Fuzzy Metagraph and Its Combination with the Indexing Approach in Rule-Based Systems

Zheng-Hua Tan, *Senior Member, IEEE*

Abstract—This paper presents a graph-theoretic construct called a fuzzy metagraph (FM) with the capability of describing the relationships between sets of fuzzy elements instead of only single fuzzy elements. The algebraic structure of FM and its properties are extensively investigated. Subsequently, the FM construct is applied to rule-based systems. First, we propose FM-based knowledge representation in both graphic and algebraic format. The representation is capable of identifying dependencies across compound propositions in the rules. In the algebraic representation, the FM closure matrix is considered a precompiled rule base enabling efficient query processing. An iterative approach is presented to facilitate the construction and expansion of the FM closure matrix, which is a key for real-world applications. Next, we introduce the concept of indexing, which was originally developed for information retrieval (IR), to enable an immediate extraction of relevant entries from the FM closure matrix. The indexing approach is applied in combination with the FM closure matrix. Based on the combination, corresponding inference mechanisms are introduced to achieve instant acquisition of relevant rules over a large collection of rules. The application in rule-based systems indicates that the combination of FM and IR techniques offers advantages for the mathematical analysis of systems.

Index Terms—Fuzzy graphs, knowledge representation, fuzzy reasoning, information retrieval, rule-based systems.

1 Introduction

In past years, much effort has been devoted to net-based mathematical modeling [1]. A net-based or graph-based approach provides a two-dimensional language suitable for visualization and a mathematical analysis scheme for the analysis of system structures [2]. Because of their role in reducing computations and improving visualization and problem understanding, graph and net theory have become important computational paradigms for representing and analyzing intelligent systems [1], [2], [3], [4].

Various net and graph models have been reported in the literature, e.g., Bayesian networks, digraphs, (fuzzy) Petri nets, (fuzzy) hypergraphs, and so forth [5]. In modeling and analyzing fuzzy systems, fuzzy graphs and nets are widely applied [6], [7], [8], [9]. Despite the new developments, graphical modeling remains a challenging and attractive research topic. For example, existing fuzzy graph techniques are not suitable for the analysis of the directed relationships between sets of elements, which widely exist in complicated fuzzy systems.

In crisp graphs, a number of new constructs that are geared toward set-to-set mappings have been proposed, e.g., directed hypergraphs [4], higraphs [10], and meta graphs [11], [12]. In particular, the metagraph has shown its capability of graphic presentation as well as its efficiency of algebraic analysis. Directed hypergraphs possess some similar properties to metagraphs. Both of them can be regarded as a combination of directed graphs and hypergraphs. Metagraphs, however, demonstrate a great potential for mathematical analysis.

Even though metagraphs have been widely applied in model management, data management, rule management, and hierarchical modeling, the metagraph construct is unable to tackle the issue of uncertainty and imprecision. Originally, metagraphs focused on structural aspects, i.e., the connectivity relationships among components of systems. To extend the approach to include process attributes such as time, attributed metagraphs have also been proposed [13]. But, since the attributes are logical and qualitative rather than quantitative, the extensions of metagraphs still cannot manipulate imprecise information. As a consequence, these approaches cannot support uncertain knowledge representation and approximate reasoning. However, most modes of human reasoning—and, especially, common sense reasoning—fall under this category [14]. We propose introducing fuzzy set operators to create a fuzzy metagraph (FM) as a generalization of the (crisp) metagraph. Subsequently, a number of concepts are developed and the properties of FM are investigated.

The FM is then applied to fuzzy rule-based systems for knowledge representation and reasoning. In the format of algebraic representation, the FM closure matrix is considered a precompiled rule base enabling efficient query processing. The fact that an FM closure matrix is often a sparse matrix, however, indicates that a method providing a quick extraction of entries from the matrix is a necessity. Inspired by the success of information retrieval (IR) techniques in Web-based queries, we introduce the concept of indexing for creating a compact representation of an FM-based rule base. Based on a combination of the indexing approach and the FM closure matrix, inference mechanisms are introduced to allow instant access to relevant rules over a large collection of rules.

- *The author is with the Department of Communication Technology, Aalborg University, Niels Jernes Vej 12, DK-9220, Aalborg, Denmark. E-mail: zt@kom.aau.dk.*

Manuscript received 14 Mar. 2005; revised 28 Sept. 2005; accepted 7 Dec. 2005; published online 20 Apr. 2006.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0092-0305.

The rest of this paper is organized as follows: Section 2 contains a brief review of fuzzy graphs and IR techniques. In addition, the motivation of this work is discussed. In Section 3, we define the proposed FM construct. Section 4 investigates the algebraic structure of FM. In Section 5, FM-based knowledge representation and the indexing table are presented and their implementation issues are addressed. Based on the FM closure matrix and the indexing table, corresponding inference mechanisms are developed in Section 6. Finally, conclusions are summarized in the last section.

2 Background and Motivation

This section briefly reviews fuzzy graphs and IR techniques, followed by a discussion of the motivation of this work.

2.1 Fuzzy Graphs

A graph is defined by a pair $G = (X, E)$, where $X = \{x_1, x_2 \dots x_n\}$ is a finite set of vertices and E a collection of edges that happen to connect these vertices. The edge set E graphically demonstrates a relation on $X \times X$.

The concept of a fuzzy graph is the "fuzzification" of the crisp graphs using fuzzy sets. A fuzzy graph $\tilde{G}$ can be defined as a triple $(X, \tilde{X}, \tilde{E})$, where $\tilde{X}$ is a fuzzy set on X and $\tilde{E}$ a fuzzy relation on $X \times X$ [15].

A fuzzy set $\tilde{X}$ on X is completely characterized by its membership function $\mu : X \to [0, 1]$. For each $x \in X$, $\mu(x)$ illustrates the truth value of the statement of "x belongs to $\tilde{X}$." A fuzzy edge set $\tilde{E}$ is defined as a function $\rho : E \to [0, 1]$. Therefore, a fuzzy graph can be described by two functions μ and ρ. For the sake of notational convenience, X is omitted and, thus, the notation $\tilde{G} = (\tilde{X}, \tilde{E})$ or $\tilde{G} = (\mu, \rho)$ is used in literature where both vertices and edges have membership values [16]. Often, the membership value of an edge is also called certainty factor (CF) of the edge. For simplicity, assign $\tilde{x}_i$ denoting $(x_i, \mu(x_i))$ and $\tilde{e}_k$ denoting (e_k, CF_k), i.e., $(e_k, \rho(e_k))$.

Over the past decades, a number of fuzzy graphs have been proposed to represent uncertain relationships between fuzzy elements or sets of fuzzy elements. However, as mentioned before, existing fuzzy graphs are not capable of effectively modeling the directed relationships between sets of fuzzy elements.

In understanding the value of metagraphs, [11] gives a detailed comparison between metagraphs and conventional graphs with illustrative examples. The commonly used fuzzy graphs share the following limitations with conventional graphs:

1. Albeit a fuzzy undirected graph can represent relationships existing between two variables, it cannot provide the direction of the relationships.
2. The input and output relationships between the element pairs can be described by the fuzzy directed graph. However, it cannot represent the relationships where there is more than one variable in the input and/or in the output.
3. Fuzzy hypergraph describes any fuzzy relationship as a set of fuzzy elements, but it is not able to distinguish the input variables from the outputs.
4. Using arcs to combine edges, a fuzzy AND/OR graph attempts to represent relationships even where there are more than one input and output variables. When describing the relationships between the sets of variables, however, there are too many combined edges for fuzzy AND/OR graph to distinguish them.

It is critical that all of the graphs lack powerful algebraic analysis methods for manipulating the directed relationships between sets of elements. This motivates the development of the FM to be defined in Section 3. The major distinction between FMs and traditional graph-theoretic constructs is that an FM describes the directed relationship between sets of elements instead of single elements. Each edge in the FM is an ordered pair of sets of elements. So, the edge is neither an ordered pair of elements in a fuzzy directed graph nor a disordered set of elements in a fuzzy hypergraph.

TABLE 1
An Example of Word-Level Inverted File [17]

Number	Term	<Documents; Words>
1	cold	<2; (1; 6), (4; 8)>
2	days	<2; (3; 2), (6; 2)>
3	hot	<2; (1; 3), (4; 4)>
4	in	<2; (2; 3), (5; 4)>
5	it	<2; (4; 3, 7), (5; 3)>

2.2 Information Retrieval and Indexing Approach

The field of IR has witnessed the great success of search engines such as Google and AlltheWeb, which are capable of providing users instant acquisition of requested information through more than three billion pages. For example, both Google and AlltheWeb can immediately answer questions like "What is the largest animal in the world?" The instant query answering is achieved by seeking matches between words in a query and words in the indexed pages rather than in the original ones. The heart of search engines is indexing, which is a process of converting a collection of documents into a form suitable for efficient search and retrieval. It is believed that the difficulty of searching is due to an inadequate index or no index at all [17].

The creation of an IR system typically includes three steps: gathering a collection of documents, preprocessing the collection, and indexing the preprocessed collection. Preprocessing and indexing are the two key processes. The goal of preprocessing is to construct a canonical form of documents by employing techniques such as tokenization, stop-word filtering, stemming, and so forth. To enable fast search for individual terms, an inverted indexing file is often stored [17], [18]. An inverted file typically comprises a lexicon and, for each word (or item) in the lexicon, an inverted file stores a list of pointers to all occurrences of that item in the main text. As a result, a document is represented by indexing terms. This sort of method is followed by almost all commercial IR systems [19].

Table 1 shows an example of word-level inverted file in which terms are listed in the second column and, for each term, the number of documents containing the term and the

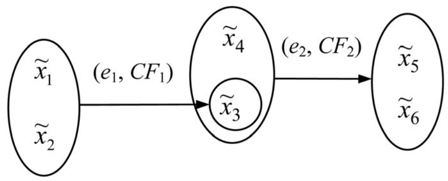

Fig. 1. Fuzzy metagraph with simple path.

positions where the term occurs within each document are demonstrated in the third column [17]. In reality, constructing the inverted file (index) can be a very challenging task. One commonly used approach called sort-based inversion creates an inverted file by the following procedure. First, documents are parsed to extract tokens (items), which are then saved with their corresponding document ID and positions. After all documents have been parsed, the inverted file is then sorted alphabetically. Finally, multiple term entries for a single document are merged.

