMES OCCURS',/,
     + '          5: SCARCELY OCCURS',/,
     + '          6: VERY SCARCELY OCCURS',/,
     + '          7: DOES NOT OCCUR',//, )
      DO 16 I = 1,M
        DO 17 J = 1,N
        WRITE (*,18)I,J
18       FORMAT(1X,'ENTER THE RELALATIONSHIP FROM THE ABOVE BETWEEN
     +   FAULT',1X,I2,'AND SYMPTOM',1X,I2,/)
        READ(*,*)NOPT4
        IF ( NOPT4 .EQ. 1 ) THEN
           T(I,J) = 0.9
         ELSE IF ( NOPT4 .EQ. 2 ) THEN
           T(I,J) = 0.7
         ELSE IF ( NOPT4 .EQ. 3 ) THEN
           T(I,J) = 0.6
         ELSE IF ( NOPT4 .EQ. 4 ) THEN
           T(I,J) = 0.5
         ELSE IF ( NOPT4 .EQ. 5 ) THEN
           T(I,J) = 0.3
         ELSE IF ( NOPT4 .EQ. 6 ) THEN
           T(I,J) = 0.2
      ENDIF
17     CONTINUE
16     CONTINUE
15     CONTINUE
      WRITE (7,*) M,N, ( ( T(I,J), J = 1,N ), I = 1,M )
      REWIND 7
      ENDIF
25     READ (7,*)M,N,( ( T(I,J), J = 1,N ), I = 1,M )
      DO 21 I =1,M
        DO 22 J =1,N
        IF ( T(I,J).GT. 0.0 .AND. T(I,J) .LE. 1.0 ) THEN
        R(I,J) = 1.0
      ELSE
        R(I,J) = 0.0
```

```
      END IF
22     CONTINUE
21     CONTINUE
      WRITE (*,26)
26     FORMAT (1X,'FAILURES',/ )
      DO 27 I = 1,M
      WRITE (*,28)I,FAULT(I)
28     FORMAT (1X,I2,2X,A30,/ )
27     CONTINUE
      WRITE (*,29)
29     FORMAT (1X,'SYMPTOMS',/ )
      DO 30 J = 1,N
      WRITE(*,31)J,SYMP(J)
31     FORMAT (1X,I2,2X,A30,/ )
30     CONTINUE
      WRITE (*,32)
32     FORMAT (1X,'ENTER THE OBSERVED SYMPTOMS VALUES, B(J)', BASED ON
     + THE DEGREE TO WHICH IT IS OBSERVED', /,
     + '           1: FOUND WITH ALMOST CERTAINTY',/
     + '           2: FOUND WITH LESS CERTAINTY',/
     + '           3: RECOGNIZED STRONGLY',/
     + '           4: RECOGNIZED LESS STRONGLY',/
     + '           5: FELT STRONGLY',/
     + '           6: FELT LESS STRONGLY',/
     + '           7: SUSPECTED',/ )
      DO 33 J = 1,N
      WRITE (*,34)J
34    FORMAT (1X,'ENTER THE DEGREE TO WHICH THE SYMPTOM',1X,I2,1X,'IS
     + OBSERVED',/)
      READ (*,*) NOPT5
      IF ( NOPT5 .EQ. 1 ) THEN
           B(J) = 0.8
         ELSE IF ( NOPT5 .EQ. 2 ) THEN
           B(J) = 0.7
         ELSE IF ( NOPT5 .EQ. 3 ) THEN
           B(J) = 0.6
         ELSE IF ( NOPT5 .EQ. 4 ) THEN
           B(J) = 0.5
         ELSE IF ( NOPT5 .EQ. 5 ) THEN
           B(J) = 0.4
         ELSE IF ( NOPT5 .EQ. 6 ) THEN
           B(J) = 0.3
         ELSE IF ( NOPT5 .EQ. 7 ) THEN
           B(J) = 0.2
      ENDIF
33     CONTINUE
      WRITE (8,35)
35     FORMAT (1X,'****************************************************
     +*************',/)
      DO 40 I = 1,M
        UL(I) = 1.0
        DO 50 J = 1,N
          TEMP = B(J) + 1.0 - T(I,J)
          IF ( UL(I).GT.TEMP ) UL(I) = TEMP
50        CONTINUE
```

