Optimization Models for Training Belief-Rule-Based Systems

Jian-Bo Yang, Jun Liu, Dong-Ling Xu, Jin Wang, and Hongwei Wang, *Associate Member, IEEE*

***Abstract*—A belief Rule-base Inference Methodology using the Evidential Reasoning approach (RIMER) has been developed recently, where a new belief rule representation scheme is proposed to extend traditional IF–THEN rules. The belief rule expression matrix in RIMER provides a compact framework for representing expert knowledge. However, it is difficult to accurately determine the parameters of a belief rule base (BRB) entirely subjectively, particularly, for a large-scale BRB with hundreds or even thousands of rules. In addition, a change in rule weight or attribute weight may lead to changes in the performance of a BRB. As such, there is a need to develop a supporting mechanism that can be used to train, in a locally optimal way, a BRB that is initially built using expert knowledge. In this paper, several new optimization models for locally training a BRB are developed. The new models are either single- or multiple-objective nonlinear optimization problems. The main feature of these new models is that only partial input and output information is required, which can be either incomplete or vague, either numerical or judgmental, or mixed. The models can be used to fine tune a BRB whose internal structure is initially decided by experts' domain-specific knowledge or common sense judgments. As such, a wide range of knowledge representation schemes can be handled, thereby facilitating the construction of various types of BRB systems. Conclusions drawn from such a trained BRB with partially built-in expert knowledge can simulate real situations in a meaningful, consistent, and locally optimal way. A numerical study for a hierarchical rule base is examined to demonstrate how the new models can be implemented as well as their potential applications.**

***Index Terms*—Belief rule base (BRB), decision making, evidential reasoning (ER), expert system, inference, multiple-objective optimization, uncertainty.**

Manuscript received August 28, 2005; revised September 11, 2006. The work was supported in part by the Natural Science Foundation of China under Grants 70540430127 and 70631003 and forms part of the projects that are supported by the U.K. Engineering and Physical Sciences Research Council under Grant GR/S85498/01 and Grant GR/S85504/01. This paper was recommended by Associate Editor J. Lambert.

J.-B. Yang is with Manchester Business School, University of Manchester, M15 6PB Manchester, U.K., and also with Huazhong University of Science and Technology, Wuhan 430074, China, and Hefei University of Technology, Hefei 230009, China (e-mail: jian-bo.yang@mbs.ac.uk).

J. Liu is with the School of Computing and Mathematics, Faculty of Engineering, University of Ulster at Jordanstown, BT37 0QB Newtownabbey, U.K. (e-mail: j.liu@ulster.ac.uk).

D.-L. Xu is with the Manchester Business School, University of Manchester, M15 6PB Manchester, U.K. (e-mail: ling.xu@mbs.ac.uk).

J. Wang is with the School of Engineering, Liverpool John Moores University, L3 3AF Liverpool, U.K. (e-mail: j.wang@livjm.ac.uk).

H. Wang is with the Institute of Systems Engineering and the Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: hwwang@mail.hust.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2007.897606

I. Introduction

It HAS become increasingly important to model and analyze decision problems using both numerical data and human judgmental information that is likely to be incomplete and can hardly be accurate. This is due to the fact that human beings hold ultimate responsibilities in most decision situations, and their preferences play an irreplaceable role in making final decisions. In addition, many decision problems are of one-off nature and are associated with the control of future events. As such, complete historical data of any statistical significance may not be available to support traditional "objective" decision analysis, and the prediction of future impact of any decision could hardly be accurate. On the other hand, analytical techniques and scientific procedures should always be employed, if at all possible and practical, to support effective, consistent, and informative decision making and to avoid making costly wrong decisions. For example, to analyze system safety in design and operations of large engineering systems of high-level innovation, experts' judgmental information may have to be used at certain stages due to lack of historical data. In modeling and assessing the quality of consumer products such as food and drinks, experts' judgments on product characteristics and consumers' perceived quality attributes must be taken into account, which cannot always be measured using hard or accurate data.

In recognition of the need to handle hybrid information with uncertainty in human decision making, a new belief rule base (BRB) inference methodology [Rule-base Inference Methodology using the Evidential Reasoning approach (RIMER)] has been proposed [28] to represent inference in rule-based systems using the evidential reasoning (ER) approach [20]–[22], [25]–[27], [29]. This methodology is developed on the basis of the Dempster–Shafer theory of evidence [3], [16], the decision theory [4], and rule-based systems. In the RIMER approach, a generic knowledge representation scheme is proposed using a belief structure. A rule base that is designed on the basis of the belief structure and called BRB is used to capture nonlinear causal relationships as well as continuity, incompleteness, and vagueness. Relevant knowledge representation parameters, including the weights of both attributes and rules, are also taken into account in the scheme. RIMER is developed on the basis of and to enhance conventional IF–THEN rule-based systems. Both RIMER and conventional rule-based systems can model and simulate explicit expert knowledge using IF–THEN rules, but RIMER can model both discrete and continuous relationships with uncertainty, such as ignorance [28], [29]. The unique feature of the RIMER method, which differentiates it from other existing modeling methods, is that it can explicitly model and infer with ignorance (incompleteness) that may exist either inside the model structure (rules) or in the input data [28]. Ignorance can be caused by incomplete or missing data, or the inability of experts or decision makers to provide complete or accurate judgments, which is common in situations where human knowledge needs to be used.

In an established BRB, the input of an antecedent is transformed into a distribution on the referential values of this antecedent. Such a distribution describes the degree of the antecedent being activated. Moreover, the antecedents of an IF–THEN rule form an overall attribute, which is called a *packet antecedent attribute*. The activation weight of a rule can be generated by aggregating the degrees to which all the antecedents in the rule are activated. In this context, an IF–THEN rule can be considered as an evaluation problem of a packet antecedent attribute being assessed to an output term in the consequent of the rule with certain degrees of belief. Finally, the inference of a rule-based system is implemented using the ER approach. RIMER has already been applied to the safety analysis of offshore systems [10], [11] and the leak detection of oil pipelines [19].

A BRB can be represented as a belief rule expression matrix, which forms a basis in the inference mechanism of RIMER and provides a framework for representing expert knowledge in a compact format. However, it is difficult to accurately determine the parameters of a BRB entirely subjectively, particularly, for a large-scale rule base with hundreds or thousands of rules. In addition, a change in rule weight or attribute weight may lead to changes in the performance of a BRB. As such, there is a need to develop a method that can generate an optimal rule expression matrix using both judgmental information and statistical data, both of which could be incomplete and vague. RIMER provides scopes and flexibility for such rule learning and updating.

In this paper, several new optimization models for locally training the parameters of a belief rule expression matrix and other knowledge representation parameters in RIMER are proposed. The new models are either single or multiple-objective *nonlinear optimization* problems. The optimization models are further extended to train hierarchical BRB systems. In a BRB, input data, attribute weights, and rule weights are combined to generate activation weights for rules, and all activated belief rules are then combined to generate appropriate conclusions using the ER algorithm. This combination process is formulated as *nonlinear objective functions* to minimize the differences between observed outputs and the outputs of a BRB. Parameter-specific limits and partial expert judgments can be formulated as constraints. The optimization problems can be solved using existing tools such as the optimization tool box that is provided in Matlab.

This paper is organized as follows: RIMER is briefly reviewed in Section II. Optimal learning models for training the parameters of an initial BRB constructed using expert knowledge are proposed in Section III. Extended optimal learning models for hierarchical BRB systems are investigated in Section IV. Section V presents a numerical study. This paper is concluded in Section VI.

II. BRB Inference Methodology Using the ER Approach (RIMER)

A. BRB Structure and Representation

The RIMER approach is summarized in this section. More details can be found in [28]. The starting point of constructing a rule-based system is to collect IF–THEN rules from human experts or through data mining based on domain-specific knowledge. A knowledge base and an inference engine are then designed to infer useful conclusions from rules and observation facts that are provided by users.

A rule-based model can be formally represented as follows:

$$R = \langle X, A, D, F \rangle$$

where $X = \{X_i; i = 1, \ldots, T\}$ is the set of antecedent attributes, with each of them taking values (or propositions) from an array of sets $A = \{A_1, A_2, \ldots, A_T\}$. $A_i = \{A_{ij}, j = 1, \ldots, Ji = |A_i|\}$ is a set of *referential* values (or propositions) for an attribute $X_i (i = 1, \ldots, T)$ with A_{ij} referred to as a *referential value*, which can be taken as different types of value, including linguistic or numerical value. The array $\{X_1 \rightarrow A_1, X_2 \rightarrow A_2, \ldots, X_T \rightarrow A_T\}$ defines a list of finite conditions representing the elementary states of a problem domain, which may be linked by AND and OR connectives. $D = \{D_n, n = 1, \ldots, N\}$ is the set of all consequents, which can be conclusions or actions. F is a logical function, reflecting the relationship between conditions and their associated conclusions. More specifically, the kth rule in a conventional rule base in forms of a conjunctive IF–THEN rule can be written as

$$R_k: \text{ IF } A_1^k \text{ and } A_2^k \text{ and } \cdots \text{ and } A_{T_k}^k \text{ THEN } D_k. \quad (1)$$

$A_i^k (\in A_i, i = 1, \ldots, T_k)$ is the referential value of the ith antecedent attribute that is used in the kth rule, and T_k is the number of the antecedent attributes that are used in the kth rule. $D_k (\in D)$ is the consequent in the kth rule.

A basic rule base is composed of a collection of such simple IF–THEN rules. To take into account a degree of belief in a consequent, attribute weights, and a rule weight, a simple IF–THEN rule is extended to a so-called *belief rule* with all possible consequents associated with belief degrees. A collection of *belief rules* consists of a BRB defined as follows:

$$\begin{aligned}
&R_k: \\
&\quad \text{IF } A_1^k \text{ and } A_2^k \text{ and} \cdots \text{and } A_{T_k}^k \\
&\quad \text{THEN} \\
&\qquad \{(D_1, \beta_{1,k}), (D_2, \beta_{2,k}), \ldots, (D_N, \beta_{N,k})\}, \left(\sum_{i=1}^{N} \beta_{i,k} \leq 1 \right), \\
&\qquad \text{with rule weight } \theta_k \\
&\qquad \text{and attribute weights } \delta_{1k}, \delta_{2k}, \ldots, \delta_{T_k k}, k \in \{1, \ldots, L\}
\end{aligned} \quad (2)$$

where $\beta_{i,k} (i \in \{1, \ldots, N\})$ is the belief degree to which D_i is believed to be the consequent if, in the kth rule, the

TABLE I
BELIEF RULE EXPRESSION MATRIX FOR A BRB

Belief Output	Input					
	$A^1(w_1)$	$A^2(w_2)$	...	$A^k(w_k)$	...	$A^L(w_L)$
D_1	$\beta_{1,1}$	$\beta_{1,2}$	...	$\beta_{1,k}$	...	$\beta_{1,L}$
D_2	$\beta_{2,1}$	$\beta_{2,2}$		$\beta_{2,k}$		$\beta_{2,L}$
⋮	⋮	⋮	...	⋮	...	⋮
D_i	$\beta_{i,1}$	$\beta_{i,2}$	...	$\beta_{i,k}$	...	$\beta_{i,L}$
⋮	⋮	⋮	...	⋮	...	⋮
D_N	$\beta_{N,1}$	$\beta_{N,2}$	...	$\beta_{N,k}$	...	$\beta_{N,L}$

input satisfies the antecedent referential value vector $\boldsymbol{A}_k = \{A_1^k, A_2^k, \ldots, A_{T_k}^k\}$. θ_k is the relative weight of the kth rule, and $\delta_{1k}, \delta_{2k}, \ldots, \delta_{T_k k}$ are the relative weights of the T_k antecedent attributes that are used in the kth rule. L is the number of all belief rules that are used in the BRB. If $\sum_{i=1}^{N} \beta_{i,k} = 1$, the kth belief rule is said to be complete; otherwise, it is incomplete. Suppose that T is the total number of antecedent attributes that are used in the rule base.