The indexing technique motivates the development of FM-based knowledge representation where FM closure matrix is treated as a preprocessed rule base and indexing table is created to speed up the searching over the rule base.

3 Fuzzy Metagraphs

The fuzzy metagraph, a generalization of the metagraph concept proposed by Basu and Blanning [11], [12], can be defined as follows [20]:

Definition 1. *Consider a finite set $X = \{x_i, i = 1 \dots I\}$. A fuzzy metagraph is a triple $\tilde{S} = \{X, \tilde{X}, \tilde{E}\}$ in which $\tilde{X}$ is a fuzzy set on X and $\tilde{E}$ is a fuzzy edge set $\{\tilde{e}_k, k = 1...K\}$. Each component $\tilde{e}_k$ in $\tilde{E}$ is characterized by an ordered pair $\langle \tilde{V}_k, \tilde{W}_k \rangle$. In the pair, $\tilde{V}_k \subseteq \tilde{X}$ is the in-vertex of $\tilde{e}_k$ and $\tilde{W}_k \subseteq \tilde{X}$ is the out-vertex. The co-input of any $\tilde{x} \in \tilde{V}_k$ is $\tilde{V}_k\{\tilde{x}\}$ and the co-output of any $\tilde{x} \in \tilde{W}_k$ is $\tilde{W}_k\{\tilde{x}\}$.*

The symbol "\" in this paper is used for set difference.

Fig. 1 shows an FM whose element set is $X = \{x_1 \dots x_6\}$ and whose edge set consists of $\tilde{e}_1 =< \{\tilde{x}_1, \tilde{x}_2\}, \{\tilde{x}_3\} >$, and $\tilde{e}_2 =< \{\tilde{x}_3, \tilde{x}_4\}, \{\tilde{x}_5, \tilde{x}_6\} >$. The in-vertex and out-vertex of $\tilde{e}_1$, for example, are $\{\tilde{x}_1, \tilde{x}_2\}$ and $\{\tilde{x}_3\}$, respectively.

An important property of graphs is that of connectivity. In Fig. 1, there is a sequence of edges $\langle \tilde{e}_1, \tilde{e}_2 \rangle$ that connects $\tilde{x}_1$ to $\tilde{x}_5$, which means that a path from $\tilde{x}_1$ to $\tilde{x}_5$ exists. In order to infer $\tilde{x}_5$ from $\tilde{x}_1$, the values of $\tilde{x}_2$ and $\tilde{x}_4$ are all required at the same time. To represent this type of connectivity, a basic concept of FM called a simple path is defined as follows:

Definition 2. *Given a finite set X, an FM $\tilde{S} = \{X, \tilde{X}, \tilde{E}\}$ and two fuzzy elements $\tilde{b}, \tilde{c} \in \tilde{X}$, a simple path from $\tilde{b}$ to $\tilde{c}$ is a sequence of fuzzy edges $\tilde{P}(\tilde{b}, \tilde{c}) = \langle \tilde{f}_l, l = 1 \dots L \rangle$ with $\tilde{f}_l = \langle \tilde{V}_l, \tilde{W}_l \rangle \in \tilde{E}$ such that:*

1. $\tilde{b} \in \tilde{V}_1$, $\tilde{c} \in \tilde{W}_L$ *and*
2. $\tilde{W}_l \cup \tilde{V}_{l+1} \neq \phi$, *for* $l = 1 \dots L - 1$.

The element $\tilde{b}$ is the source of $\tilde{P}(\tilde{b}, \tilde{c})$ and $\tilde{c}$ is the target. The number L of edges in the path is the length of the path. The co-input of $\tilde{b}$ in $\tilde{P}(\tilde{b}, \tilde{c})$ is $[\cup_{l=1}^{L} \tilde{V}_l \backslash \cup_{l=1}^{L} \tilde{W}_l] \backslash \{\tilde{b}\}$, and the co-output of $\tilde{c}$ is $\cup_{l=1}^{L} \tilde{W}_l \backslash \{\tilde{c}\}$.

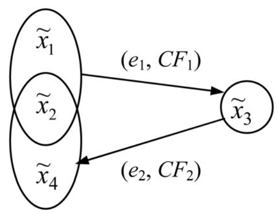

Fig. 2. Cyclic fuzzy metagraph.

In the FM in Fig. 1, for example, the sequence of edges $\langle \tilde{e}_1, \tilde{e}_2 \rangle$ is a simple path from $\tilde{x}_1$ to $\tilde{x}_5$ with a length of two. The co-input of $\tilde{x}_1$ can be calculated as

$$[\cup_{l=1}^{L} \tilde{V}_l \backslash \cup_{l=1}^{L} \tilde{W}_l] \backslash \{\tilde{b}\}$$
$$= [\{\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{x}_4\} \backslash \{\tilde{x}_3, \tilde{x}_5, \tilde{x}_6\}] \backslash \{\tilde{x}_1\} = \{\tilde{x}_2, \tilde{x}_4\}.$$

The co-output of $\tilde{x}_5$ is

$$\cup_{l=1}^{L} \tilde{W}_l \backslash \{\tilde{c}\} = \{\tilde{x}_3, \tilde{x}_4, \tilde{x}_6\} \backslash \{\tilde{x}_5\} = \{\tilde{x}_3, \tilde{x}_6\}.$$

Based on the path, $\tilde{x}_5$ can be calculated from $\tilde{x}_1$, provided $\tilde{x}_2$ and $\tilde{x}_4$ are also given. Simultaneously, $\tilde{x}_3$ and $\tilde{x}_6$ are figured out as by-product outputs. The above path $\langle \tilde{e}_1, \tilde{e}_2 \rangle$ can also be regarded as a simple path from $\tilde{x}_2$ to $\tilde{x}_5$, $\tilde{x}_2$ to $\tilde{x}_6$ or $\tilde{x}_1$ to $\tilde{x}_6$ with different co-inputs and co-outputs.

A simple path $\tilde{P}(\tilde{b}, \tilde{b})$ from an element to itself is a cycle. An FM with one or more cycles is cyclic and an FM without any cycles is acyclic. Fig. 2 illustrates a cyclic FM that contains two identical cycles $\tilde{P}(\tilde{x}_2, \tilde{x}_2) = < \tilde{e}_1, \tilde{e}_2 >$ and $\tilde{P}(\tilde{x}_3, \tilde{x}_3) = < \tilde{e}_2, \tilde{e}_1 >$.

In this paper, both relationships and elements (variables) are fuzzy or uncertain. Although an FM models the relationships and the corresponding CFs, it does not specify the truth values of elements. In other words, the truth values of elements in an FM are unknown parameters to be decided on-the-fly. The notation $\tilde{x}_i$ in the graph just indicates that x_i is a fuzzy variable and its truth value is to be determined by input or inference.

4 FM Adjacency Matrix and its Closure

An alternative approach to the representation and analysis of graphical structures is a matrix. In the following, we develop the algebraic structure for the FM, called an FM adjacency matrix, and a closure of the FM adjacency matrix.

4.1 FM Adjacency Matrix

Demonstrating all the direct linkages between the elements in a fuzzy set, the adjacency matrix A of an FM is a square matrix with rows and columns labeled by the elements in the fuzzy set. One of the usages of an FM adjacency matrix is to determine whether there is any edge with a length of one connecting an element in the row to one in the column. Each entry $a_{i,j}$ in the matrix is a null set or a few ordered triples according to whether $\tilde{x}_i$ and $\tilde{x}_j$ are adjacent or not and how many edges connect $\tilde{x}_i$ to $\tilde{x}_j$. In each triple, the first component is the co-input of $\tilde{x}_i$; the second is the co-output of $\tilde{x}_j$; the third is a simple path from $\tilde{x}_i$ to $\tilde{x}_j$ with a length of one.

Definition 3. *Given a finite set $X = \{x_i, i = 1 \dots I\}$ and an FM $\tilde{S} = \{X, \tilde{X}, \tilde{E}\}$ with $\tilde{E}$ being a fuzzy edge set*

TABLE 2
The Adjacency Matrix of the FM in Fig. 1

	$\tilde{x}_3$	$\tilde{x}_5$	$\tilde{x}_6$
$\tilde{x}_1$	$<\{\tilde{x}_2\},\phi,<\tilde{e}_1>>$	ϕ	ϕ
$\tilde{x}_2$	$<\{\tilde{x}_1\},\phi,<\tilde{e}_1>>$	ϕ	ϕ
$\tilde{x}_3$	ϕ	$<\{\tilde{x}_4\},\{\tilde{x}_6\},<\tilde{e}_2>>$	$<\{\tilde{x}_4\},\{\tilde{x}_5\},<\tilde{e}_2>>$
$\tilde{x}_4$	ϕ	$<\{\tilde{x}_3\},\{\tilde{x}_6\},<\tilde{e}_2>>$	$<\{\tilde{x}_3\},\{\tilde{x}_5\},<\tilde{e}_2>>$

$\{\tilde{e}_k, k = 1 \ldots K\}$, the FM adjacency matrix A of $\tilde{S}$ is an $I \times I$ matrix. For $i, j \in \{1 \ldots I\}$, each entry $a_{i,j}$ is defined as:

$$a_{i,j} = \cup_{k=1}^{K} \left(\alpha_{i,j}\right)_k,$$

where

$$\left(\alpha_{i,j}\right)_k = \begin{cases} \langle \tilde{V}_k \backslash \{\tilde{x}_i\}, \tilde{W}_k \backslash \{\tilde{x}_j\}, \langle \tilde{e}_k \rangle \rangle & \text{if both } \tilde{x}_i \in \tilde{V}_k \\ & \text{and } \tilde{x}_j \in \tilde{W}_k, \\ \phi & \text{otherwise.} \end{cases}$$

The FM adjacency matrix of the FM in Fig. 1 can be illustrated as Table 2. Strictly speaking, this FM adjacency matrix should be a matrix of 6×6. But, in this paper, the null entries are omitted to make the table as concise as possible. Each nonempty entry $a_{i,j}$ consists of one or more triples, one for each path of length one connecting $\tilde{x}_i$ to $\tilde{x}_j$. For instance, $a_{1,3} = <\{\tilde{x}_2\}, \phi, <\tilde{e}_1>>$ indicating that a path $<\tilde{e}_1>$ connects $\tilde{x}_1$ to $\tilde{x}_3$ with co-input $\{\tilde{x}_2\}$ and co-output ϕ. Since there is no path of length one connecting $\tilde{x}_1$ to $\tilde{x}_5$, $a_{1,5} = \phi$.