```
      WRITE (8,60)I,UL(I)
60     FORMAT (1X,'UPPER BOUND FOR FAULT',1X,I2,1X,'IS =',F4.2,/)
40     CONTINUE
      WRITE (8,62)
62     FORMAT (1X,'**************************************************
     +************')
      DO 70 I = 1,M
      GLB(I) = 0.0
        DO 80 J = 1,N
        IF ( R(I,J) .EQ. 0.0 ) GO TO 80
        IF ( GLB(I) .GE. B(J) ) GO TO 80
        GLB(I) = B(J)
80     CONTINUE
      WRITE (8,75) I,GLB(I)
75     FORMAT ( 2X,' G.L.B. OF THE FAULT',1X,I2,1X,'IS =',1X,F4.2,/)
70     CONTINUE
      WRITE(8,74)
74     FORMAT (1X,'**************************************************
     +************',/ )
      WRITE (8,85)
85     FORMAT ( 2X, 'GRETAEST LOWER BOUND OF FAULT AFTER REDUCTION TO ME
     +ET THE UPPER LIMIT',/ )
      DO 90 I = 1,M
      IF ( UL(I) .LT. GLB(I) ) THEN
        GLB(I) = UL(I)
      ENDIF
100    WRITE (8,105) I,GLB(I)
105    FORMAT ( 2X,'G.L.B. OF THE FAULT',1X,I2,1X,'IS =',1X,F4.2,/)
90     CONTINUE
      WRITE (8,107)
107    FORMAT(1X,'**************************************************
     +*************',/)
      MODE = 0
      DO 110 J = 1,N
      IF ( B(J) .EQ. 0.0 ) GO TO 110
        DO 120 I = 1,M
        IF ( R(I,J) .EQ. 0.0 ) GO TO 120
        IF ( B(J) .LE. GLB(I) ) GO TO 110
120    CONTINUE
      WRITE (8,125)J
125    FORMAT (2X,'CONDITIONS FOR SYMPTOM',1X,I2,1X,'CANNOT BE MET',/)
      MODE = MODE + 1
110    CONTINUE
      IF ( MODE .GT. 0 ) THEN
       WRITE (8,127)MODE
127     FORMAT (1X,'CONDITIONS FOR',1X,I2,1X,'SYMPTOMS CANNOT BE MET',/)
       GO TO 260
      ELSE
      WRITE (8,130)
130    FORMAT (1X,'THIS PROBLEM HAS AT LEAST ONE SOLUTION',/)
      ENDIF
      DO 135 I = 1,M
      VAL(I) = 1.0
      VLU(I) = 0.0
135    CONTINUE
      DO 140 K = 1,M
```

```
      DO 145 I = 1,M
      X(I) = GLB(I)
145   CONTINUE
      DO 150 L = 1,M
      I = K + L - 1
      IF ( I .GT. M ) I = I - M
      X(I) = 0.0
      DO 155 J = 1,N
      IF ( B(J) .EQ. 0.0 ) GO TO 155
      DO 160 IN = 1,M
      IF ( R(IN,J) .EQ. 0.0 ) GO TO 160
      IF ( X(IN) .GE. B(J) ) GO TO 155
160   CONTINUE
      X(I) = B(J)
155   CONTINUE
150   CONTINUE
      L = 0
      DO 151 I = 1,M
      Z(K,I) = X(I)
151   CONTINUE
      DO 152 I = 1,M
      IF ( VAL(I) .GT. X(I) ) VAL(I) = X(I)
      IF ( VLU(I) .LT. X(I) ) VLU(I) = X(I)
      IF ( K .LT. M ) GO TO 152
      IF ( VAL(I) .NE. VLU(I) ) THEN
        L = L + 1
        NUM(L) = I
        WRITE (8,154)I
154     FORMAT ( 2X,'LOWER BOUND OF FAULT',1X,I2,1X,'VARIES',/ )
      ENDIF
152   CONTINUE
140   CONTINUE
      IF ( L .EQ. 0 ) THEN
        DO 141 I = 1,M
        WRITE (8,142)I,VAL(I),UL(I)
142     FORMAT (1X,'FAULT',1X,I2,1X,'HAS A LOWER BOUND OF',1X,F4.2,
     +  1X,'AND AN UPPER BOUND OF',1X,F4.2,/)
141     CONTINUE
        GO TO 260
      ELSE
        WRITE (8,165)L
165     FORMAT (2X,'LOWER BOUND(S) OF ',1X,I2,1X,'FAULT(S) VARY',/ )
      ENDIF
      WRITE (8,168)
168   FORMAT (1X,'**********************************************************
     +*************',/)
      FIRST = .FALSE.
      IZ = 0
      IK = 1
      NC = 0
        DO 169 I = 1,M
          F(IK,I) = 0.0
169   CONTINUE
167   IZ = IZ + 1
      DO 170 I = 1,M
```