A BRB given in (2) represents functional mappings between antecedents and consequents possibly with uncertainty. It provides a more informative and realistic scheme than a simple IF–THEN rule base for knowledge representation. Note that the degrees of belief $\beta_{i,k}$ and the weights could be assigned initially by experts and then trained or updated using dedicated learning algorithms if system input and output information is available. It is the purpose of this paper to develop such optimal learning models. Once a BRB is constructed and trained, its knowledge can be used to perform inference from given inputs.

A BRB can be represented in a compact format as follows: First, the kth rule can be represented as the following vector form:

$$R_k : \text{IF } \boldsymbol{X} \text{ is } \boldsymbol{A}^k \text{ THEN } \boldsymbol{D} \text{ with belief degree } \boldsymbol{\beta}^k \tag{3}$$

where $\boldsymbol{X}$ represents the antecedent attribute vector $(X_1, X_2, \ldots, X_{T_k})$, $\boldsymbol{A}_k$ is the antecedent referential value vector $\{A_1^k, A_2^k, \ldots, A_{T_k}^k\}$, $\boldsymbol{D}$ is the consequent vector $(D_1, D_2, \ldots, D_N)$, and $\boldsymbol{\beta^k}$ is the vector of the belief degrees $(\beta_{1,k}, \beta_{2,k}, \ldots, \beta_{N,k})$ for $k \in \{1, \ldots, L\}$. The preceding rule reads that the referential value vector $\boldsymbol{A}^k$ of the antecedent attribute vector $\boldsymbol{X}$ is assessed to a consequent D_i with a belief degree of $\boldsymbol{\beta^k}$. So, this assessment can be further represented by

$$S(\boldsymbol{A}^k) = \{(D_i, \beta_{i,k}), \qquad i = 1, \ldots, N\} \tag{4}$$

which is a distribution assessment.

Suppose that all L rules are independent of each other, which means that the antecedent referential value vectors $A^1, \ldots, A^L$ are independent of each other. A BRB that is given by (4) can then be summarized using a belief rule expression matrix, as shown in Table I, where w_k is an activation weight of $\boldsymbol{A}^k$, which measures the degree to which the kth rule is weighted and activated.

B. Belief Rule Inference Using the ER Approach

In the belief rule expression matrix, the degree of activation of the kth rule w_k is calculated by

$$w_k = \frac{\theta_k \prod_{i=1}^{T_k} \left(\alpha_{i,j}^k\right)^{\bar{\delta}_i}}{\sum_{l=1}^{L} \left[\theta_l \prod_{i=1}^{T_l} \left(\alpha_{i,j}^l\right)^{\bar{\delta}_i}\right]} \text{ and } \bar{\delta}_i = \frac{\delta_i}{\max_{i=1,\ldots,T_k}\{\delta_i\}} \tag{5}$$

where $\theta_k (\in \mathbf{R}^+, k = 1, \ldots, L)$ is the relative weight of the kth rule, and $\delta_i (\in \mathbf{R}^+, i = 1, \ldots, T_k)$ is the relative weight of the ith antecedent attribute that is used in the kth rule. θ_k and δ_i can be assigned to any value in $\mathbf{R}^+$ because w_k will be eventually normalized, so that $w_k \in [0, 1]$ using (5). Without loss of generality, however, we assume that $\theta_k \in [0, 1] (k = 1, \ldots, L)$, and $\delta_i \in [0, 1] (i = 1, \ldots, T_k)$. $\alpha_{i,j}^k (i = 1, \ldots, T_k)$, which is called the *individual matching degree*, is the degree of belief to which the input for the ith antecedent attribute belongs to its jth referential value $A_{i,j}^k$ in the kth rule. $\alpha_k = \prod_{i=1}^{T_k} (\alpha_{i,j}^k)^{\bar{\delta}_i}$ is called *the normalized combined matching degree*. $\alpha_{i,j}^k$ could be generated using various ways, depending on the nature of an antecedent attribute and data that are available such as a qualitative attribute using linguistic values. To facilitate data collection, a scheme for handling various types of input information is summarized for the following cases [25], [28]:

1) quantitative attribute that is assessed using referential terms:
   a) *rule- or utility-based equivalence transformation techniques for quantitative data;*
   b) *transformation based on fuzzy membership function.*

1) quantitative attributes that are assessed using interval;
2) qualitative attributes that are assessed using subjective judgments;
3) symbolic attributes that are assessed using subjective judgments.

Note that the referential values of an attribute and the types of input information are problem specific; thus, their definitions depend on the problems in hand. More discussions about this issue can be found in [10], [11], [19], and [28].

A referential value of an attribute may in general be regarded as an evaluation grade. An input x_i^* for an attribute X_i can be equivalently transformed to a distribution over the referential values that are defined for the attribute using belief degrees as follows [25]:

$$S(x_i^*) = \{(A_{ij}, \alpha_{ij}), \qquad j = 1, \ldots, J_i\}, \qquad i = 1, 2, \ldots, T \tag{6}$$

where A_{ij} is the jth referential value of the attribute X_i, α_{ij} is the degree to which the input for X_i belongs to the referential value A_{ij} with $\alpha_{ij} \geq 0$ and $\sum_{j=1}^{J_i} \alpha_{ij} \leq 1 (i = 1, 2, \ldots, T)$, and J_i is the number of the referential values that are used for describing the ith antecedent attribute X_i. The preceding distributed assessment reads that the input for the attribute X_i is assessed to the referential value A_{ij} with the degree of belief of α_{ij} ($j = 1, 2, \ldots, J_i$ and $i = 1, 2, \ldots, T$). If a BRB has T antecedent attributes, then the rules in the rule base will be normally constructed by taking all possible combinations of the referential values for the T attributes. Hence, α_i^k in the kth rule can be generated in correspondence to (2) as follows:

$$\left(A_1^k, \alpha_1^k\right) \text{ and } \left(A_2^k, \alpha_2^k\right) \text{ and } \cdots \text{ and } \left(A_{T_k}^k, \alpha_{T_k}^k\right) \quad (7)$$

where $A_i^k \in \{A_{ij}, j = 1, \ldots, J_i\}$, and $\alpha_i^k \in \{\alpha_{ij}, j = 1, \ldots, J_i\}$.

As a result, each input can be represented as a distribution on referential values using a belief structure. The main advantage of doing so is that precise data, random numbers, and subjective judgments with uncertainty can be consistently modeled under the same framework. However, the focus of this paper is to train a BRB and its weights. The details on how to get the individual matching degree α_i^k are discussed in [25] and [28].

Having represented each belief rule using (4), the ER approach can be directly applied to combine activated belief rules and generate final conclusions as follows. Note that the kth rule is activated if $w_k > 0$. First, transform the degrees of belief β_{ik} for all $i = 1, \ldots, N$, $k = 1, \ldots, L$ into basic probability masses and then aggregate all activated rules to generate the combined degree of belief in each possible consequent D_j in D using the ER algorithm [18], [25], [26]. Using the ER analytical algorithm [18], the final conclusion $O(Y)$ that is generated by aggregating all rules that are activated by the actual input vector $\boldsymbol{x}^*$ can be represented as follows:

$$O(Y) = f(\boldsymbol{x}^*) = \{(D_j, \beta_j), \qquad j = 1, \ldots, N\} \quad (8)$$

where (9), shown at the bottom of the page, holds and

$$\mu = \left[\sum_{j=1}^{N} \prod_{k=1}^{L} \left(w_k \beta_{j,k} + 1 - w_k \sum_{i=1}^{N} \beta_{i,k}\right) - (N-1) \prod_{k=1}^{L} \left(1 - w_k \sum_{i=1}^{N} \beta_{i,k}\right)\right]^{-1}$$

where w_k is calculated by (5). Note that β_j is a function of the belief degrees $\beta_{i,k}$ $(i = 1, \ldots, N,\ k = 1, \ldots, L)$, the rule weights θ_k $(k = 1, \ldots, L)$, the attribute weights δ_i $(i = 1, \ldots, T)$, and the input vector $\boldsymbol{x}^*$.

The logic behind the approach is that, if the consequent in the kth rule includes D_i with $\beta_{j,k} > 0$ and the kth rule is activated, then the overall output must be D_i to a certain degree. The degree is measured by both the degree to which the kth rule is important to the overall output and the degree to which the antecedents of the kth rule are activated by the actual input $\boldsymbol{x}^*$.

It can be seen from (5) and (9) that the activation weight and belief degrees in each rule play an essential part in the inference procedure. The degree to which the final output can be affected is determined by the magnitude of the activation weight and the belief degrees of each rule. On the other hand, if the parameters of a BRB such as $\beta_{i,k}$, θ_k, and δ_i are not given *a priori* or only known partially or imprecisely, they could be trained using observed input and output information. This is exactly the topic for the rest of this paper.

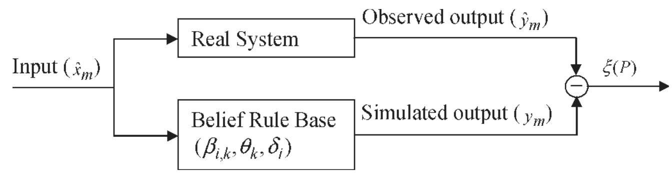


Fig. 1. Illustration of optimal learning process.

III. Optimal Learning Models and Procedures for Training BRBs

A. Generic Learning Framework

Beliefs in a rule base may initially be provided by human experts based on individuals' experiences and personal judgments, and then optimally trained if observed input–output data are available. In other words, $\beta_{i,k}$ can be trained if appropriate data become available. In addition, a change in rule weights θ_k and attribute weights δ_i may have significant impact on the performance of a BRB system [28], so they also need to be trained for achieving desirable performances.