4.2 The Closure of FM Adjacency Matrix

The FM adjacency matrix only shows the adjacent linkages in the graph. There may be many other paths existing, but not visible from the FM adjacency matrix. For example, if $\tilde{x}_i$ is adjacent to $\tilde{x}_j$ and $\tilde{x}_j$ adjacent to $\tilde{x}_k$, then it can be inferred that there is a path with a length of two from $\tilde{x}_i$ to $\tilde{x}_k$. The closure of FM adjacency matrix (FM closure matrix for short) is developed to disclose all paths of any length connecting two arbitrary vertices $\tilde{x}_i$ and $\tilde{x}_j$, if any exist.

The FM closure matrix A^* is formed by adding successive powers of the FM adjacency matrix, namely, the multiplication by itself. Obviously, the addition and multiplication operations of adjacency matrices are essential for computing the FM closure matrix, which are defined in the appendix. The square, A^2, of the adjacency matrix of the FM in Fig. 1 is given in Table 3. Each nonempty entry $a^2_{i,j}$ consists of one or more triples, one for each path of length two connecting $\tilde{x}_i$ to $\tilde{x}_j$.

TABLE 3
The Square of the Adjacency Matrix of the FM in Fig. 1

	$\tilde{x}_5$	$\tilde{x}_6$
$\tilde{x}_1$	$<\{\tilde{x}_2,\tilde{x}_4\},\{\tilde{x}_3,\tilde{x}_6\},<\tilde{e}_1,\tilde{e}_2>>$	$<\{\tilde{x}_2,\tilde{x}_4\},\{\tilde{x}_3,\tilde{x}_5\},<\tilde{e}_1,\tilde{e}_2>>$
$\tilde{x}_2$	$<\{\tilde{x}_1,\tilde{x}_4\},\{\tilde{x}_3,\tilde{x}_6\},<\tilde{e}_1,\tilde{e}_2>>$	$<\{\tilde{x}_1,\tilde{x}_4\},\{\tilde{x}_3,\tilde{x}_5\},<\tilde{e}_1,\tilde{e}_2>>$

Then, the FM closure matrix is defined as follows:

Definition 4. *Given an FM $\tilde{S} = \{X, \tilde{X}, \tilde{E}\}$ and its FM adjacency matrix A, the FM closure matrix is defined as an infinite sum, namely, $A^* = A + A^2 + A^3 + \ldots + A^n, n \to \infty$.*

It can be proven that the limit exists and

$$A^* = \sum_{n=1}^{\infty} A^n = \sum_{n=1}^{K} A^n,$$

where K is the overall number of edges in the FM. In fact, each triple in a nonempty entry in A represents a path with a length of one, each in A^2 represents a path with a length of two, each in A^3 represents a path with a length of three, and so forth. When $n > K$, A^n may have nonempty entries, but only along their diagonal and only for elements that are members of a cycle. The truncation operator $Trnc()$ used in matrix multiplication (see the Appendix), however, removes the repetition of an edge and, thus, limits triples in nonempty entries along diagonal to the length of cycles. Therefore, all information about paths and cycles is revealed by the first K powers of A [12], [21]. In other word, all $A^n (n > K)$ have no contribution to the sum A^*.

The FM closure matrix for the FM in Fig. 1 is given in Table 4, which discloses paths of all lengths in the FM. The FM closure matrix for the cyclic FM in Fig. 2 is given in Table 5. It is noticed that entries $a^*_{2,2}$ and $a^*_{3,3}$ of the diagonal are nonempty, indicating that two identical cycles are passing through $\tilde{x}_2$ and $\tilde{x}_3$. Since an FM is acyclic if and only if all the diagonal entries of A^* are null, the FM closure matrix is also capable of identifying cycles. The issue of cycles in FM will not be further addressed in this paper.

A normal transitive closure matrix of a graph is a 0-1 matrix $M = [m_{ij}]$, $m_{ij} = 1$, if there is a path from vertex i to vertex j; 0, otherwise. By describing all the paths, co-inputs and co-outputs between any two elements, the FM closure matrix distinguishes the FM from other fuzzy graphs and can be utilized as a preprocessed rule base. As the focus of

TABLE 4
The Closure of the Adjacency Matrix of the FM in Fig. 1

	$\tilde{x}_3$	$\tilde{x}_5$	$\tilde{x}_6$
$\tilde{x}_1$	$<\{\tilde{x}_2\},\phi,<\tilde{e}_1>>$	$<\{\tilde{x}_2,\tilde{x}_4\},\{\tilde{x}_3,\tilde{x}_6\},<\tilde{e}_1,\tilde{e}_2>>$	$<\{\tilde{x}_2,\tilde{x}_4\},\{\tilde{x}_3,\tilde{x}_5\},<\tilde{e}_1,\tilde{e}_2>>$
$\tilde{x}_2$	$<\{\tilde{x}_1\},\phi,<\tilde{e}_1>>$	$<\{\tilde{x}_1,\tilde{x}_4\},\{\tilde{x}_3,\tilde{x}_6\},<\tilde{e}_1,\tilde{e}_2>>$	$<\{\tilde{x}_1,\tilde{x}_4\},\{\tilde{x}_3,\tilde{x}_5\},<\tilde{e}_1,\tilde{e}_2>>$
$\tilde{x}_3$	ϕ	$<\{\tilde{x}_4\},\{\tilde{x}_6\},<\tilde{e}_2>>$	$<\{\tilde{x}_4\},\{\tilde{x}_5\},<\tilde{e}_2>>$
$\tilde{x}_4$	ϕ	$<\{\tilde{x}_3\},\{\tilde{x}_6\},<\tilde{e}_2>>$	$<\{\tilde{x}_3\},\{\tilde{x}_5\},<\tilde{e}_2>>$

TABLE 5
The Closure of the Adjacency Matrix of the Cyclic FM in Fig. 2

	$\tilde{x}_2$	$\tilde{x}_3$	$\tilde{x}_4$
$\tilde{x}_1$	$<\phi,\{\tilde{x}_3,\tilde{x}_4\},<\tilde{e}_1,\tilde{e}_2>>$	$<\{\tilde{x}_2\},\phi,<\tilde{e}_1>>$	$<\phi,\{\tilde{x}_2,\tilde{x}_3\},<\tilde{e}_1,\tilde{e}_2>>$
$\tilde{x}_2$	$<\{\tilde{x}_1\},\{\tilde{x}_3,\tilde{x}_4\},<\tilde{e}_1,\tilde{e}_2>>$	$<\{\tilde{x}_1\},\phi,<\tilde{e}_1>>$	$<\{\tilde{x}_1\},\{\tilde{x}_2,\tilde{x}_3\},<\tilde{e}_1,\tilde{e}_2>>$
$\tilde{x}_3$	$<\phi,\{\tilde{x}_4\},<\tilde{e}_2>>$	$<\{\tilde{x}_1\},\{\tilde{x}_2,\tilde{x}_4\},<\tilde{e}_2,\tilde{e}_1>>$	$<\phi,\{\tilde{x}_2\},<\tilde{e}_2>>$

this paper is on the use of graphical tools for mathematical analysis, our exploration will concentrate on the algebraic analysis by employing the specific FM closure matrix.

5 FM-Based Knowledge Representation and Indexing Table

A fuzzy inference system comprises a set of fuzzy IF-THEN rules that describes the input-output mapping relationship of the system [22]. The basic structure of a fuzzy inference system consists of four principal components: a knowledge base, a data base, an inference mechanism, and an interface of fuzzification and defuzzification [22], where only the knowledge base and inference mechanism are particularly relevant to the specific modeling approach, namely, FM in this paper. In the following two sections, the FM-based knowledge representation approach and the corresponding reasoning algorithms will be proposed for rule-based systems.

Knowledge representation is one of the most important and actively investigated areas in artificial intelligence [14]. Fuzzy production rules (FPRs) among others are widely used in expert systems to represent fuzzy, imprecise, and vague concepts [23]. In this section, we address issues of applying FM to FPRs. In FM-based knowledge representation, each edge represents a rule in which the in-vertex represents the antecedent of the rule and the out-vertex represents the consequent. Furthermore, each path—a sequence of edges—represents a reasoning chain.

5.1 FM-Based Representation of Rules

The uncertainty of an elementary rule can be modeled by an FM as shown in Fig. 3. The figure illustrates the following rule: IF $\tilde{x}_1$ THEN $\tilde{x}_2$ (CF_1). According to the rule, the truth value of the consequent is the product of the truth value of the antecedent and the CF of the rule.