```
      X(I) = VLU(I)
170   CONTINUE
      CALL IPRMER ( NUM,L,FIRST )
      DO 175 NL = 1,L
      X( NUM(NL) ) = 0.0
      DO 180 J = 1,N
      IF ( B(J) .EQ. 0.0 ) GO TO 180
        DO 185 I = 1,M
        IF ( R(I,J) .EQ. 0.0 ) GO TO 185
        IF ( X(I) .GE. B(J) ) GO TO 180
185   CONTINUE
      X ( NUM(NL) ) = B(J)
180   CONTINUE
175   CONTINUE
      DO 177 NL = 1,L
        XN( IZ, ( NUM(NL) ) ) = X ( NUM(NL) )
        IF ( IK .GT. 1 ) F(IK,NL) = X ( NUM(NL) )
177   CONTINUE
      DO 192 I = 1,M
        DO 195 NL = 1,L
         IF ( XN ( IZ, (NUM(NL))) .NE. Z( I,( NUM(NL) ) ) )GO TO 192
195   CONTINUE
      NT = 0
      DO 198 NK = 1,IK
        DO 202 NL = 1,L
        IF ( XN ( IZ, (NUM(NL)) ) .EQ. F(IK,( NUM(NL) ) ) ) NT = NT+1
202   CONTINUE
198   CONTINUE
          IF ( NT .NE. (IK*L) ) THEN
           NC = NC + 1
           DO 201 KJ = 1,M
                F(IK,KJ) = Z(I,KJ)
                XMC(NC,KJ) = Z(I,KJ)
201               CONTINUE
               IK = IK + 1
               GO TO 197
          ENDIF
192   CONTINUE
197   IF ( NC .GE. 10 ) THEN
        WRITE(8,199)
199      FORMAT(1X,'THERE ARE TOO MANY LOWER BOUNDS ARRAYS.',/)
        GO TO 260
      ENDIF
      IF ( FIRST ) GO TO 167
      WRITE (8,205)NC
205   FORMAT (2X,I2,1X,'CONFIGURATION(S) OF LOWER BOUND(S) EXIST',/ )
      WRITE (8,207)
207   FORMAT (1X,'**************************************************
     +************',/)
      DO 210 J = 1,NC
      DO 215 I = 1,M
      WRITE (8,220)I,XMC(J,I),UL(I)
220   FORMAT ( 2X,'FAULT',1X,I2,1X,'HAS A LOWER BOUND OF',1X,F4.2,1X,
     A 'AND AN UPPER BOUND OF',1X,F4.2,/ )
215   CONTINUE
```

```
      WRITE(8,217)
217   FORMAT (1X,'**************************************************
     +***********',/)
210   CONTINUE
260   STOP
      END

C     ***********************************************************
      SUBROUTINE IPRMER ( IPRM,M,FIRST )
      DIMENSION IPRM(M),NP(10),ND(10)
      LOGICAL FIRST
      IF ( FIRST ) GO TO 2

C     INITIATE REFERENCE ARRAYS FOR SUBSCRIPTING
      DO 1 K = 2,M
      NP(K) = 0
      ND(K) = 1
1     CONTINUE
      FIRST = .TRUE.
2     N = M
      K = 0
3     NP(N) = NP(N) + ND(N)
      NQ = NP(N)
      IF ( NQ .NE. N ) GO TO 4
      ND(N) = -1
      GO TO 5
4     IF ( NQ .NE. 0 ) GO TO 7
      ND(N) = 1
      K = K + 1
5     IF ( N .LE. 2 ) GO TO 6
      N = N - 1
      GO TO 3

C     ALL PERMUTATIONS HAVE BEEN FOUND
6     NQ = 1
      FIRST = .FALSE.