In this section, optimization models and procedures are investigated in the RIMER framework to help search for optimally trained belief rules and weights simultaneously. Fig. 1 shows the process of training a BRB, where $\hat{x}_m$ is a given input; $\hat{y}_m$ is the corresponding observed output, either measured using instruments or assessed by experts; y_m is the simulated output that is generated by the BRB system; and $\xi(\boldsymbol{P})$ is the difference between $\hat{y}_m$ and y_m, as defined later.

It is desirable that $\xi(\boldsymbol{P})$ is as small as possible where $\boldsymbol{P}$ is the vector of training parameters including $\beta_{i,k}$, θ_k, and δ_i. This objective is difficult to achieve if a BRB is constructed using expert judgments only. Several optimal learning models are designed to adjust the parameters in order to minimize the difference between the observed output $\hat{y}_m$ and the simulated output y_m, i.e., $\xi(\boldsymbol{P})$. Such an optimally trained BRB may then be used to predict the behavior of the system. In general, the optimal learning problem can be represented as the following nonlinear (multiple objective) programming problem:

$$\begin{aligned} &\min f(\boldsymbol{P}) \\ &\text{s.t. } A(\boldsymbol{P}) = 0, \quad B(\boldsymbol{P}) \geq 0 \end{aligned} \quad (10)$$

$$\beta_j = \frac{\mu * \left[\prod_{k=1}^{L} \left(w_k \beta_{j,k} + 1 - w_k \sum_{i=1}^{N} \beta_{i,k}\right) - \prod_{k=1}^{L} \left(1 - w_k \sum_{i=1}^{N} \beta_{i,k}\right)\right]}{1 - \mu * \left[\prod_{k=1}^{L} (1 - w_k)\right]}, \qquad j = 1, \ldots, N \quad (9)$$

where $f(\boldsymbol{P})$ is the objective function, $\boldsymbol{P}$ is the training parameter vector, $A(\boldsymbol{P})$ is the equality constraint functions, and $B(\boldsymbol{P})$ is the inequality constraint functions.

In the learning process, a set of observations on the system inputs and outputs is required. In the following, we assume that a set of observation pairs $(\boldsymbol{x}, \boldsymbol{y})$ is available, where $\boldsymbol{x}$ is an input vector and $\boldsymbol{y}$ is the corresponding output vector. Both $\boldsymbol{x}$ and $\boldsymbol{y}$ can be either numerical, judgmental, or both. The format of the objective function is important for the parameter optimization. Depending on the type of input and output, the optimal learning model can be constructed in different ways, as discussed in detail in the next sections.

B. Optimal Learning Model Based on Numerical Output

In this case, it is assumed that a set of observed training data is provided in the form of M input–output pairs $(\hat{x}_m, \hat{y}_m)$ $(m = 1, \ldots, M)$, with $\hat{y}_m$ being a numerical value. The output that is shown in (8) is represented as a distribution, and its average score is given by

$$y = \sum_{j=1}^{N} u(D_j)\beta_j \tag{11}$$

where the utility (or score) of an individual consequent D_j is denoted by $u(D_j)$ [25].

Note that the score $u(D_j)$is used for characterizing an assessment but not for aggregation. $u(D_j)$ can be either given using a scale or estimated using the decision maker's preferences. Without loss of generality, suppose that the least preferred consequent having the lowest score is D_1 and the most preferred consequent having the highest score is D_N, i.e., $u(D_i) < u(D_j)$ if $i < j$ for all $i, j = 1, \ldots, N$.

Since the output in the training data set is numerical, the output of the BRB system is also given as the scores that were defined by (11); the objective of the optimal training model is thus to determine the belief rule matrix $((\beta_{i,k})_{N\times L})$, the rule weights (θ_k), the attribute weight vector (δ_i), and the scores $u(D_j)$ of the individual consequents D_j, which are denoted as $\boldsymbol{P} = \boldsymbol{P}(\beta_{i,k}, \delta_i, \theta_k, u(D_j))$, in order to minimize the total mean squared error that is defined as follows:

$$\min_{\boldsymbol{P}} \ \{\xi(\boldsymbol{P})\} \tag{12}$$

where $\xi(\boldsymbol{P}) = (1/M)\sum_{m=1}^{M}(y_m - \hat{y}_m)^2$ is the total mean squared error, $y_m = \sum_{j=1}^{N} u(D_j)\beta_j(m)$ is the expected score of the output of the BRB system for the mth input, and $\beta_j(m)$ is given by (9) for the mth input $(m = 1, \ldots, M)$. M is the number of the input–output pairs, $\hat{y}_m$ is the observed output, and $(y_m - \hat{y}_m)$ is the residual for the mth training data set.

The construction of the constraints of the learning model are given here.

1) A belief degree (subjective probability) must not be less than zero or more than one, i.e.,

$$0 \le \beta_{j,k} \le 1, \qquad j = 1, \ldots, N; \quad k = 1, \ldots, L. \tag{12a}$$

2) If the kth belief rule is complete, its total belief degree in the consequent will be equal to one, i.e.,

$$\sum_{j=1}^{N} \beta_{j,k} = 1. \tag{12b}$$

Otherwise, the total belief degree is less than one.

3) A rule weight is normalized, so that it is between zero and one, i.e.,

$$0 \le \theta_k \le 1, \qquad k = 1, \ldots, L. \tag{12c}$$

4) An attribute weight is normalized, so that it is between zero and one, i.e.,

$$0 \le \delta_i \le 1, \qquad i = 1, \ldots, T. \tag{12d}$$

5) The more preferred a consequent, the higher its score, i.e.,

$$u(D_i) < u(D_j) \text{ if } i < j, \qquad i, j = 1, \ldots, N. \tag{12e}$$

6) For qualitative output, the score (utility) of a consequent can be normalized, so that it is between zero and one, i.e.,

$$0 \le u(D_j) \le 1, \qquad j = 1, \ldots, N. \tag{12f}$$

Note that, if $\sum_{j=1}^{N} \beta_{j,k} = 1$ is required for each rule, then the trained rule base will be complete. The number of training parameters is given by $S_1 = N \times L + L + T + N$, and the number of constraints is given by $S_2 = N \times L + 2L + T + 2N - 1$, excluding the nonnegativity constraints of the parameters. The objective function that was defined in (12) is a nonlinear function of the training parameters as the $\beta_j(m)$ that is given in (9) is a nonlinear function of $\boldsymbol{P}$. Equation (12) is therefore an S_1 variable and S_2 constraint continuous nonlinear optimization problem, and can be solved using existing optimization software packages, such as the Matlab Optimization Toolbox [2].

C. Optimal Learning Model Based on Subjective Output

In this case, a set of observed training data is assumed to be composed of M input–output pairs $(\hat{x}_m, \hat{y}_m)$ $(m = 1, \ldots, M)$, with $\hat{y}_m$ being subjective and represented using a distributed assessment with different degrees of belief as follows:

$$\hat{y}_m = \left\{(D_j, \hat{\beta}_j(m)), \qquad j = 1, \ldots, N\right\} \tag{13}$$

where D_j is a referential (linguistic) term in the consequent part of a rule, and $\hat{\beta}_j(m)$ is the degree of belief to which D_j is assessed for the mth pair of observed data. This is indeed the default output format of RIMER, which provides a panoramic view about output variations. This format is useful to describe truly subjective output in a natural way.

A subjective conclusion that is generated by aggregating the activated rules can also be represented using the same referential terms as for the observed output $\hat{y}_m$ as follows:

$$y = \{(D_j, \beta_j), \qquad j = 1, \ldots, N\} \tag{14}$$

where β_j is generated by the BRB using (9) for a given input. It is desirable that, for a given input $\hat{x}_m$, the BRB system can generate an output, which is represented as $y_m = \{(D_j, \beta_j(m)), j = 1, \ldots, N\}$, which can be as close to $\hat{y}_m$ as possible. In other words, for the mth pair of the observed data $(\hat{x}_m, \hat{y}_m)$, the BRB is trained to minimize the difference between the observed belief $\hat{\beta}_j(m)$ and the belief $(\beta_j(m))$ that is generated by the BRB system for each referential term. Such a requirement is true for all pairs of the observed data. This leads to the definition of the objectives for all referential output terms as follows:

$$\min_{\boldsymbol{P}} \ \xi_j(\boldsymbol{P}) \quad j = 1, \ldots, N \tag{15a}$$

where

$$\xi_j(\boldsymbol{P}) = \frac{1}{M} \sum_{m=1}^{M} \left(\beta_j(m) - \hat{\beta}_j(m)\right)^2, \quad j = 1, \ldots, N \tag{15b}$$

$(\beta_j(m) - \hat{\beta}_j(m))$ is the residual at the mth data point, and $\boldsymbol{P} = \boldsymbol{P}(\beta_{i,k}, \delta_i, \theta_k)$ is the training parameter vector without $u(D_j)$ because D_j does not need to be quantified in this case.

The optimal training problem that is formulated in Section III-B is to minimize the numerical difference between the observed output and the simulated output. It is therefore a single-objective optimization problem. The training problem for the subjective output is a multiple objective nonlinear optimization problem with N objectives that are defined as in (15a) and (15b), $S_1(N \times L + L + T)$ training parameters as given by ($\beta_{i,k}$, θ_k, δ_i), and $S_2(N \times L + 2L + T)$ constraints, as defined in (12a)–(12d).

This multiple-objective optimization problem can be solved using various methods. For example, the following *minimax* formulation can be used to generate efficient solutions [9], [15], [17]:

$$\min_{\boldsymbol{P}} \max_{\xi_j} \left\{ \omega_j \frac{\xi_j(\boldsymbol{P}) - \xi_j^*}{\xi_j^+ - \xi_j^*}, \qquad j = 1, \ldots, N \right\}$$
$$\text{s.t. (12a)} - \text{(12d)} \tag{16}$$

where ω_j is the weighting parameter representing the relative importance of the jth referential term and can be regulated, so that $0 \le \omega_j \le 1$, $\sum_{j=1}^{N} \omega_j = 1$, or simply set to be equivalent for all terms, i.e., $\omega_j = 1/N$ for $j = 1, \ldots, N$. ξ_j^+ and ξ_j^* are the maximal feasible residual and the minimal feasible residual for the jth referential term, respectively.

The *minimax* method [2], [13], [14], [24] is also called ideal point method. It is composed of three computational steps.

Step 1) Generate a payoff matrix by solving the following single-objective optimization problems using conventional methods such as the *FMINCON* function in Matlab, i.e.,

$$\min_{\boldsymbol{P}} \ \xi_j(\boldsymbol{P})$$
$$\text{s.t. (12a)} - \text{(12d)} \tag{17}$$

for all $j = 1, \ldots, N$. Suppose that $\boldsymbol{P}^j$ is the optimum of the jth problem. Then, a payoff matrix can be formulated by defining $\xi_j^+ = \max\{\xi_j(\boldsymbol{P}^i), i = 1, \ldots, N\}$ and $\xi_j^* = \xi_j(\boldsymbol{P}^j)$ for $j = 1, \ldots, N$.