An FPR is called a compound FPR if its antecedent part and/or consequent part "AND" or "OR" connectors. Fig. 4 models various types of compound production rule in accordance with [25], [9]. Fig. 4a describes the following rule: IF $\tilde{x}_3$ AND $\tilde{x}_4$ THEN $\tilde{x}_5$ (CF_2). There is conjunction operation in this rule. Zadeh proposed to use the operators "min = ^" for conjunction, and "max = v" for union [26]. Thus, $\mu(x_5) = \min(\mu(x_3), \mu(x_4)) \times CF_2$. Fig. 4b describes the following two rules: 1) IF $\tilde{x}_6$ THEN $\tilde{x}_8$ (CF_3) and 2) IF $\tilde{x}_7$ THEN $\tilde{x}_8$ (CF_4). In this case, the truth value of consequent equals the maximal one among those obtained from different rules. When the two rules have the same CF, i.e., $CF_3 = CF_4 = CF$, the combined rule illustrated by the FM is IF $\tilde{x}_6$ OR $\tilde{x}_7$ THEN $\tilde{x}_8$ (CF). If there are several union antecedents, the overall truth value of the antecedents equals the maximal one. Fig. 4c illustrates the following rule: IF $\tilde{x}_9$ THEN $\tilde{x}_{10}$ AND $\tilde{x}_{11}$ (CF_5). In Fig. 4d, two separate basic rules are described. We do not model a rule like: IF $\tilde{x}_{12}$ THEN $\tilde{x}_{13}$ OR $\tilde{x}_{14}$ (CF) since it is not allowed in a rule base [9].

The graphic representation is of help in understanding the system yet difficult for computers to process. In addition, a graph of any kind will not be interpretable for a complex system. Hence, there is a need for an alternative representation in an algebraic format namely the FM closure matrix, e.g., as shown in Table 4.

In order for the FM closure matrix of any use in real-world applications, there are two key issues to be addressed, namely, the construction of the matrix and the extraction of entries from the matrix. The following two sections are devoted to solving these problems. In Section 5.2, the indexing table is introduced to enable efficient extraction of entries. The issue of construction will be coped with in Section 5.3 by presenting an iterative construction scheme. The two approaches are capable of improving query processing time and matrix construction time, respectively.

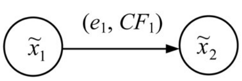

Fig. 3. FM-based representation of a basic rule.

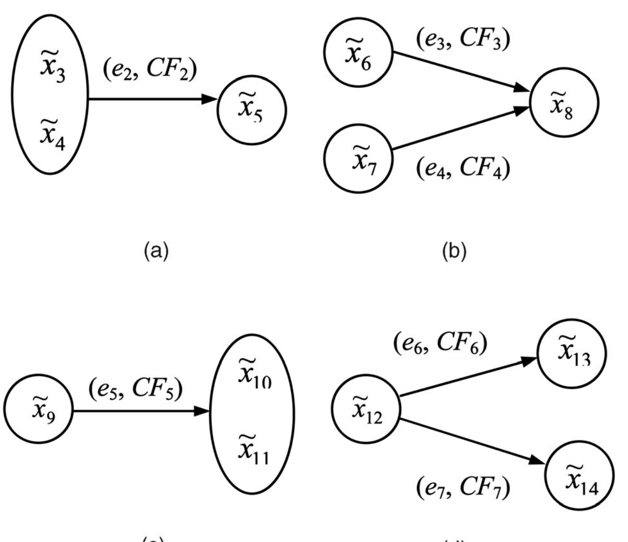

Fig. 4. FM-based representations of compound rules.

TABLE 6
The Indexing Table of the FM Closure Matrix in Table 4

Number	Element	(Entries)	(Entries)
1	$\tilde{x}_1$	$(a^*_{1,3}, a^*_{1,5}, a^*_{1,6})$	ϕ
2	$\tilde{x}_2$	$(a^*_{2,3}, a^*_{2,5}, a^*_{2,6})$	ϕ
3	$\tilde{x}_3$	$(a^*_{3,5}, a^*_{3,6})$	$(a^*_{1,3}, a^*_{2,3})$
4	$\tilde{x}_4$	$(a^*_{4,5}, a^*_{4,6})$	ϕ
5	$\tilde{x}_5$	ϕ	$(a^*_{1,5}, a^*_{2,5}, a^*_{3,5}, a^*_{4,5})$
6	$\tilde{x}_6$	ϕ	$(a^*_{1,6}, a^*_{2,6}, a^*_{3,6}, a^*_{4,6})$

5.2 Indexing Table

According to the definition, an FM closure matrix is an $I \times I$ matrix where I is the total number of elements. For a large system, however, empty entries are often the majority in the matrix, i.e., the FM closure matrix is a sparse matrix. This results in a demand for a method that is able to avoid searching over a mass of empty entries. The indexing technique described in Section 2.2 becomes an obvious choice, as the motivation of indexing technique is to handle the issue of how relevant portions of data can be located and extracted. An indexing table is thus proposed to speed up access to the FM closure matrix. Similar to the inverted file example in Table 1, an indexing table contains a complete list of elements. To facilitate both forward and backward chaining, the indexing table further uses *two*—instead of only one in an inverted file—tuples for each element where the first one stores a list of pointers to all nonempty entries in the row of that element in the closure matrix (for forward chaining) and the second tuple stores a list of pointers to all nonempty entries in the column of that element (for backward chaining). An FM closure matrix is thus indexed by elements—equivalent to terms in an IR system. The indexing table for the FM closure matrix in Table 4 is given in Table 6, whereas the generating process of the indexing table will be discussed in the next subsection. In Table 6, for example, the third column of row $\tilde{x}_3$ of the indexing table is $(a^*_{3,5}, a^*_{3,6})$ meaning that in the row $\tilde{x}_3$ of the corresponding FM closure matrix in Table 4, only $a^*_{3,5}$ and $a^*_{3,6}$ are nonempty entries. The fourth column of row $\tilde{x}_3$ of the indexing table is $(a^*_{1,3}, a^*_{2,3})$ meaning that in the column $\tilde{x}_3$ of the corresponding FM closure matrix, only $a^*1,3$ and $a^*_{2,3}$ are nonempty entries. The third column of row $\tilde{x}_5$ of the indexing table is indicating that in the row $\tilde{x}_5$ of the corresponding FM closure matrix, there is no nonempty entry.

5.3 Construction and Expansion of FM-Based Knowledge Base

The other issue is how to efficiently construct the FM-based rule base (i.e., the FM closure matrix). It is not efficient to calculate the FM closure matrix as defined in Section 4. Especially, it is not feasible to expand the FM-based rule base in that way. Therefore, we propose an iterative way of constructing it, which can significantly reduce the construction time and facilitate the expansion.

The FM-based knowledge base can be constructed in similar steps to IR system as described in Section 2.2, namely, gathering a collection of rules, preprocessing the collection, and indexing them. Consider the following five example rules $R_1 \sim R_5$ taken out from a real-world, large-scale fault diagnosis system.

- R_1: IF voltage is high AND altitude is low THEN guidance system breaks down ($CF_1 = 0.90$).
- R_2: IF guidance system breaks down AND relay is abnormal THEN guidance cabin breaks down ($CF_2 = 0.95$).
- R_3: IF voltage is high AND altitude is high THEN control system breaks down ($CF_3 = 0.80$).
- R_4: IF voltage is low AND output is large THEN guidance cabin cable net breaks down ($CF_4 = 0.95$).
- R_5: IF guidance system breaks down AND relay is normal THEN guidance cabin cable net breaks down ($CF_5 = 0.85$).

First, each proposition is assigned a number $\tilde{x}_i$ starting from $\tilde{x}_1$ one by one, resulting in a list of propositions as shown in Table 7. The list is then sorted alphabetically to speed up querying, as shown in Table 8. By using these notations, rules $R_1 \sim R_5$ are represented by $r_1 \sim r_5$ as follows:

- r_1: IF $\tilde{x}_1$ AND $\tilde{x}_2$ THEN $\tilde{x}_3$ ($CF_1 = 0.90$).
- r_2: IF $\tilde{x}_3$ AND $\tilde{x}_4$ THEN $\tilde{x}_5$ ($CF_2 = 0.95$).
- r_3: IF $\tilde{x}_1$ AND $\tilde{x}_6$ THEN $\tilde{x}_7$ ($CF_3 = 0.80$).

TABLE 7
List of Propositions

Number	Element	Proposition
1	$\tilde{x}_1$	voltage is high
2	$\tilde{x}_2$	altitude is low
3	$\tilde{x}_3$	guidance system breaks down
4	$\tilde{x}_4$	relay is abnormal
5	$\tilde{x}_5$	guidance cabin breaks down
6	$\tilde{x}_6$	altitude is high
7	$\tilde{x}_7$	control system breaks down
8	$\tilde{x}_8$	voltage is low
9	$\tilde{x}_9$	output is large
10	$\tilde{x}_{10}$	guidance cabin cable net breaks down
11	$\tilde{x}_{11}$	relay is normal

TABLE 8
List of Propositions in Alphabetic Order

Number	Proposition	Element
1	altitude is high	$\tilde{x}_6$
2	altitude is low	$\tilde{x}_2$
3	control system breaks down	$\tilde{x}_7$
4	guidance cabin breaks down	$\tilde{x}_5$
5	guidance cabin cable net breaks down	$\tilde{x}_{10}$
6	guidance system breaks down	$\tilde{x}_3$
7	output is large	$\tilde{x}_9$
8	relay is abnormal	$\tilde{x}_4$
9	relay is normal	$\tilde{x}_{11}$
10	voltage is high	$\tilde{x}_1$
11	voltage is low	$\tilde{x}_8$

- r_4: IF $\tilde{x}_8$ AND $\tilde{x}_9$ THEN $\tilde{x}_{10}$ ($CF_4 = 0.95$).
- r_5: IF $\tilde{x}_3$ AND $\tilde{x}_{11}$ THEN $\tilde{x}_{10}$ ($CF_5 = 0.85$).