C     INTERCHANGE ELEMENTS
7     NQ = NQ + K
      TEMP = IPRM(NQ)
      IPRM(NQ) = IPRM(NQ+1)
      IPRM(NQ+1) = TEMP
      RETURN
      END
```

Appendix C. Program Output

Compressor

Case I:

```
************************************************************

UPPER BOUND FOR FAULT  1 IS = .30

UPPER BOUND FOR FAULT  2 IS = .10

UPPER BOUND FOR FAULT  3 IS = .50

UPPER BOUND FOR FAULT  4 IS = .90

UPPER BOUND FOR FAULT  5 IS = .10

UPPER BOUND FOR FAULT  6 IS = .50

UPPER BOUND FOR FAULT  7 IS = .50

UPPER BOUND FOR FAULT  8 IS = .30

UPPER BOUND FOR FAULT  9 IS = .30
************************************************************
  G.L.B. OF THE FAULT  1 IS =  .00

  G.L.B. OF THE FAULT  2 IS =  .80

  G.L.B. OF THE FAULT  3 IS =  .80

  G.L.B. OF THE FAULT  4 IS =  .80

  G.L.B. OF THE FAULT  5 IS =  .80

  G.L.B. OF THE FAULT  6 IS =  .30

  G.L.B. OF THE FAULT  7 IS =  .40

  G.L.B. OF THE FAULT  8 IS =  .80

  G.L.B. OF THE FAULT  9 IS =  .80
************************************************************

 GRETAEST LOWER BOUND OF FAULT AFTER REDUCTION TO MEET THE UPPER BOUND

 G.L.B. OF THE FAULT  1 IS =  .00

 G.L.B. OF THE FAULT  2 IS =  .10
```

G.L.B. OF THE FAULT 3 IS = .50

G.L.B. OF THE FAULT 4 IS = .80

G.L.B. OF THE FAULT 5 IS = .10

G.L.B. OF THE FAULT 6 IS = .30

G.L.B. OF THE FAULT 7 IS = .40

G.L.B. OF THE FAULT 8 IS = .30

G.L.B. OF THE FAULT 9 IS = .30

****

THIS PROBLEM HAS AT LEAST ONE SOLUTION

LOWER BOUND OF FAULT 6 VARIES

LOWER BOUND OF FAULT 9 VARIES

LOWER BOUND(S) OF 2 FAULT(S) VARY

****

2 CONFIGURATION(S) OF LOWER BOUND(S) EXIST

****

FAULT 1 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .30

FAULT 2 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .10

FAULT 3 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

FAULT 4 HAS A LOWER BOUND OF .80 AND AN UPPER BOUND OF .90

FAULT 5 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .10

FAULT 6 HAS A LOWER BOUND OF .30 AND AN UPPER BOUND OF .50

FAULT 7 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

FAULT 8 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .30

FAULT 9 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .30

****

FAULT 1 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .30

FAULT 2 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .10

FAULT 3 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

FAULT 4 HAS A LOWER BOUND OF .80 AND AN UPPER BOUND OF .90

FAULT 5 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .10

FAULT 6 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

FAULT 7 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

FAULT 8 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .30

FAULT 9 HAS A LOWER BOUND OF .30 AND AN UPPER BOUND OF .30

****

Case II:

****

UPPER BOUND FOR FAULT 1 IS = .70

UPPER BOUND FOR FAULT 2 IS = .10

UPPER BOUND FOR FAULT 3 IS = .10

UPPER BOUND FOR FAULT 4 IS = .30

UPPER BOUND FOR FAULT 5 IS = .50

UPPER BOUND FOR FAULT 6 IS = .50

UPPER BOUND FOR FAULT 7 IS = .30

UPPER BOUND FOR FAULT 8 IS = .30

UPPER BOUND FOR FAULT 9 IS = .80

****

G.L.B. OF THE FAULT 1 IS = .80

G.L.B. OF THE FAULT 2 IS = .00