Step 2) Set the relative weight vector $\omega = (w_1, \ldots, \omega_N)$. For example, set equal weights for all terms or $\omega_j = 1/N$ for $j = 1, \ldots, N$.

Step 3) Reformulate the problem (16) as the following equivalent problem, and solve it:

$$\min \ r$$
$$\text{s.t. } \omega_j \frac{\xi_j(\boldsymbol{P}) - \xi_j^*}{\xi_j^+ - \xi_j^*} \le r, \quad j = 1, \ldots, N$$
$$\text{(12a)} - \text{(12d)}. \tag{18}$$

The process can be repeated if the relative weights need to be regulated in an interactive fashion.

D. Optimal Learning Model Based on Mixed Output

In the mixed case, a set of observed training data is composed of M input–output pairs $(\hat{x}_m, \hat{y}_m)$ $(m = 1, \ldots, M)$, with $\hat{y}_m$ being either a numerical value or a subjective judgment that is represented as a distribution. Without loss of generality, suppose that the first M_1 pairs of training data are subjective judgments, and the last $M_2 = M - M_1$ pairs of training data are numerical values. In this case, the optimization problem can be formulated as the following multiple-objective optimization problem for minimizing the differences between the outputs of the BRB system and the corresponding observed outputs in both the numerical and subjective formats:

$$\min_{\boldsymbol{P}} \{\xi_1(\boldsymbol{P}), \xi_2(\boldsymbol{P}), \ldots, \xi_N(\boldsymbol{P}); \xi(\boldsymbol{P})\}$$
$$\text{s.t. (12a)} - \text{(12f)} \tag{19}$$

where $\boldsymbol{P}$ is the training parameter vector, and ξ_j is the total mean squared error for the jth referential term that is given as follows:

$$\xi_j = \frac{1}{M_1} \sum_{m=1}^{M_1} \left(\beta_j(m) - \hat{\beta}_j(m)\right)^2, \qquad j = 1, \ldots, N. \tag{19a}$$

$\hat{\beta}_j(m)$ is the observed belief degree for the jth consequent $\boldsymbol{D}_j$ corresponding to the mth observed output in the training data set $(m = 1, \ldots, M_1)$, and $\beta_j(m)$ is the belief degree for D_j that is given by (9) corresponding to the mth input of the training data set. ξ is the total mean squared error for all the numerical data set given as follows:

$$\xi = \frac{1}{M_2} \sum_{m=M_1+1}^{M} (y_m - \hat{y}_m)^2 \tag{19b}$$

where $y_m = \sum_{j=1}^{N} u(D_j)\beta_j(m)$, $\beta_j(m)$ is given by (9) for the mth input in the numerical training data set, and $\hat{y}_m$ is the observed numerical output $(m = M_1 + 1, \ldots, M)$.

The objective that is given by (19) has $N+1$ nonlinear objective functions and can be solved using the following *minimax* formulation to generate efficient solutions:

$$\min_{\boldsymbol{P}} \max_{\xi_j} \left\{ \omega_j \frac{\xi_j(\boldsymbol{P}) - \xi_j^*}{\xi_j^+ - \xi_j^*}, \qquad j = 1, \ldots, N+1 \right\}$$
$$\text{s.t. (12a)} - \text{(12f)} \tag{19c}$$

where ξ_j^+ and ξ_j^* are defined as follows. The computational steps of the *minimax* method for solving the multiple-objective optimization problem that is given in (19) are summarized here.

Step 1) Solve the following single-objective optimization problems:

$$\min_{\boldsymbol{P}} \xi_j(\boldsymbol{P})$$
$$\text{s.t. (12a)} - \text{(12d)} \tag{20a}$$

for $j = 1, \ldots, N$. Suppose that $\boldsymbol{P}^j$ is the optimum of the jth problem. In addition, solve the following single-objective problem and formulate a payoff table:

$$\min_{\boldsymbol{P}} \xi(\boldsymbol{P})$$
$$\text{s.t. (12a)} - \text{(12f)}. \tag{20b}$$

Suppose that $\boldsymbol{P}^{N+1}$ is the optimum of the preceding problem (20b). Then, a payoff table can be formulated by defining $\xi_j^+ = \max\{\xi_j(\boldsymbol{P}^i),\ i = 1, \ldots, N+1\}$ and $\xi_j^* = \xi_j(\boldsymbol{P}^j)$ for $j = 1, \ldots, N+1$.

Step 2) Set the relative weight vector $\omega = (w_1, \ldots, \omega_N; \omega_{N+1})$. For example, one possible way to set the weight vector is given as follows:

$$\omega_1 = \cdots = \omega_N = \frac{M_1}{N} \quad \omega_{N+1} = M_2. \tag{21}$$

This means that the importance of one type of objective function depends on the number of training data sets corresponding to this type of objective function.

Step 3) Reformulate the single-objective problem (16) into the following equivalent problem, and solve it:

$$\min r$$
$$\text{s.t. } \omega_j \frac{\xi_j(\boldsymbol{P}) - \xi_j^*}{\xi_j^+ - \xi_j^*} \le r, \qquad j = 1, \ldots, N+1$$
$$\text{(12a)} - \text{(12f)}. \tag{22}$$

The process can be repeated if the relative weights need to be regulated in an interactive fashion.

A BRB system can be initialized either by experts or arbitrarily, depending on initial information that is available. In the former case, the domain-specific knowledge of experts is used to assign the initial belief-rule-base matrix and the weights, which are refined using the preceding models when more information becomes available. In the latter case, the initial BRB matrix and the weights are arbitrarily generated and then refined using the preceding models. However, a trained BRB with arbitrary initialization may generate intuitively wrong conclusions if not all rules are trained, which may be the case if the observed data set does not cover all possible regions where the BRB is designed to operate. This problem is discussed in more detail in Section V-D.

The optimization models that are proposed in this section are for a single-level BRB system. The same principle can be applied to handle more complex multilevel systems, in which the consequent of a rule may be used as the antecedent of another rule. The optimization models for hierarchical BRB system are investigated in the next section.

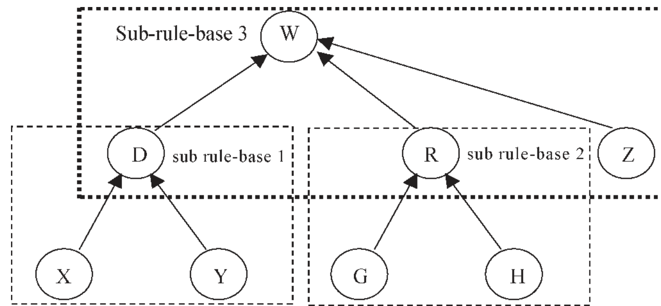


Fig. 2. Simple hierarchical structure.

IV. Optimal Learning Models for Hierarchical BRBs

The rules of a knowledge base for a complex decision-making problem can be of a hierarchical structure. In general, a bottom-up approach can be used to solve such a problem. Pieces of evidence for the bottom-level rule bases are aggregated into evidence for the second lowest level rule bases, which is in turn aggregated to produce evidence for higher level rule bases. Fig. 2 shows a hierarchical knowledge base having three subrule bases: subrule base 1, subrule base 2, and subrule base 3.

Information is propagated from the bottom-level states (X, Y, G, H, and Z) up to the goal state W. Subrule base 1 and subrule base 2 are assumed to be independent of each other. The output of subrule base 1 D and the output of subrule base 2 R, together with the independent input state Z, are taken as the input states to subrule base 3. Each of the three subrule bases constitutes a basic rule base and can be dealt with using the RIMER approach, where subrule base 1 and subrule base 2 are solved first, followed by the solution of subrule base 3. The optimal training for this hierarchical rule-based system is to systematically adjust the belief rule expression matrices and the weights of all the three subrule bases simultaneously using training data for the inputs at the bottom level and the overall output at the top level.

A. Construction of a Hierarchical BRB System

Given an input vector $\boldsymbol{X} = \{X_1, \ldots, X_n\}$, a hierarchical BRB system having Z levels can be described as follows. Without loss of generality, suppose that each upper level subrule base has several independent inputs and is linked to a lower level subrule base. Other models with an upper subrule base linked to multiple lower level subrule bases can be represented in a similar way.

Suppose that the first level has n_1 independent input variables $X_1, \ldots, X_{n_1}$ and an output variable Y_1, which is later used as an intermediate variable in the hierarchy. The subrule base can be expressed as follows:

$$R_l^1 : \text{IF } X_1 \text{ is } A_1^l \text{ and } \cdots \text{ and } X_{n_1} \text{ is } A_{n_1}^l, \quad \text{THEN } Y^1 \text{ is } \left\{\left(D_j^1, \beta_{j,l}^1\right), \quad j = 1, \ldots, N_1\right\} \tag{23a}$$

and the output can be described as follows:

$$O(Y^1) = \left\{\left(D_j^1, \beta_{j,l}^1\right), \quad j = 1, \ldots, N_1\right\} \tag{23b}$$

where $l = 1, \ldots, L_1$, L_1 is the total number of rules at the first level of the hierarchical system, D_j^1 is the jth referential value for assessing Y^1, N_1 is the total number of referential values for Y^1, and $\beta_{j,l}^1$ is the belief degree to which the output Y^1 is assessed to D_j^1 in the lth rule.

Suppose that the ith level subrule base $(i > 1)$ has n_i $(n_i \geq 1)$ independent input variables X_{K_i+1} and an intermediate input variable Y^{i-1}, which are used to construct the following rules:

$$R_l^i : \text{IF } X_{K_i+1} \text{ is } A_{K_i+1}^l \text{ and} \cdots \text{and } X_{K_i+n_i} \text{ is } A_{K_i+n_i}^l \text{ and } Y^{i-1} \text{ is } \left\{\left(D_j^{i-1}, \beta_j^{i-1}\right), \ j = 1, \ldots, N_{i-1}\right\} \text{ THEN } Y^i \text{ is } \left\{\left(D_j^i, \beta_{j,l}^i\right), \ j = 1, \ldots, N_i\right\} \tag{24}$$

where $K_i = \sum_{j=1}^{i-1} n_j$, $l = 1, \ldots, L_i$, L_i is the total number of rules at the ith level, N_i is the total number of referential values for Y^i, and $\beta_{j,l}^i$ $(j = 1, \ldots, N_i)$ is the belief degree to which the output could be assessed to D_j^i in the lth rule at the ith level. A_k^l $(k = K_i + 1, \ldots, K_i + n_i)$ is the referential value for assessing X_j in the lth rule at the ith level. D_j^{i-1} $(j = 1, \ldots, N_{i-1})$ is the referential value for assessing Y^{i-1}, and β_j^{i-1} is inferred from the $(i-1)$th level subrule base. Consequently, the output for Y^i is given by

$$O(Y^i) = f_i\left(X_{K_i+1}, \ldots, X_{K_i+n_i}, Y^{i-1}\right) = \left\{\left(D_j^i, \beta_j^i\right), \ j = 1, \ldots, N_i\right\}. \tag{25}$$

At the top level $i = Z$ with $\sum_{j=1}^{Z} n_j = n$, the overall output Y^Z can be represented as follows:

$$O(Y^Z) = \left\{\left(D_j^Z, \beta_j^Z\right), \ j = 1, \ldots, N_Z\right\} \tag{26}$$

where D_j^Z is a referential value for describing Y^Z, β_j^Z is the belief degree to which the output could be D_j^Z, and N_Z is the total number of referential value for assessing Y^Z.