These rules are then illustrated by the FM as shown in Fig. 5 where edge e_k represents rule r_k. The closure matrix of the FM can be generated from scratch in the following way. It starts from r_1: IF $\tilde{x}_1$ AND $\tilde{x}_2$ THEN $\tilde{x}_3$ ($CF_1 = 0.90$), in which there are two input elements $\tilde{x}_1$ and $\tilde{x}_2$ and one output element $\tilde{x}_3$. This results in two nonempty entries in the closure matrix, namely, $a^*_{1,3}$ and $a^*_{2,3}$ with values of $a^*1,3 = < \{\tilde{x}_2\}, \phi, < \tilde{e}_1 >>$ and $a^*2,3 =< \{\tilde{x}_1\}, \phi, < \tilde{e}_1 >>$, respectively. These entries containing paths with a length of one can be immediately added into the closure matrix as shown in Table 9. Since this is the first rule added, there is no effect on other entries that are in fact all empty. The indexing table is intuitively constructed by listing elements and grouping their occurrences, as shown in Table 10. Next, we consider r_2: IF $\tilde{x}_3$ AND $\tilde{x}_4$ THEN $\tilde{x}_5$ ($CF_2 = 0.95$), where there are two input elements $\tilde{x}_3$ and $\tilde{x}_4$ and one output element $\tilde{x}_5$. It yields two nonempty entries, namely, $a^*_{3,5}$ and $a^*_{4,5}$ with values of $a^*_{3,5} =< \{\tilde{x}_4\}, \phi, < \tilde{e}_2 >>$ and $a^*_{4,5} =< \{\tilde{x}_3\}, \phi, < \tilde{e}_2 >>$, respectively. These entries can be immediately merged into the closure matrix. Subsequently, each of the added rules (here is $\tilde{e}_2$) should be concatenated with existing paths if possible. It is done by checking whether the input elements $\tilde{x}_3$ and $\tilde{x}_4$ are other rules' (edges') output elements and whether the output element $\tilde{x}_5$ is other rules' input element. From the fourth column of the row $\tilde{x}_3$ in Table 10, it is found that the input element $\tilde{x}_3$ is the output of both $\tilde{x}_1$ and $\tilde{x}_2$. So, $a^*_{3,5}$ should be combined with $a^*_{1,3}$ and $a^*_{2,3}$ to form new entries $a^*_{1,5}$ and $a^*_{2,5}$. According to the definition of a simple path in Definition 2, the co-input, co-output, and path can be easily calculated and then we have $a^*_{1,5} =< \{\tilde{x}_2, \tilde{x}_4\}, \{\tilde{x}_3\}, < \tilde{e}_1, \tilde{e}_2 >>$, and so forth, as shown in Table 11. This sort of update for path concatenation ought to be done in only one step. For example, the input element $\tilde{x}_3$ results in an update from $\tilde{x}_1$ and $\tilde{x}_2$, whereas there is no need for the further check of whether $\tilde{x}_1$ and $\tilde{x}_2$ are other elements' output since it should have been done when they were added. From Table 10, it is found that $\tilde{x}_5$ is not an input element so far. Thus, there is no update to follow. Accordingly, the indexing table is updated as shown in Table 12. To the end, the closure matrix and corresponding indexing table are generated and shown in Table 13 and Table 14, respectively.

In rule-based systems, the demand for updating a rule base is obvious, indicating a rule base should be possible and easy to add new rules or modify existing ones. The proposed iterative method is able to efficiently recapture or

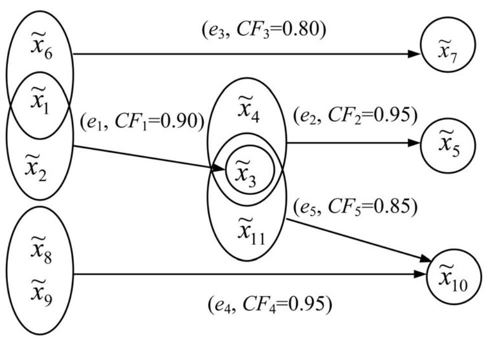

Fig. 5. Fuzzy metagraph of rules $r_1 \sim r_5$.

TABLE 9
The FM Closure Matrix of R_1

	$\tilde{x}_3$
$\tilde{x}_1$	$< \{\tilde{x}_2\}, \phi, < \tilde{e}_1 >>$
$\tilde{x}_2$	$< \{\tilde{x}_1\}, \phi, < \tilde{e}_1 >>$

TABLE 10
The Indexing Table of the FM Closure Matrix in Table 9

Number	Element	(Entries)	(Entries)
1	$\tilde{x}_1$	$(a^*_{1,3})$	ϕ
2	$\tilde{x}_2$	$(a^*_{2,3})$	ϕ
3	$\tilde{x}_3$	ϕ	$(a^*_{1,3}, a^*_{2,3})$

TABLE 11
The FM Closure Matrix of R_1 and R_2

	$\tilde{x}_3$	$\tilde{x}_5$
$\tilde{x}_1$	$<\{\tilde{x}_2\},\phi,<\tilde{e}_1>>$	$<\{\tilde{x}_2,\tilde{x}_4\},\{\tilde{x}_3\},<\tilde{e}_1,\tilde{e}_2>>$
$\tilde{x}_2$	$<\{\tilde{x}_1\},\phi,<\tilde{e}_1>>$	$<\{\tilde{x}_1,\tilde{x}_4\},\{\tilde{x}_3\},<\tilde{e}_1,\tilde{e}_2>>$
$\tilde{x}_3$	ϕ	$<\{\tilde{x}_4\},\phi,<\tilde{e}_2>>$
$\tilde{x}_4$	ϕ	$<\{\tilde{x}_3\},\phi,<\tilde{e}_2>>$

TABLE 12
The Indexing Table of the FM Closure Matrix in Table 11

Number	Element	(Entries)	(Entries)
1	$\tilde{x}_1$	$(a^*_{1,3}, a^*_{1,5})$	ϕ
2	$\tilde{x}_2$	$(a^*_{2,3}, a^*_{2,5})$	ϕ
3	$\tilde{x}_3$	$(a^*_{3,5})$	$(a^*_{1,3}, a^*_{2,3})$
4	$\tilde{x}_4$	$(a^*_{4,5})$	ϕ
5	$\tilde{x}_5$	ϕ	$(a^*_{1,5}, a^*_{2,5}, a^*_{3,5}, a^*_{4,5})$

update the FM closure matrix and the indexing table in the same way as they are constructed.

It is seen that the indexing table can be constructed in a similar, but slightly different, way to the inverted file [17]. First, indexing elements (terms) are gathered, e.g., as given in Table 7 and sorted alphabetically, e.g., as given in Table 8. Second, nonempty entries associated with each elements are grouped, resulting in an indexing table, e.g., as given in Table 14. As a result, the search space is reduced from $I \times I$ to I, which is further facilitated by sorting the list in alphabetic order such that binary searching is achieved. With the implementation of the indexing technique, it is possible to find nonempty entries without resorting to an entry-by-entry search through the closure matrix.

6 Inference Mechanisms

On the basis of the FM closure matrix and the indexing table, this section investigates corresponding inference mechanisms with the aim of instantly acquiring relevant rules over a large collection of rules. In general, there exist three inference strategies. The first one is called a forward chaining where the truth values of initial propositions are known and truth values of unspecified noninitial propositions need to be computed. In the second strategy, called a backward chaining, the existence of a goal needs to be established. The last one is a hybrid strategy in which truth values of initial propositions are known and the goal propositions are specified in advance for which their truth values need to be computed.

In order for inference, the system will need to have access to facts that are statements like "Temperature is low." The collection of facts known to the system at any given time is called a fact base [27]. In general, a fact base $\tilde{B}$ is established in advance and can be updated during the inference process.

6.1 Mechanism 1—Forward Chaining (A Data-Driven Strategy)

In the forward chaining, rules are examined and fired on the basis of the available fact base without any predetermined goals, indicating all possible propositions are to be explored during the reasoning procedure. The reasoning algorithm is proposed as follows:

Step 1. Construct fact base $\tilde{B}$ according to the given facts. Select the first fact from $\tilde{B}$ and go to next step.

Step 2. Assume the fact is $\tilde{x}_i$ and find the tuple in the third column in the row $\tilde{x}_i$ of the indexing table. According to the tuple, collect entries from the FM closure matrix, and then sort them by the length of edge in ascending order. Pick up the first entry and go to next step.

Step 3. Assume the entry is $a^*_{i,j}$ which is one or more triples. The algorithm handles these triples one by one. Each triple consists of three components. The first one is co-inputs that cannot be inferred by the reasoning path according to the definition of A^* and, thus, should be given prior to reasoning. If all co-inputs are known, the reasoning algorithm invokes the sequence of edges (i.e., rules) given by the third component, namely, a simple path. Otherwise, request the truth values of the unknowns in co-inputs (if any of them does not exist, this path(triple) fails and the algorithm deals with next triple). The co-outputs in the second component can be expected as by-product outputs during the reasoning process. Append all the results to the end of $\tilde{B}$. After

TABLE 13
The Closure of the Adjacency Matrix of the FM in Fig. 5

	$\tilde{x}_3$	$\tilde{x}_5$	$\tilde{x}_7$	$\tilde{x}_{10}$
$\tilde{x}_1$	$<\{\tilde{x}_2\},\phi,<\tilde{e}_1>>$	$<\{\tilde{x}_2,\tilde{x}_4\},\{\tilde{x}_3\},<\tilde{e}_1,\tilde{e}_2>>$	$<\{\tilde{x}_6\},\phi,<\tilde{e}_3>>$	$<\{\tilde{x}_2,\tilde{x}_{11}\},\{\tilde{x}_3\},<\tilde{e}_1,\tilde{e}_5>>$
$\tilde{x}_2$	$<\{\tilde{x}_1\},\phi,<\tilde{e}_1>>$	$<\{\tilde{x}_1,\tilde{x}_4\},\{\tilde{x}_3\},<\tilde{e}_1,\tilde{e}_2>>$	ϕ	$<\{\tilde{x}_1,\tilde{x}_{11}\},\{\tilde{x}_3\},<\tilde{e}_1,\tilde{e}_5>>$
$\tilde{x}_3$	ϕ	$<\{\tilde{x}_4\},\phi,<\tilde{e}_2>>$	ϕ	$<\{\tilde{x}_{11}\},\phi,<\tilde{e}_5>>$
$\tilde{x}_4$	ϕ	$<\{\tilde{x}_3\},\phi,<\tilde{e}_2>>$	ϕ	ϕ
$\tilde{x}_6$	ϕ	ϕ	$<\{\tilde{x}_1\},\phi,<\tilde{e}_3>>$	ϕ
$\tilde{x}_8$	ϕ	ϕ	ϕ	$<\{\tilde{x}_9\},\phi,<\tilde{e}_4>>$
$\tilde{x}_9$	ϕ	ϕ	ϕ	$<\{\tilde{x}_8\},\phi,<\tilde{e}_4>>$
$\tilde{x}_{11}$	ϕ	ϕ	ϕ	$<\{\tilde{x}_3\},\phi,<\tilde{e}_5>>$