G.L.B. OF THE FAULT 3 IS = .80

G.L.B. OF THE FAULT 4 IS = .00

G.L.B. OF THE FAULT 5 IS = .70

G.L.B. OF THE FAULT 6 IS = .70

G.L.B. OF THE FAULT 7 IS = .00

G.L.B. OF THE FAULT 8 IS = .00

G.L.B. OF THE FAULT 9 IS = .80

****

GRETAEST LOWER BOUND OF FAULT AFTER REDUCTION TO MEET THE UPPER BOUND

G.L.B. OF THE FAULT 1 IS = .70

G.L.B. OF THE FAULT 2 IS = .00

G.L.B. OF THE FAULT 3 IS = .10

G.L.B. OF THE FAULT 4 IS = .00

G.L.B. OF THE FAULT 5 IS = .50

G.L.B. OF THE FAULT 6 IS = .50

G.L.B. OF THE FAULT 7 IS = .00

G.L.B. OF THE FAULT 8 IS = .00

G.L.B. OF THE FAULT 9 IS = .80

****

THIS PROBLEM HAS AT LEAST ONE SOLUTION

FAULT 1 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .70

FAULT 2 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .10

FAULT 3 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .10

FAULT 4 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .30

FAULT 5 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

FAULT 6 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

FAULT 7 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .30

FAULT 8 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .30

FAULT 9 HAS A LOWER BOUND OF .80 AND AN UPPER BOUND OF .80

****

Longwall Shearer

****

UPPER BOUND FOR FAULT 1 IS = .50

UPPER BOUND FOR FAULT 2 IS = .50

UPPER BOUND FOR FAULT 3 IS = .30

UPPER BOUND FOR FAULT 4 IS = .50

UPPER BOUND FOR FAULT 5 IS = .80

UPPER BOUND FOR FAULT 6 IS = .10

UPPER BOUND FOR FAULT 7 IS = .50

****

G.L.B. OF THE FAULT 1 IS = .70

G.L.B. OF THE FAULT 2 IS = .80

G.L.B. OF THE FAULT 3 IS = .80

G.L.B. OF THE FAULT 4 IS = .80

G.L.B. OF THE FAULT 5 IS = .80

G.L.B. OF THE FAULT 6 IS = .80

G.L.B. OF THE FAULT 7 IS = .80

****

GRETAEST LOWER BOUND OF FAULT AFTER REDUCTION TO MEET THE UPPER BOUND

G.L.B. OF THE FAULT 1 IS = .50

G.L.B. OF THE FAULT 2 IS = .50

G.L.B. OF THE FAULT 3 IS = .30

G.L.B. OF THE FAULT 4 IS = .50

G.L.B. OF THE FAULT 5 IS = .80

G.L.B. OF THE FAULT 6 IS = .10

G.L.B. OF THE FAULT 7 IS = .50

****

THIS PROBLEM HAS AT LEAST ONE SOLUTION

LOWER BOUND OF FAULT 4 VARIES

LOWER BOUND OF FAULT 7 VARIES

LOWER BOUND(S) OF 2 FAULT(S) VARY

****

2 CONFIGURATION(S) OF LOWER BOUND(S) EXIST

****

FAULT 1 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

FAULT 2 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

FAULT 3 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .30

FAULT 4 HAS A LOWER BOUND OF .20 AND AN UPPER BOUND OF .50

FAULT 5 HAS A LOWER BOUND OF .80 AND AN UPPER BOUND OF .80

FAULT 6 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .10

FAULT 7 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

****

FAULT 1 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

FAULT 2 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

FAULT 3 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .30

FAULT 4 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .50

FAULT 5 HAS A LOWER BOUND OF .80 AND AN UPPER BOUND OF .80

FAULT 6 HAS A LOWER BOUND OF .00 AND AN UPPER BOUND OF .10

FAULT 7 HAS A LOWER BOUND OF .20 AND AN UPPER BOUND OF .50

****

The vita has been removed from the scanned document