In the hierarchical inference process, the first-level subrule base aggregates n_1 input variables $X_1, \ldots, X_{n_1}$ into one output variable Y^1, which is then used as an input to the second-level subrule base. In the second level, other n_2 independent input variables $X_{n_1+1}, \ldots, X_{n_1+n_2}$ and the intermediate variable Y^1 are aggregated into the second-level output variable Y^2, which is then used as an input to the third-level subrule base. The process continues until the overall output is generated at the top level.

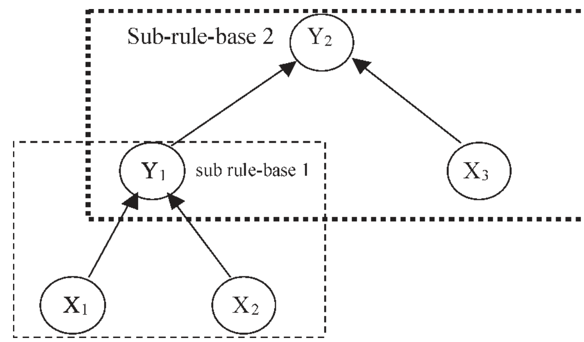


Fig. 3. Three-input two-level structure.

For illustration purposes, we use three input variables (X_1, X_2, X_3), one intermediate output Y^1, one overall output Y^2, and two subrule bases to demonstrate the construction of a hierarchical rule-based system step by step, as shown in Fig. 3.

Step 1) Before the inference can be started, input values need to be equivalently transformed and represented in terms of the referential values using belief degrees to which the values belong to the referential values. Such a belief degree is regarded as the individual matching degree corresponding to each rule, as shown in (6).

Step 2) Using (5) and (9), the output of the first-level subrule base is given by

$$O(Y^1) = f_1(X_1, X_2) = \left\{\left(D_j^1, \beta_j^1\right), \ j = 1, \ldots, N_1\right\} \tag{27}$$

where D_j^1 is the jth referential value for Y^1, N_1 is the number of the referential values for Y^1, and β_j^1 is the inferred belief to which Y^1 is believed to be assessed to D_j^1.

Step 3) The output of the second-level subrule base is given by $S(Y^2) = f_2(X_3, Y^1)$. Suppose that the input X_3 is transformed to a distribution on the referential values for X_3 using belief degrees as follows:

$$S(X_3) = \{(A_{3j}, \alpha_{3j}), \ j = 1, \ldots, J_3\} \tag{28}$$

where A_{3j} is the jth referential value for X_3, α_{3j} is the degree to which X_3 belongs to the referential value A_{3j}, and J_3 is the number of referential values that are used for assessing X_3. The second-level subrule base for generating Y^2 is given by

$$\text{IF } X_3 \text{ and } Y^1 \text{ THEN } Y^2$$

where X_3 is given by (28) and $O(Y^1)$ is given by (27).

Based on (7), the input corresponding to the lth rule is expressed as follows:

$$\left(A_3^l, \alpha_3^l\right) \text{ and } \left(D_l^1, \beta_l^1\right) \tag{29}$$

where $l = 1, \ldots, L_2$, $A_3^l \in \{A_{3j}, j = 1, \ldots, J_3\}$, $\alpha_3^l \in \{\alpha_{3j}, j = 1, \ldots, J_3\}$, $D_l^1 \in \{D_j^1, j = 1, \ldots, N_1\}$, and $\beta_l^1 \in \{\beta_j^1; j = 1, \ldots, N_1\}$. Therefore, the activation weight for the lth rule of the second subrule base is calculated by (5) as follows:

$$w_l^2 = \frac{\theta_l^2 \times \alpha_l^2}{\sum_{k=1}^{L_2} [\theta_k^2 \times \alpha_k^2]} \tag{30}$$

where $\alpha_k^2 = (\alpha_3^k)^{\bar{\delta}_{X_3}^2} \times (\beta_k^1)^{\bar{\delta}_{Y_1}^2}$ is the combined matching degree in the lth rule.

It is clear from the preceding process that the output of the first-level subrule base [or $O(Y^1)$] becomes the input of the second-level subrule base. Equation (30) shows that the belief distribution $O(Y^1)$ is directly combined with the belief distribution $S(X_3)$ to obtain the activation weight for each rule of the second-level subrule base.

Step 4) The overall hierarchical output Y^2 is generated as follows:

$$O(Y^2) = f(X_1, X_2, X_3) = f_2\left(f_1(X_1, X_2), X_3\right) \tag{31}$$

where f_1 and f_2 are each defined by (5) and (9), respectively.

B. Optimal Learning Models for a Hierarchical System

Case 1) Suppose that the overall output of a hierarchical rule based system is in numerical form. Then, the objective function of the optimal learning model can be expressed as follows:

$$\min\{\xi\} \tag{32}$$

where $\xi = (1/M)\sum_{m=1}^{M}(y_m - \hat{y}_m)^2$, $y_m = \sum_{j=1}^{N_z} u(D_j^Z)\beta_j^Z(m)$ is the expected utility (or score) of the output $O(Y^Z)$ corresponding to the mth input, and $\beta_j^Z(m)$ is given by (26) for the mth input in the training data set $(m = 1, \ldots, M)$. M is the number of the training data sets, $\hat{y}_m$ is the observed output, and $(y_m - \hat{y}_m)$ is the residual at the mth data set.

Similar to (12a)–(12f), the constraints for a hierarchical rule-based system are given by

$$0 \le \beta_{j,l}^i \le 1, \quad j = 1, \ldots, N_i; \quad l = 1, \ldots, L_i; \quad i = 1, \ldots, Z \tag{32a}$$

$$\sum_{j=1}^{N_i} \beta_{j,l}^i = 1, \quad l = 1, \ldots, L_i; \quad i = 1, \ldots, Z \tag{32b}$$

$$0 \le \delta_t^i \le 1, \quad t = 1, \ldots, T_i; \quad i = 1, \ldots, Z \tag{32c}$$

$$0 \le \theta_l^i \le 1, \quad l = 1, \ldots, L_i; \quad i = 1, \ldots, Z \tag{32d}$$

$$0 \le u\left(D_j^Z\right) \le 1, \quad j = 1, \ldots, N_Z \tag{32e}$$

$$u\left(D_j^Z\right) < u\left(D_j^Z\right) \quad \text{if } i < j,\ i, j = 1, \ldots, N_Z \tag{32f}$$

where T_i is the number of antecedent attributes in the ith level.

Case 2) Suppose that the output is in the form of subjective judgment. Then, the multiobjective function of the optimal learning model can be expressed as follows:

$$\min_{\boldsymbol{P}} \max_{\{\xi_j\}} \{\xi_j(\boldsymbol{P}),\ j = 1, \ldots, N_Z\}$$
$$\text{s.t. (32a)} - \text{(32d)} \tag{33}$$

where

$$\xi_j(\boldsymbol{P}) = \frac{1}{M}\sum_{m=1}^{M}\left(\beta_j^Z(m) - \hat{\beta}_j^Z(m)\right)^2, \quad j = 1, \ldots, N_Z. \tag{33a}$$

$\boldsymbol{P}$ is the parameter vector, $\hat{\beta}_j^Z(m)$ is the expected belief corresponding to the individual consequent D_j^Z, and $(\beta_j^Z(m) - \hat{\beta}_j^Z(m))$ is the residual at the mth data point.

Case 3) Suppose that the output is in the form of both numerical value and subjective judgment. Then, the multiobjective function of the optimal learning model can be expressed as follows:

$$\min_{\boldsymbol{P}} \{\xi_1(\boldsymbol{P}), \xi_2(\boldsymbol{P}), \ldots, \xi_{N_Z}(\boldsymbol{P}), \xi(\boldsymbol{P})\}$$
$$\text{subject to (32a)} - \text{(32f)} \tag{34}$$

where $\xi(\boldsymbol{P})$ is given by (32), and $\xi_j(\boldsymbol{P})$ $(j = 1, \ldots, N_Z)$ are given by (33a). This multiple-objective optimization problem can also be solved using the *minimax* approach, as discussed in the previous section.

V. Numerical Study

A. Problem Description

A numerical example is studied in this section to demonstrate the implementation and potential applications of the proposed training models. Several case studies for applying the BRB systems and the training models to real-world problems are reported in other published papers [10], [11], [19]. This example is based on a fuzzy rule base for an exploratory expert system that is discussed by Hodges *et al.* [5] and was also used to demonstrate how a BRB system could be built [28]. More detail about this example can be found in these references.

In this example, it is aimed to determine confidence degrees to which the system believes that a container may contain graphite. So, the output variable is the confidence degree to which a container contains graphite. The input variables that are defined in the exploratory expert system include the following:

- accuracy of the weight measurement;
- degree to which the calculated density is consistent with graphite;
- observer's experience;
- observer's confidence that the real-time radiography (RTR) shows graphite.

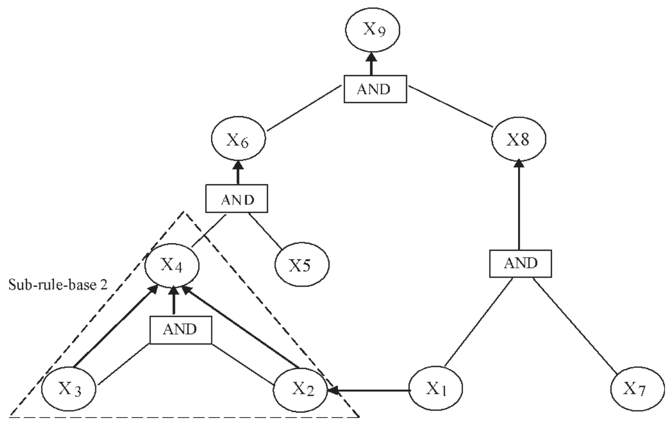


Fig. 4. Hierarchical structure of a small exploratory expert system.