TABLE 14
The Indexing Table of the FM Closure Matrix in Table 13

Number	Element	(Entries)	(Entries)
1	$\tilde{x}_1$	$(a^*_{1,3}, a^*_{1,5}, a^*_{1,7}, a^*_{1,10})$	ϕ
2	$\tilde{x}_2$	$(a^*_{2,3}, a^*_{2,5}, a^*_{2,10})$	ϕ
3	$\tilde{x}_3$	$(a^*_{3,5}, a^*_{3,10})$	$(a^*_{1,3}, a^*_{2,3})$
4	$\tilde{x}_4$	$(a^*_{4,5})$	ϕ
5	$\tilde{x}_5$	ϕ	$(a^*_{1,5}, a^*_{2,5}, a^*_{3,5}, a^*_{4,5})$
6	$\tilde{x}_6$	$(a^*_{6,7})$	ϕ
7	$\tilde{x}_7$	ϕ	$(a^*_{1,7}, a^*_{6,7})$
8	$\tilde{x}_8$	$(a^*_{8,10})$	ϕ
9	$\tilde{x}_9$	$(a^*_{9,10})$	ϕ
10	$\tilde{x}_{10}$	ϕ	$(a^*_{1,10}, a^*_{2,10}, a^*_{3,10}, a^*_{8,10}, a^*_{9,10}, a^*_{11,10})$
11	$\tilde{x}_{11}$	$(a^*_{11,10})$	ϕ

examining all the triples of $a^*_{i,j}$ in the same way, go to next step.

Step 4. Is there another entry that has been collected from the FM closure matrix?
1) If yes, pick up next entry and then go to Step 3.
2) Otherwise, go to next step.

Step 5. Is there another nonprocessed fact in $\tilde{B}$?
1) If yes, select the next fact from $\tilde{B}$ and then go to Step 2.
2) Otherwise, go to next step.

Step 6. Draw a conclusion. One or more conclusion propositions can be generated depending on the actual application. For each conclusion proposition, one or more truth values will be produced depending on the number of rules leading to this proposition. Since different rules have a union relationship, select the maximum truth value as its truth value.

In the above reasoning process, a depth-first search strategy is adopted as described in [29]. To illustrate the procedure, we utilize the rules $R_1 \sim R_5$ as an example whose FM, closure matrix, and indexing table are given in Fig. 5, in Table 13 and in Table 14, respectively.

Assume that the initial proposition is "altitude is low" with a truth value of 0.95. According to Table 8, the proposition is $\tilde{x}_2$ so that a fact base $\tilde{B} = \{\tilde{x}_2\} = \{(x_2, 0.95)\}$ can be constructed. The reasoning starts by selecting the only element $\tilde{x}_2$ from $\tilde{B}$. First, the system finds the tuple in the third column in the row $\tilde{x}_2$ of Table 14, namely, $(a^*_{2,3}, a^*_{2,5}, a^*_{2,10})$, meaning that $a^*_{2,3}$, $a^*_{2,5}$, and $a^*_{2,10}$ are nonempty entries in the FM closure matrix. They are sorted as $a^*_{2,3}$, $a^*_{2,5}$, and $a^*_{2,10}$ by the length of edge in ascending order. The first nonempty entry $a^*_{2,3}$ consists of only one triple $< \{\tilde{x}_1\}, \phi, < \tilde{e}_1 >>$, where the co-input is $In\left(a^*_{2,3}\right) = \{\tilde{x}_1\}$. Since $\tilde{x}_1$ is not included in the fact base, we need the truth value of x_1. According to Table 7, $\tilde{x}_1$ is "voltage is high." Assume that the truth value of "voltage is high" is 0.85, i.e., $\mu(x_1) = 0.85$. According to $Path\left(a^*_{2,3}\right) =< \tilde{e}_1 >$, we have r_1: IF $\tilde{x}_1$ AND $\tilde{x}_2$ THEN $\tilde{x}_3$ ($CF_1 = 0.90$). It can be inferred that $\tilde{x}_3$ has a truth value of

$$\begin{aligned}\mu(x_3) &= \min\{\mu(x_1), \mu(x_2)\} \times CF_1 \\ &= \min\{0.85, 0.95\} \times 0.90 \approx 0.77.\end{aligned}$$

In addition, the co-output is $Out\left(a^*_{2,3}\right) = \phi$. As a result, the fact base is updated as $\tilde{B} = \{(x_2, 0.95), (x_1, 0.85), (x_3, 0.77)\}$.

The next step is to check the second entry $a^*_{2,5}$ consisting of one triple $< \{\tilde{x}_1, \tilde{x}_4\}, \{\tilde{x}_3\}, < \tilde{e}_1, \tilde{e}_2 >>$ that requires co-inputs of $In\left(a^*_{2,5}\right) = \{\tilde{x}_1, \tilde{x}_4\}$. Although $\tilde{x}_1$ is given by the fact base, $\tilde{x}_4$ is not. According to Table 7, $\tilde{x}_4$ is "relay is abnormal." Assume "relay is abnormal" has a truth value of 0.1, i.e., $\mu(x_4) = 0.1$. To cope with the low truth value, one useful parameter in approximate reasoning is threshold value. A threshold value may be assigned to each proposition in the antecedent of a FPR to prevent and reduce rule misfiring [24]. In this paper, we assign a single threshold value ($\lambda \in [0, 1]$) to all propositions, which is set as 0.2. Therefore, the proposition with a truth value lower than 0.2 should be disregarded. Since $\tilde{x}_4$ has a truth value of 0.1, the path—chain of rules—consisting of $< \tilde{e}_1, \tilde{e}_2 >$ cannot be fired. The fact base does not need to be updated. It is noticed that even though a single co-input fact is not true, the chain of rules (in $a^*_{2,5}$) will not be examined and fired. This is another advantage of FM-based representation.

The last entry $a^*_{2,10}$ is one triple $< \{\tilde{x}_1, \tilde{x}_{11}\}, \{\tilde{x}_3\}, < \tilde{e}_1, \tilde{e}_5 >>$ with a co-input of $In\left(a^*_{2,10}\right) = \{\tilde{x}_1, \tilde{x}_{11}\}$. According to Table 7, $\tilde{x}_{11}$ is "relay is normal." Assume "relay is normal" has a truth value of 0.9, i.e., $\mu(x_{11}) = 0.9$. Since $\tilde{x}_1$ is already given by the fact base, two rules r_1 and r_5 identified by $< \tilde{e}_1, \tilde{e}_5 >$ can be fired. Because r_1 has been executed previously, only r_5: IF $\tilde{x}_3$ AND $\tilde{x}_{11}$THEN $\tilde{x}_{10}$ ($CF_5 = 0.85$) will be fired. Therefore, a conclusion $\tilde{x}_{10}$ can be drawn with a truth value of

$$\begin{aligned}\mu(x_{10}) &= \min\{\mu(x_3), \mu(x_{11})\} \times CF_5 \\ &= \min\{0.77, 0.9\} \times 0.85 \approx 0.65.\end{aligned}$$

According to Table 7, $\tilde{x}_{10}$ is "guidance cabin cable net breaks down." The fact base is updated again as

$$\tilde{B} = \{(x_2, 0.95), (x_1, 0.85), (x_3, 0.77), (x_{11}, 0.9), (x_{10}, 0.65)\}.$$

At the end of the reasoning, the conclusion of "guidance cabin cable net breaks down" can be reached with the truth value of 0.65. The truth value of proposition $\tilde{x}_3$—"guidance system breaks down" is also obtained.

The main advantage of the proposed method is that in the entire reasoning process only rules relevant to initial proposition $\tilde{x}_2$ have been examined whereas the remaining rules (for example, r_3 and r_4) in the rule base are untouched. Furthermore, even though a single co-input fact in a reasoning chain (a simple path) is not true, the whole chain will immediately be disregarded without requirement for examining individual rules. In the standard forward chaining, however, all of the rules (or a subset) in the rule base are examined [27] and so it can be wasteful. When the rule base is large, the time saving is significant.

6.2 Mechanism 2—Backward Chaining (A Goal-Driven Strategy)

The backward chaining is the same as the forward chaining except that the fourth column of the indexing table is applied. The algorithm is described as follows:

Step 1. Select the goal proposition and go to next step.

Step 2. Assume the goal is $\tilde{x}_j$ and find the tuple in the fourth column in the row $\tilde{x}_j$ of the indexing table. According to the tuple, collect entries from the FM closure matrix, and then sort them by the length of edge in ascending order. Pick up the first entry and go to next step.

Step 3. Assume the entry is $a^*_{i,j}$ which is one or more triples. The algorithm deals with these triples one by one. Each triple consists of three components. The first one is co-inputs that should be given prior to reasoning. If all co-inputs are known, the reasoning algorithm invokes the sequence of edges (i.e., rules) given by the third component, namely, a simple path. Otherwise, request the truth values of the unknowns in co-inputs (if any of them does not exist, this path (triple) fails and the algorithm deals with next triple). The co-outputs in the second component can be expected as by-product outputs during the reasoning process. Append all the results to the end of $\tilde{B}$. After examining all the triples of $a^*_{i,j}$ in the same way, go to next step.

Step 4. Is there another entry that has been collected from the FM closure matrix?
1) If yes, pick up next entry and then go to Step 3.
2) Otherwise, go to next step.

Step 5. Is there another goal?
1) If yes, select the next goal and then go to Step 2.
2) Otherwise, go to next step.

Step 6. Draw a conclusion.