For illustration purposes, each of these input variables and the output variable are all assessed using linguistic terms such as *high* (H), *medium* (M), or *low* (L). However, they could be assessed in various ways such as different linguistic terms, numerical values, random numbers, intervals, or fuzzy sets [28]. The system structure is shown in Fig. 4. The parameters in Fig. 4 are defined as follows:

X_1 Observer's experience.
X_2 Accuracy of the fill height determination.
X_3 Accuracy of the weight measurement.
X_4 Accuracy of the calculated density.
X_5 Consistency of the calculated density with graphite.
X_6 Confidence that the density indicates graphite.
X_7 Observer's confidence that the RTR shows graphite.
X_8 Confidence that the RTR shows graphite.
X_9 Confidence that the container contains graphite.

This example uses four input variables (X_1, X_3, X_5, X_7) and four intermediate variables (X_2, X_4, X_6, X_8) to predict X_9 in terms of qualitative linguistic terms. The expert knowledge is coded as IF–THEN rules that are hierarchically organized in five subrule bases [28].

The BRB is constructed on the basis of the original IF–THEN rule base that is provided by Hodges *et al.* [5]. Belief degrees were initially assigned to each rule by the researchers by examining the original rule base and perceiving the expert's way of making the judgments [28]. For example, in terms of the original rule base in [5], the expert's final confidence in the verification of the graphite code tends to be quite low in the presence of a piece of negative information, such as inexperience of the observer. Such human reasoning processes can be closely imitated by, for example, assigning weights to antecedent attributes and/or by adjusting belief degrees in the consequents of rules in a systematic manner [28]. This is one of the prominent features of a BRB, which can also be optimally trained in the ways that were investigated in the previous sections and is to be demonstrated in this section.

B. Training Data

To train the BRB for the preceding example, 12 sets of training data are used and listed in Table II [28]. In Table II, the original input and output data were all provided as numerical numbers, as shown in columns 2–6. In the last column, the output data are equivalently transformed into the belief structures. This is accomplished by using the information transformation technique for numerical data [25], [28], with the three linguistic terms *high* (H), *medium* (M), and *low* (L) quantified by 1, 0.5, 0, respectively, i.e., $u(H) = 1$, $u(M) = 0.5$, and $u(L) = 0$, respectively. For the third data set, for example, the observed output X_9 was originally expressed by 0.6, i.e., $X_9 = 0.6$. It can also be expressed using the distribution (belief structure) $S(X_9 = 0.6) = \{(H, 0.2), (M, 0.8), (L, 0)\}$, as shown in the last column of Table II. The two expressions (assessments) are said to be equivalent in the sense that the expected value of the distribution is equal to 0.6, i.e., $X_9 = u(H) \times 0.2 + u(M) \times 0.8 + u(L) \times 0.2 = 1 \times 0.2 + 0.5 \times 0.8 + 0 \times 0 = 0.6$.

TABLE II
TRAINING DATA SET (INPUT AND OBSERVED OUTPUT VALUES)

Data Set	Given input				Observed output (X_9)	
	X_3	X_1	X_5	X_7	Numerical value	Judgment with belief
1	0.98	1.00	1.00	1.00	1.00	{(H, 1), (M, 0), (L, 0)}
2	0.98	0.80	0.80	0.80	0.90	{(H, 0.8), (M, 0.2), (L, 0)}
3	0.98	0.80	0.20	0.80	0.60	{(H, 0.2), (M, 0.8), (L, 0)}
4	0.98	0.40	0.40	0.80	0.30	{(H, 0), (M, 0.6), (L, 0.4)}
5	0.98	0.40	0.60	0.80	0.40	{(H, 0), (M, 0.8), (L, 0.2)}
6	0.98	1.00	0.00	0.80	0.20	{(H, 0), (M, 0.4), (L, 0.6)}
7	0.98	0.00	0.00	0.00	0.00	{(H, 0), (M, 0), (L, 1)}
8	0.98	1.00	1.00	0.20	0.30	{(H, 0), (M, 0.6), (L, 0.4)}
9	0.98	0.40	0.60	0.20	0.10	{(H, 0), (M, 0.2), (L, 0.8)}
10	0.98	0.60	0.40	0.20	0.20	{(H, 0), (M, 0.4), (L, 0.6)}
11	0.98	0.80	0.20	0.20	0.20	{(H, 0), (M, 0.4), (L, 0.6)}
12	0.98	0.80	0.80	0.20	0.40	{(H, 0), (M, 0.8), (L, 0.2)}

TABLE III
TRANSFORMATION OF INPUT TRAINING DATA

Antecedent attribute	Input data	Transformed input data		
		high (α_{iH})	*medium* (α_{iM})	*low* (α_{iL})
X_1	0.4	0	0.8	0.2
X_3	0.98	0.96	0.04	0
X_5	0.6	0.2	0.8	0
X_7	0.8	0.6	0.4	0

The given values for the four input variables (X_1, X_3, X_5, X_7) can also be transformed into the belief structures with respect to the defined three linguistic terms using the information transformation technique. In training data set 5, for example, the input data for the input variable X_1 is given by 0.4, i.e., $X_1 = 0.4$. This can be equivalently transformed to the distribution (belief structure) of {(*high*, 0), (*medium*, 0.8), (*low*, 0.2)}. By equivalence, we mean that the expected value of the distribution is equal to 0.4, i.e., $X_1 = u(H) \times 0 + u(M) \times 0.8 + u(L) \times 0.2 = 1 \times 0 + 0.5 \times 0.8 + 0 \times 0.2 = 0.4$. Similarly, for input data set 5, $X_3 = 0.98$, $X_5 = 0.6$, and $X_7 = 0.8$ can be equivalently transformed to {(*high*, 0.96), (*medium*, 0.04), (*low*, 0)}, {(*high*, 0.2), (*medium*, 0.8), (*low*, 0)}, and {(*high*, 0.6), (*medium*, 0.4), (*low*, 0)}, respectively, as shown in Table III. The other given input training data sets can be transformed in a similar way.

C. Hierarchical Rule Based System and Optimal Learning Models

Note that the training data are given in two types of output form, i.e., numerical values and judgments with beliefs, as shown in Table II. They are used to demonstrate the optimization models for numerical output that is given in (32) and for judgmental output that is given in (33), respectively.

As discussed in [28], the BRB system has five subrule bases, so $Z = 5$. In the first subrule base, for example, since there is only one input (X_1), we have $T_1 = 1$ and one attribute weight (δ_1^1). As its input (X_1) and output (X_2) are both assessed using three linguistic terms, we have $N_1 = 3$ and three possible belief rules (or $L_1 = 3$) for this subrule base, which are defined as follows:

R_1 : IF X_1 is $high$
THEN X_2 is $\{(high, \beta_{H1}^1), (medium, \beta_{M_1}^1), (low, \beta_{L_1}^1)$
R_2 : IF X_1 is $medium$
THEN X_2 is $\{(high, \beta_{H2}^1), (medium, \beta_{M_2}^1), (low, \beta_{L_2}^1)$
R_3 : IF X_1 is low
THEN X_2 is $\{(high, \beta_{H3}^1), (medium, \beta_{M3}^1), (low, \beta_{L3}^1)$

with the rule weights that are given by θ_1^1, θ_2^1, and θ_3^1. The training parameters for this subrule base include nine belief degrees $(\beta_{Hl}^1, \beta_{Ml}^1, \beta_{Ll}^1, l = 1, 2, 3)$, three rule weights $(\theta_l^1, l = 1, 2, 3)$, one attribute weight (δ_1^1), and the utilities of the linguistic terms $(u(H), u(M), u(L))$, which need to be trained if numerical output values are given and the single-objective model (32) is used.

In the second subrule base, there are two input variables X_2 and X_3. So, we have $T_1 = 2$ and two attribute weights (δ_1^2, δ_2^2). Since its inputs and the output (X_4) are all assessed using three linguistic terms, we have $N_2 = 3$ and nine possible belief rules (or $L_2 = 9$) for this subrule base, which are defined as follows:

R_4: IF X_1 is $high$ AND X_2 is $high$
THEN X_4 is $\{(high, \beta_{H4}^2), (medium, \beta_{M4}^2), (low, \beta_{L4}^2)$
R_5: IF X_1 is $high$ AND X_2 is $medium$
THEN X_4 is $\{(high, \beta_{H5}^2), (medium, \beta_{M5}^2), (low, \beta_{L5}^2)$
R_6: IF X_1 is $high$ AND X_2 is low
THEN X_4 is $\{(high, \beta_{H6}^2), (medium, \beta_{M6}^2), (low, \beta_{L6}^2)$
R_7: IF X_1 is $medium$ AND X_2 is $high$
THEN X_4 is $\{(high, \beta_{H7}^2), (medium, \beta_{M7}^2), (low, \beta_{L7}^2)$
R_8: IF X_1 is $medium$ AND X_2 is $medium$
THEN X_4 is $\{(high, \beta_{H8}^2), (medium, \beta_{M8}^2), (low, \beta_{L8}^2)$
R_9: IF X_1 is $medium$ AND X_2 is low
THEN X_4 is $\{(high, \beta_{H9}^2), (medium, \beta_{M9}^2), (low, \beta_{L9}^2)$
R_{10}: IF X_1 is low AND X_2 is $high$
THEN X_4 is
$\{(high, \beta_{H10}^2), (medium, \beta_{M10}^2), (low, \beta_{L10}^2)$
R_{11}: IF X_1 is low AND X_2 is $medium$
THEN X_4 is
$\{(high, \beta_{H11}^2), (medium, \beta_{M11}^2), (low, \beta_{L11}^2)$
R_{12}: IF X_1 is low AND X_2 is low
THEN X_4 is
$\{(high, \beta_{H12}^2), (medium, \beta_{M12}^2), (low, \beta_{L12}^2)$

with the rule weights θ_4^2, θ_5^2, θ_6^2, θ_7^2, θ_8^2, θ_9^2, θ_{10}^2, θ_{11}^2, and θ_{12}^2, respectively. The training parameters for the second subrule base include 27 belief degrees $(\beta_{Hl}^2, \beta_{Ml}^2, \beta_{Ll}^2), l = 4, \ldots, 12)$, nine rule weights $(\theta_l^2, l = 4, \ldots, 12)$, and two attribute weights (δ_1^2, δ_2^2).

Similarly, for the other three subrule bases, we have $T_3 = T_4 = T_5 = 2$; $N_3 = N_4 = N_5 = 3$, and $L_3 = L_4 = L_5 = 9$. Their possible rules can be listed in the same way. The belief degrees $\beta_{j,l}^i$ in the consequents of the output variables, the rule weights θ_l^i, and attribute weights δ_t^i are the parameters to be trained.

The optimization model framework is given in Section IV-B. To solve the single-objective model that is given by (32), existing optimization tools such as the *FMINCON* function in Matlab can be used. In this example, the calculations are done using a medium-scale algorithm in *FMINCON*, and all the results are generated using the sequential quadratic programming, quasi-Newton, and line-search methods that are provided in *FMINCON*. To solve the multiobjective model that is given by (33), existing optimization tools such as the *FMINIMAX* function in Matlab can be used.