For example, assume the goal is $\tilde{x}_{10}$—"guidance cabin cable net breaks down." The procedure will be to first find the tuple in the fourth column of the row $\tilde{x}_{10}$ of the indexing table in Table 14. According to the tuple, collect the entries from the FM closure matrix and examine the co-input facts for each entry. If co-input facts are true with a certain truth value, the corresponding rule is fired. In this example, rules r_4, r_5, and r_1 may be examined and fired, yet r_2 and r_3 will not be touched at all.

6.3 Mechanism 3—Backward and Forward Chaining (A Hybrid Strategy)

The backward and forward chaining is suggested to solve a problem of if there is an antecedent-consequent relationship from the initial proposition to the goal proposition. This strategy makes the computation applicable to more situations [28].

Assuming the given fact is $\tilde{x}_i$ and the goal proposition is $\tilde{x}_j$, the reasoning algorithm can be presented as follows:

Step 1. Examine the entry $a^*_{i,j}$. Is it an empty entry?
1) If yes, it means $\tilde{x}_j$ cannot be derived from $\tilde{x}_i$ and the reasoning process terminates.
2) Otherwise, construct fact base $\tilde{B}$ according to the given fact $\tilde{x}_i$. Select $\tilde{x}_i$ and go to next step.

Step 2. $a^*_{i,j}$ is one or more triples. The algorithm deals with these triples one by one. Each triple consists of three components. The first one is co-inputs that should be given prior to reasoning. If all co-inputs are known, the reasoning algorithm invokes the sequence of edges (i.e., rules) given by the third component, namely, a simple path. Otherwise, request the truth value of the unknown in co-inputs (if any of them does not exist, the path (triple) fails and the algorithm deals with next triple). The co-outputs in the second component can be expected as by-product outputs during the reasoning process. Append all the results to the end of $\tilde{B}$. After examining all the triples of $a^*_{i,j}$ in the same way, go to next step.

Step 3. Draw a conclusion.

Assume that the truth value of $\tilde{x}_2$ "altitude is low" is 0.95 and the question is what the truth value of proposition $\tilde{x}_{10}$ "guidance cabin cable net breaks down" is. Therefore, a fact base $\tilde{B} = \{\tilde{x}_2\} = \{(x_2, 0.95)\}$ can be constructed. It is found that $a^*_{2,10}$ consists of one triple $< \{\tilde{x}_1, \tilde{x}_{11}\}, \{\tilde{x}_3\}, < \tilde{e}_1, \tilde{e}_5 >>$. The first component of this triple is the co-inputs $\{\tilde{x}_1, \tilde{x}_{11}\}$ that should be given beforehand. Even if only one of the co-inputs is false, it can immediately be concluded that the proposition $\tilde{x}_{10}$ cannot be derived from $\tilde{x}_2$. Otherwise, if the truth value of $\tilde{x}_1$ is 0.85 and $\tilde{x}_{11}$ is 0.9, the rules r_1 and r_5 corresponding to $\tilde{e}_1$ and $\tilde{e}_5$ can be fired successively. According to r_1, the truth value of $\tilde{x}_3$ is

$$\mu(x_3) = \min\{0.85, 0.95\} \times 0.90 \approx 0.77.$$

As a result, the fact base is updated as

$$\tilde{B} = \{(x_2, 0.95), (x_1, 0.85), \{x_{11}, 0.9\}, (x_3, 0.77)\}.$$

Next, r_5 is fired and we can conclude

$$\mu(x_{10}) = \min\{0.77, 0.9\} \times 0.85 \approx 0.65.$$

By employing backward and forward chaining, only a single entry is explored and there is no need for indexing table because the reasoning path between antecedent and consequent is acquired directly from the FM closure matrix.

6.4 Discussion

Standard (conventional) inference algorithms involve a great deal of redundant rule examination [27], which wastefully explores all or a subset of rules. The search time for conventional method is linear on the rule base size and may be exponential on the size of variables (the number of elements) since the total number of rules in a system is often an exponential function of the number of system variables [30]. By employing the FM and IR-based approaches, the time-consuming task of searching and matching in a large rule base becomes a relatively simple task of picking-up and only relevant rules are to be examined. Since there is no need to match rules, the time saving is tremendous as compared to standard inference algorithms.

The improvement comes from two-fold. First, the FM-based method is able to directly acquire relevant reasoning paths since the entries of the FM closure matrix provide all the paths, co-inputs, and co-outputs between any two elements (i.e., propositions in rule-based systems). It means that the FM closure matrix can be considered a powerful precompiled rule base. Second, the alphabetic indexing table enables a binary search and, therefore, instant extraction of entries from the FM closure matrix is achieved.

7 Conclusion

In this paper, we have proposed a new graph-theoretic construct, namely, the FM for graphical modeling and mathematical analysis. As a combination of fuzzy directed graphs and fuzzy hypergraphs, the FM is capable of describing directed relationships between sets of fuzzy elements. Moreover, the FM closure matrix—the algebraic structure of FM—distinguishes the FM from conventional fuzzy graphs by providing all the paths, co-inputs and co-outputs between any two elements. This makes the FM a powerful tool for the analysis of the connectivity between sets of fuzzy elements.

In applying the FM to rule-based systems, we first developed a knowledge representation scheme based on the FM closure matrix, which is regarded as a preprocessing of rule base. As the complexity of this preprocessing is a key for real-world applications, an iterative approach was presented to facilitate the construction and expansion of the FM closure matrix. Inspired by the success of IR techniques in Web-based query, we further introduced the indexing table to allow a quick extraction of relevant entries from the FM closure matrix by means of a binary search. In the end, on the basis of the combination of the indexing table and the FM closure matrix, inference algorithms were proposed to enable an instant access to relevant rules over a large collection of rules. As compared to standard inference algorithms, the improvement of the proposed algorithms in terms of inference efficiency is significant. The larger the rule base is, the more time the method is expected to save. Therefore, the proposed approach is most effective in the situation where the rule base is large and often accessed as the cost of using the approach is to establish the system though not high.

There are three main contributions in this paper. The first one is the proposal of the FM construct and the FM closure matrix, which brings metagraphs into the realm of uncertain representation and approximate reasoning. The second contribution is to present an iterative approach for facilitating the construction and expansion of the FM closure matrix. The third one is to employ an inverted-file approach for a quick index into the FM closure matrix and to apply the combined approaches to rule-based systems. The last two are considered as optimization algorithms for computing the FM closure matrix. It is justified that the combination of the FM and the IR technique offers an effective tool for rule-based systems. In addition, the iterative approach and IR technique can be applied to the metagraph closure matrix as well.

Appendix

Multiplication and Addition of Adjacency Matrices

The addition and multiplication operations of adjacency matrices are essential for the calculation of the FM closure matrix. Before defining such matrix operations, we deploy some special operators particularly for the triples in the entries of the FM adjacency matrix.

Similar to [12], we first define the following three operators: $In(\,)$, $Out(\,)$, and $Path(\,)$. These operators can be applied on the triples to extract the three components (i.e., co-inputs, co-outputs, and path) in the triples, respectively. Consider the entry $a_{2,3}$ of the FM adjacency matrix in Table 2. There is only one triple as follows:

$$(a_{2,3})_1 = <\{\tilde{x}_1\}, \phi, <\tilde{e}_1>> .$$

Then, $In((a_{2,3})_1) = \{\tilde{x}_1\}$, $Out((a_{2,3})_1) = \phi$, and

$$Path((a_{2,3})_1) = <\tilde{e}_1> .$$

Subsequently, we introduce two set operators, namely, a catenation operator $Cat(\,)$ and a truncation operator $Trnc(\,)$. The catenation operator $Cat(\,)$ catenates two ordered sets as following:

$$Cat(<a>,\ <b>) = <a, b>,$$

e.g., $Cat(<w, u, v>,\ <y, z, v, t>) = <w, u, v, y, z, v, t>$.

On the right side of the above equation, we notice that component v appears twice. In FM, an ordered set represents a series of edges (a path) and each component in the set is an edge. The truncation operator $Trnc(\,)$ is defined to remove the repetition of an edge in a path by operating on the ordered set. The operator continuously checks the elements in the set, starting at the first one. When a repetition of a component is found, the operator deletes one of the two repeated components and all other components between them. The process continues in the same way until the end of the ordered set and returns the remaining set in which all elements are unique. For example,

$$Trnc(<w, u, v, y, z, v, t>) = <w, u, v, t> .$$

Then, we can define an important operator for the multiplication of adjacency matrices, called the "$\circ$" operator. Given that $\tilde{x}_i$ links to $\tilde{x}_k$ in FM $\tilde{S}$ and $\tilde{x}_k$ links to $\tilde{x}_j$ in FM

$\tilde{T}$, we want to combine $\tilde{S}$ and $\tilde{T}$ defined on X or to combine the path from $\tilde{x}_i$ to $\tilde{x}_k$ and the path from $\tilde{x}_k$ to $\tilde{x}_j$. By manipulating the triples in adjacency matrices, the "$\circ$" operator can be used to combine the co-inputs, co-outputs, and paths of $(a_{i,k})_n$ and $(b_{k,j})_m$ in order to generate a new path from $\tilde{x}_i$ to $\tilde{x}_j$ and its co-input and co-output.

Definition 5. *Given a finite set X, FM $\tilde{S}$ with an FM adjacency matrix A and $\tilde{T}$ with an FM adjacency matrix B are defined on X. Assume that $(a_{i,k})_n$ and $(b_{k,j})_m$ are triples in A and B, respectively, such that*

$$a_{i,k} = \{(a_{i,k})_n, n = 1 \dots N\},$$
$$b_{k,j} = \{(b_{k,j})_m, m = 1 \dots M\}.$$

Then, "$\circ$" operator can define a triple or null as follows:

1. If $(a_{i,k})_n \neq \phi$ and $(b_{k,j})_m \neq \phi$, then $(a_{i,k})_n^{\circ}(b_{k,j})_m$ is a triple with the following co-input, co-output, and path

$$\begin{aligned}
&In((a_{i,k})_n^{\circ}(b_{k,j})_m)\\
&= (In((a_{i,k})_n) \cup In((b_{k,j})_m)) \backslash (Out((a_{i,k})_n) \cup \{\tilde{x}_i\})\\
&Out((a_{i,k})_n^{\circ}(b_{k,j})_m)\\
&= (Out((a_{i,k})_n) \cup Out((b_{k,j})_m) \cup \{\tilde{x}_k\}) \backslash \{\tilde{x}_j\}\\
&Path((a_{i,k})_n^{\circ}(b_{k,j})_m)\\
&= Trnc(Cat(Path((a_{i,k})_n), Path((b_{k,j})_m))).
\end{aligned}$$

2. If $(a_{i,k})_n = \phi$ or $(b_{k,j})_m = \phi$, $(a_{i,k})_n^{\circ}(b_{k,j})_m = \phi$.

With the "$\circ$" operator, the multiplication of adjacency matrices follows the same rule as the multiplication of conventional matrices, except that the multiplication of elements should obey the "$\circ$" operator.

When a matrix multiplies another matrix, the row-by-column dot product is executed by replacing real multiplication with the "$\circ$" operator. Real addition is not needed. Instead, the operation just concatenates all the triples to generate a set of triples.

In the case of the multiplication of matrix by itself, $A = B$. For instance, we examine the multiplication of the FM adjacency matrix in Table 2 by itself, especially the operation $a_{2,3}^{\circ}a_{3,5}$. According to the table, we have

$$a_{2,3} = \{< \{\tilde{x}_1\}, \phi, < \tilde{e}_1 >>\},$$
$$a_{3,5} = \{< \{\tilde{x}_4\}, \{\tilde{x}_6\}, < \tilde{e}_2 >>\}.$$

Obviously, both of the entries have only one triple. Thus, we consider the dot operation $(a_{2,3})_1^{\circ}(a_{3,5})_1$.

$$\begin{aligned}
&In((a_{2,3})_1^{\circ}(a_{3,5})_1) = (\{\tilde{x}_1\} \cup \{\tilde{x}_4\}) \backslash (\phi \cup \{\tilde{x}_2\}) = \{\tilde{x}_1, \tilde{x}_4\}\\
&Out((a_{2,3})_1^{\circ}(a_{3,5})_1) = (\phi \cup \{\tilde{x}_6\}) \cup \{\tilde{x}_3\}) \backslash \{\tilde{x}_5\} = \{\tilde{x}_3, \tilde{x}_6\}\\
&Path((a_{2,3})_1^{\circ}(a_{3,5})_1) = Trnc(Cat(<\tilde{e}_1>, <\tilde{e}_2>)) = <\tilde{e}_1, \tilde{e}_2> .
\end{aligned}$$

Thus,

$$(a_{2,5}^2)_1 = (a_{2,3})_1^{\circ}(a_{3,5})_1 = < \{\tilde{x}_1, \tilde{x}_4\}, \{\tilde{x}_3, \tilde{x}_6\}, < \tilde{e}_1, \tilde{e}_2 >>,$$

which means there is one path with a length of two from $\tilde{x}_2$ to $\tilde{x}_5$, as shown in Table 3. The path is a sequence of edges $< \tilde{e}_1, \tilde{e}_2 >$ determined by the $Path(\,)$ operation. The $Trnc(\,)$ operator is necessary to prevent the path from growing without bound as $n \to \infty$ in the calculation of A^n.

The matrix addition for FM adjacency matrix is similar to standard matrix addition. The only difference is to replace real addition with the concatenation of the triples.

ACKNOWLEDGMENTS

The author would like to thank Professor Charles E. Hoequist and the anonymous reviewers for their constructive comments.

REFERENCES

[1] S.J.H. Yang, J.J.P. Tsai, and C.-C. Chen, "Fuzzy Rule Base Systems Verification Using High-Level Petri Nets," *IEEE Trans. Knowledge and Data Eng.*, vol. 15, pp. 457-473, 2003.
[2] W. Pedrycz and F. Gomide, "A Generalized Fuzzy Petri Net Model," *IEEE Trans. Fuzzy Systems*, vol. 2, pp. 295-301, 1994.
[3] A. Chaudhury, D.C. Marinescu, and A. Whinston, "Net-Based Computational Models of Knowledge-Processing Systems," *IEEE Expert*, vol. 8, pp. 79-86, 1993.
[4] M. Ramaswamy, S. Sarkar, and Y.-S. Chen, "Using Directed Hypergraphs to Verify Rule-Based Expert Systems," *IEEE Trans. Knowledge and Data Eng.*, vol. 9, pp. 221-237, 1997.
[5] B.J. Frey, *Graphical Models for Machine Learning and Digital Communication*. London: The MIT Press, 1998.
[6] L.A. Zadeh, "Fuzzy Logic and the Calculi of Fuzzy Rules, Fuzzy Graphs, and Fuzzy Probabilities," *Computers and Math. with Applications*, vol. 37, p. 35, 1999.
[7] S.-M. Chen, "Interval-Valued Fuzzy Hypergraph and Fuzzy Partition," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 27, pp. 725-733, 1997.
[8] M. Chandwani and N.S. Chaudhari, "Knowledge Representation Using Fuzzy Deduction Graphs," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 26, pp. 848-854, 1996.
[9] S.-M. Chen, J.-S. Ke, and J.-F. Chang, "Knowledge Representation Using Fuzzy Petri Nets," *IEEE Trans. Knowledge and Data Eng.*, vol. 2, pp. 311-319, 1990.
[10] S. Anderson, J. Power, and K. Tourlas, "Reasoning in Higraphs with Loose Edges," *Proc. IEEE Symp. Human-Centric Computing Languages and Environments*, pp. 23-29, 2001.
[11] A. Basu and R.W. Blanning, "Metagraphs," *Omega, Int'l J. Management Science*, vol. 23, pp. 13-25, 1995.
[12] A. Basu and R.W. Blanning, "Metagraphs: A Tool for Modeling Decision Support Systems," *Management Science*, vol. 40, pp. 1579-1600, 1994.
[13] A. Basu and R.W. Blanning, "Workflow Analyasis Using Attributed Metagraphs," *Proc. 34th Hawaii Int'l Conf. System Sciences*, pp. 3735-3743, 2001.
[14] L.A. Zadeh, "Knowledge Representation in Fuzzy Logic," *IEEE Trans. Knowledge and Data Eng.*, vol. 1, pp. 89-100, 1989.
[15] A. Rosenfeld, "Fuzzy Graph," *Fuzzy Sets and Their Application to Cognitive and Decision Processes*, pp. 77-95, 1975.
[16] J.N. Mordeson and P.S. Nair, *Fuzzy Graphs and Fuzzy Hypergraphs*. New York: Physica-Verlag Heidelberg, 2000.
[17] I.H. Witten, A. Moffat, and T.C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Image*, second ed. Morgan Kaufmann, 1999.
[18] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison Wesley Longman, 1999.
[19] C. Faloutsos and D. Oard, "A Survey of Information Retrieval and Filtering Methods," Technical Report CS-TR-3514, Dept. of Computer Science, Univ. of Maryland, 1995.
[20] Z.-H. Tan, "Fuzzy Graph Theory, Evolutionary ANNs and Their Applications in Diagnostic Expert Systems," PhD dissertation, Shanghai Jiao Tong Univ., Shanghai, China, 1999.
[21] A. Basu and R.W. Blanning, "A Graph-Theoretic Approach to Analyzing Knowledge Bases Containing Rules, Models and Data," *Annals of Operations Research 75*, pp. 3-23, 1997.
[22] J.-S. Wang and C.S. G. Lee, "Self-Adaptive Neuro-Fuzzy Inference Systems for Classification Applications," *IEEE Trans. Fuzzy Systems*, vol. 10, pp. 790-802, 2002.

[23] L.A. Zadeh, "Soft Computing and Fuzzy Logic," *IEEE Software,* vol. 11, pp. 48-56, 1994.
[24] D.S. Yeung and E.C.C. Tsang, "Weighted Fuzzy Production Rules," *Fuzzy Sets and Systems,* vol. 88, pp. 299-313, 1997.
[25] C.G. Looney and A.R. Alfize, "Logical Controls via Boolean Rule Matrix Transformations," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 17, pp. 1077-1082, 1987.
[26] A. Alaoui, "On Fuzzification of Some Concepts of Graphs," *Fuzzy Sets and Systems,* vol. 101, pp. 363-389, 1999.
[27] A.A. Hopgood, *Knowledge-Based Systems for Engineers and Scientists.* Florida: CRC Press, 1993.
[28] H. Wang, C. Jiang, and S. Liao, "Concurrent Reasoning of Fuzzy Logical Petri Nets Based on Multi-Task Schedule," *IEEE Trans. Fuzzy Systems,* vol. 9, pp. 444-449, 2001.
[29] X.Z. Wang, S.A. Yang, E. Veloso, M. Lu, and C. McGreavy, "Qualitative Process Modeling—A Fuzzy Signed Directed Graph Method," *Computers Chemical Eng.,* vol. 19, pp. S735-S740, 1995.
[30] S. Raju, J. Zhou, and R.A. Kisner, "Fuzzy Logic Control for Steam Generator Feedwater Control," *Proc. Am. Control Conf.,* pp. 1491-1493, 1990.

Zheng-Hua Tan received the BSc and MSc degrees, in 1990 and 1996, respectively, both in electrical engineering from Hunan University, China, and the PhD degree in electronic engineering from Shanghai Jiao Tong University, China, in 1999. He is an associate professor in the Department of Communication Technology at Aalborg University (AAU), Denmark. Before joining AAU, he was a postdoctoral fellow in the Department of Computer Science at the Korea Advanced Institute of Science and Technology (KAIST), Korea. He was also an associate professor in the Department of Electronic Engineering at Shanghai Jiao Tong University, China. His research interests include computational intelligence, speech recognition, and machine learning. He has published extensively in these areas in refereed journals and conference proceedings. He is a senior member of the IEEE.