D. Experimental Results Based on Numerical Data

If beliefs and weights are initially given by experts, we consider two cases.

1) Initial beliefs and weights are assigned and manually tuned by experts (denoted as *Expert* 1).
2) Initial beliefs and weights are assigned but not fine tuned by experts (denoted as *Expert* 2).

If beliefs are not given by experts, their values in each rule are initialized randomly with their sum added to one (denoted as *random*).

The initial beliefs and weights for *Expert* 1, *Expert* 2, and *random* are listed in Table IV. The trained beliefs and weights for *Expert* 1, *Expert* 2, and *random* are given in Table VI. The initial and trained utilities of the linguistic terms for X_9 are shown in Table V. In the example, the error tolerance is set to 0.000001, and the maximum iteration is set to 60 to avoid dead loop in the optimal learning process. The test results are illustrated in Fig. 5(C1)–(C4), where the comparisons are shown between the observed output and the simulated output that is generated using the trained BRB system, with the output assigned and fine tuned by experts (*Expert* 1), assigned but not fine tuned by experts (*Expert* 2), or generated randomly (*random*).

It is evident that there is great difference between the initial system output and the observed output that is shown in Fig. 5(C4). This difference is caused by the fact that the initial output is randomly generated. There are also significant differences between the initial system output and the observed output that is shown in Fig. 5(C2) and (C3). Although the difference in Fig. 5(C2) is quite small due to the small size of this BRB, it would be difficult and time consuming to manually generate relatively accurate beliefs and weights for large

TABLE IV
INITIAL BRBS AND RULE WEIGHTS

		Random				***Expert 2***				***Expert 1***			
Rule base	Rule Number	H	M	L	Rule Weight	H	M	L	Rule Weight	H	M	L	Rule Weight
1	1	0.2404	0.3760	0.3836	0.5314	1	0	0	1	1	0	0	1
	2	0.7595	0.1522	0.0882	0.1813	0	1	0	1	0	1	0	1
	3	0.3627	0.6221	0.0152	0.5019	0	0	1	1	0	0	1	1
2	4	0.4120	0.4337	0.1542	0.8702	1	0	0	1	1	0	0	1
	5	0.4135	0.4454	0.1411	0.0269	1	0	0	0.6	0.3	0.7	0	0.2
	6	0.0975	0.3820	0.5206	0.5195	1	0	0	0.2	0	0.3	0.7	0.8
	7	0.3825	0.3438	0.2737	0.1923	0	1	0	1	0.4	0.6	0	1
	8	0.5003	0.3608	0.1389	0.7157	0	1	0	0.6	0	1	0	0.4
	9	0.5291	0.3465	0.1243	0.2507	0	1	0	0.2	0	0.1	0.9	1
	10	0.4598	0.4493	0.0909	0.9339	0	0	1	1	0.1	0.3	0.6	1
	11	0.0561	0.6535	0.2904	0.1372	0	0	1	1	0	0.3	0.7	1
	12	0.0629	0.7736	0.1636	0.5216	0	0	1	1	0	0	1	1
3	13	0.4304	0.1204	0.4492	0.0324	1	0	0	1	1	0	0	1
	14	0.1786	0.3554	0.4660	0.7339	1	0	0	0.7	0.3	0.7	0	0.7
	15	0.2426	0.4190	0.3384	0.5365		0	1	1	0	0	1	1
	16	0.1296	0.7075	0.1629	0.2760	1	0	0	0.7	0.3	0.7	0	0.7
	17	0.3072	0.3216	0.3712	0.3685	0	1	0	0.7	0	1	0	0.7
	18	0.2912	0.3385	0.3703	0.0129	0	0	1	1	0	0	1	1
	19	0.7735	0.2239	0.0026	0.8892	0	0	1	1	0	0	1	1
	20	0.5209	0.3386	0.1406	0.8660	0	0	1	1	0	0	1	1
	21	0.1548	0.2354	0.6098	0.2542	0	0	1	1	0	0	1	1
4	22	0.4752	0.2660	0.2588	0.4078	1	0	0	1	1	0	0	1
	23	0.4214	0.1411	0.4374	0.0527	1	0	0	1	0.4	0.6	0	1
	24	0.4291	0.0108	0.5600	0.9418	1	0	0	1	0	1	0	1
	25	0.4093	0.3554	0.2353	0.1500	0	1	0	0.6	0.2	0.8	0	1
	26	0.0754	0.7486	0.1761	0.3844	0	1	0	0.6	0	1	0	0.4
	27	0.5806	0.0169	0.4025	0.3111	0	1	0	0.6	0	0.2	0.8	1
	28	0.4283	0.4087	0.1630	0.1685	0	0	1	0.2	0.1	0.3	0.6	0.2
	29	0.4584	0.2497	0.2919	0.8966	0	0	1	0.2	0	0.2	0.8	1
	30	0.3251	0.3494	0.3255	0.3227	0	0	1	0.2	0	0	1	1
5	31	0.1096	0.2142	0.6762	0.6851	1	0	0	1	1	0	0	1
	32	0.5373	0.0110	0.4518	0.5102	0	1	0	0.7	0.2	0.8	0	0.6
	33	0.2769	0.5512	0.1719	0.7140	0	1	0	0.8	0.1	0.2	0.7	1
	34	0.0426	0.3124	0.6449	0.5152	0	1	0	0.7	0.2	0.8	0	0.6
	35	0.4318	0.2627	0.3054	0.6059	0	1	0	1	0	1	0	0.6
	36	0.8781	0.1207	0.0012	0.9667	0	0	1	0.7	0	0.1	0.9	1
	37	0.5418	0.0069	0.4514	0.8221	0	1	0	0.8	0.1	0.2	0.7	1
	38	0.1222	0.4915	0.3863	0.3178	0	0	1	0.7	0	0.1	0.9	1
	39	0.0093	0.5341	0.4566	0.5877	0	0	1	1	0	0	1	1

systems. As shown in Fig. 5(C1), the trained BRB system can simulate the real system with great accuracy, which is strongly nonlinear.

This study shows that a BRB system can be initially constructed using partial knowledge, either numerical or judgmental, which, if not wrong in general, could help generate logical conclusions even in untrained regions. With the accumulation of new knowledge, the system can be progressively trained and updated to better mimic a real system. Note that the training data in this example do not cover the full range of regions where the BRB is designed to operate. For example, Rules 19–21 were not trained at all as the belief degrees and rule weights of these rules remain unchanged after the training, in whichever ways the BRB was initialized, as marked in gray in both Tables IV and VI. When activated, such rules that are untouched during training could lead to irrational conclusion if they were initially assigned randomly or without care. For example, if the BRB is randomly built initially as in Table IV, then the three rules would lead to intuitively wrong conclusions if they were later activated after the training of the rule base, as shown in Table VI. If there are a number of such rules in a BRB that are untouched during a training process, which are not initially assigned by experts, such a trained BRB will still be very likely to generate irrational conclusions. On the other hand, if a BRB is initialized using appropriate expert knowledge such as in case 2 (*Expert* 2) and case 3 (*Expert* 1) shown in Tables IV and VI, the untrained rules would not lead to intuitively wrong conclusions if activated. In addition, note that the optimization models may have multiple optima, and using expert initialization could help to achieve a trained BRB that is closer to expert expectations, leading to better results in extrapolation.

TABLE V
INITIAL AND TRAINED UTILITY OF THE CONSEQUENT TERM $\{H, M, L\}$ FOR X_9

	$u(H)$	$u(M)$	$u(L)$
Initial value	1	0.5	0
Random-trained	0.9543	0.4320	0
Expert 1-trained	0.9757	0.3271	0.0003
Expert 2-trained	1	0.7947	0.0008

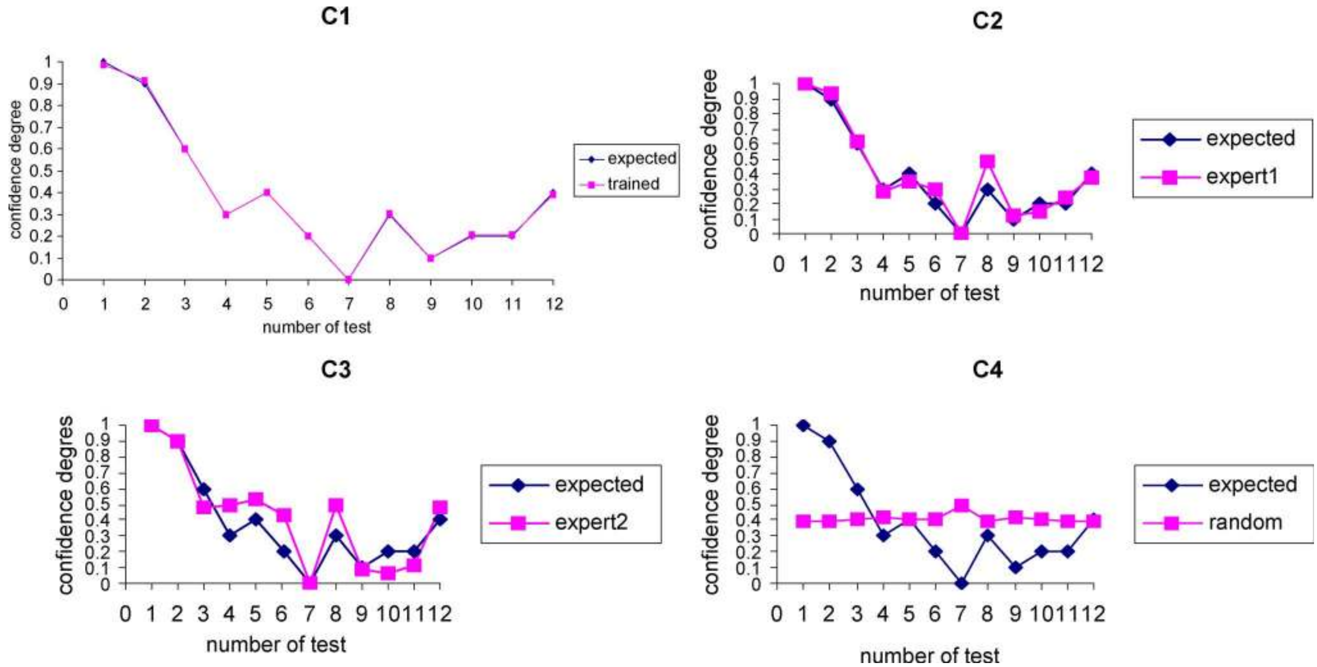


Fig. 5. Comparisons of test results.

TABLE VI
TRAINED BRBS AND RULE WEIGHTS

		Random- Trained (numerical case)				Expert 2- Trained (numerical case)				Expert 1-Trained (numerical case)				Expert 1-Trained (subjective case)			
RB	RN	H	M	L	RW	H	M	L	RW	H	M	L	RW	H	M	L	RW
	1	0.2117	0.3628	0.4254	0.5418	0.9871	0.0093	0.0036	1.0000	0.9799	0.0071	0.0000	0.9920	1.0000	0	0	1.0000
1	2	0.7752	0.0398	0.1849	0.2530	0	1.0000	0	1.0000	0.0193	0.9511	0.0032	0.8076	0.0884	0.9116	0	0.7941
	3	0.3742	0.6019	0.0239	0.4753	0	0.0048	0.9952	0.9852	0.0008	0.0418	0.9968	0.9933	0	0.0041	0.9959	0.9902
	4	0.4333	0.0095	0.5572	0.9649	0.8403	0.0609	0.0989	0.9191	0.8795	0.0474	0.0731	0.8592	0.9993	0.0007	0	1.0000
	5	0.4099	0.4456	0.1445	0.0100	0.6000	0.2075	0.1925	0.1537	0.2687	0.7223	0.0091	0.4303	0.2713	0.7287	0	0.4896
	6	0.1703	0.3131	0.5166	0.4627	0.9614	0.0207	0.0179	0.1057	0.0238	0.3370	0.6392	0.7439	0.0295	0.3815	0.5890	0.6862
	7	0.3769	0.2737	0.3494	0.2229	0.3614	0.6379	0.0008	0.8078	0.2659	0.5945	0.1396	0.6857	0.3886	0.6102	0.0012	0.7897
2	8	0.7175	0.2193	0.0632	0.3445	0.2432	0.7011	0.0557	0.7217	0.0114	0.9839	0.0047	0.6038	0	1.0000	0	0.6608
	9	0.5888	0.3832	0.0281	0.3152	0	0.9631	0.0369	0.0594	0.0011	0.1230	0.8759	0.9771	0.0009	0.0011	0.9980	0.9952
	10	0.2021	0.4952	0.3027	0.7630	0	0.3971	0.6029	0.9069	0.1858	0.1604	0.6537	0.8782	0.0015	0.2814	0.7171	0.9735
	11	0.2416	0.6703	0.0881	0.1086	0.1938	0	0.8062	0.8310	0.2895	0.2075	0.5030	0.4398	0.0564	0.4380	0.5056	0.5530
	12	0.0822	0.9092	0.0086	0.7559	0	0	1.0000	1.0000	0.0001	0	0.9999	0.9999	0	0	1.0000	1.0000
	13	0.4357	0.1311	0.4332	0.1644	0.9858	0.0120	0.0022	1.0000	0.9758	0.0030	0.0212	0.9969	0.9896	0.0058	0.0046	1.0000
	14	0.0439	0.7676	0.1884	0.6941	1.0000	0	0	0.7151	0	0.8886	0.1114	0.5664	0.1855	0.8102	0.0042	0.5573
	15	0.1057	0.3709	0.5234	0.5809	0.0022	0	0.9978	0.9927	0.0001	0.0095	0.9905	0.9933	0.0001	0.0064	0.9936	0.9945
	16	0.1284	0.7192	0.1524	0.2995	0.9999	0.0001	0	0.6999	0.3030	0.6970	0	0.6925	0.3038	0.6962	0	0.6902
3	17	0.3049	0.3257	0.3694	0.3685	0.0025	0.9975	0	0.6909	0	1.0000	0	0.7055	0	1.0000	0	0.7013
	18	0.2911	0.3385	0.3704	0.0145	0.0001	0	0.9999	0.9997	0	0.0003	0.9997	0.9998	0	0.0002	0.9998	0.9998
	19	0.7735	0.2239	0.0026	0.8892	0	0	1.0000	1.0000	0	0	1.0000	1.0000	0	0	1.0000	1.0000
	20	0.5209	0.3386	0.1406	0.8660	0	0	1.0000	1.0000	0	0	1.0000	1.0000	0	0	1.0000	1.0000
	21	0.1548	0.2354	0.6098	0.2542	0	0	1.0000	1.0000	0	0	1.0000	1.0000	0	0	1.0000	1.0000
	22	0.3749	0.2275	0.3977	0.2603	1.0000	0	0	0.9777	0.9953	0	0.0047	0.9996	1.0000	0	0	1.0000
	23	0.3943	0.1241	0.4816	0.0051	0.9996	0.0002	0.0002	0.9967	0.1961	0.7587	0.0451	0.8763	0.2823	0.7177	0	0.9227
	24	0	0	1.0000	0.9867	0.9963	0.0029	0.0008	0.9919	0.0001	0.9998	0.0001	0.9994	0	1.0000	0	0.9997
	25	0.3763	0.3882	0.2355	0.1980	0	1.0000	0	0.8564	0.3181	0.6590	0.0229	0.6323	0.2602	0.7396	0.0002	0.6760
4	26	0.0599	0.9174	0.0227	0.9139	0	1.0000	0	0.5963	0	0.9969	0.0031	0.8479	0.0002	0.9998	0	0.6142
	27	0.6534	0.2298	0.1169	0.0864	0	1.0000	0	0.6053	0	0.2212	0.7788	0.9931	0.0002	0.2088	0.7910	0.9877
	28	0.4262	0.3989	0.1749	0.1704	0.0711	0.0469	0.8821	0.0092	0.0499	0.2710	0.6791	0.1463	0.0647	0.2990	0.6363	0.0322
	29	0.8375	0.1429	0.0196	0.6431	0.0017	0.0003	0.9980	0.1963	0.0627	0.1782	0.7591	0.9584	0.0530	0.1729	0.7741	0.9744
	30	0.2794	0.2921	0.4285	0.1934	0.0005	0.0010	0.9986	0.1991	0.0044	0.0028	0.9928	0.9952	0.0005	0.0013	0.9983	0.9985
	31	0.0177	0.0325	0.9498	0.7757	1.0000	0	0	0.9845	1.0000	0	0	0.9167	0.9912	0.0064	0.0023	0.9625
	32	0	0	1.0000	0.5965	0.2419	0.7573	0.0007	0	0.2698	0.7209	0.0093	0.7615	0.1684	0.8316	0	0.8493
	33	0.1563	0.3679	0.4757	0.3245	0	0.8656	0.1344	0.7234	0.0005	0.0517	0.9478	0.9839	0.0869	0.2721	0.6410	0.9527
	34	0.1479	0.1493	0.7028	0.0303	0.6497	0.2787	0.0716	0.7840	0.1147	0.7644	0.1209	0.5424	0.1979	0.8021	0	0.6973
	35	0.0382	0	0.9618	0.8421	0.1337	0.4584	0.4079	0.9304	0.0405	0.9584	0.0011	0.2224	0.0032	0.9959	0.0009	0.3684
5	36	0.9768	0.0232	0	0.9960	0	0	1.0000	0.6478	0.0299	0	0.9701	0.9968	0	0	1.0000	0.9979
	37	1.0000	0	0	0.9746	0.1551	0.3579	0.4870	0.4552	0.0232	0.1962	0.7807	0.9997	0.1050	0.2069	0.6880	0.9535
	38	0	0.3961	0.6039	0.9048	0.0279	0.0031	0.9690	0.9831	0.0342	0.1024	0.8634	0.9720	0.0112	0.0921	0.8967	0.9879
	39	0.4959	0.4729	0.0311	0	0	0	1.0000	0.9863	0.0003	0	0.9996	0.9685	0	0	1.0000	0.9928

TABLE VII
COMPARISON BETWEEN OBSERVED AND TRAINED RESULTS

Number of training	1	2	3	4	5	6
Observed output	*1.0000*	*0.9000*	*0.6000*	*0.3000*	*0.4000*	*0.2000*
Trained from expert 1	0.9874	0.9100	0.5981	0.2994	0.4019	0.2007
Trained from random	0.9539	0.9106	0.5984	0.2982	0.3996	0.1952
Number of training	**7**	**8**	**9**	**10**	**11**	**12**
Observed output	*0.0000*	*0.3000*	*0.1000*	*0.2000*	*0.2000*	*0.4000*
Trained from expert 1	0.0011	0.3048	0.0970	0.2038	0.2027	0.3897
Trained from random	0.0061	0.3121	0.0990	0.2031	0.2137	0.3835

TABLE VIII
EFFECT OF EXPERT KNOWLEDGE ON THE LEARNING RESULTS

Iterations	FuncCount	LR-Expert	LR-Random
10	2359	0.000987251	0.0361497
20	4719	0.000581982	0.00979918
30	7079	0.000384411	0.00551611
40	9439	0.000178554	0.000818101
50	11800	7.71473e-005	0.000445483
57	13452	3.70721e-005	0.000272463
61	14159		0.000243479

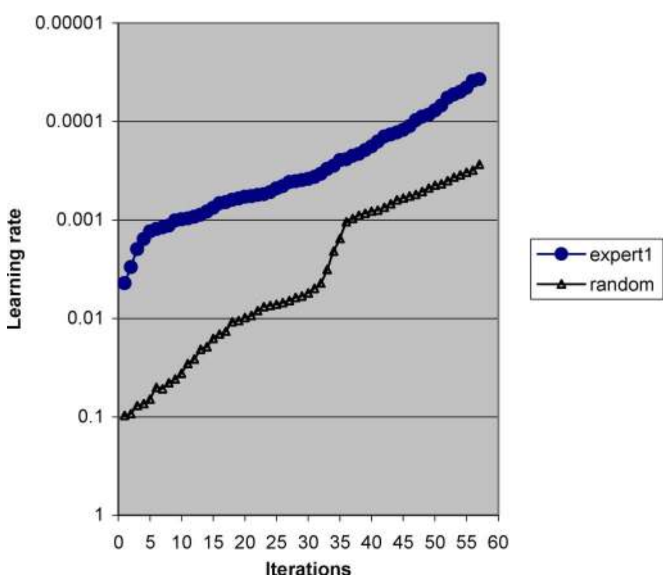


Fig. 6. Learning rate comparison between *Expert* 1 and *random*.

E. Effect of Experts' Knowledge on the Experimental Results

In Table VII, the learning results for two cases are compared, in which either expert initialization (*Expert* 1) or random initialization (*random*) is used. In both cases, the final numerical results for X_9 are similar after training, although the costs to achieve these results are rather different, so are the generated rules. In Table VIII, "Iterations" means the number of iterations that were taken in the training process, "FuncCount" is the number of function evaluations, "LR-Expert" is the learning rate based on initial beliefs that are provided by *Expert* 1, and "LR-Random" is the learning rate based on randomly generated initial beliefs.

Fig. 6 shows the learning rate comparisons between *Expert* 1 and *random*. It is clear from these experimental results that *prior* expert knowledge can help to improve the performance of the training process and also avoid generating obvious illogical conclusions in cases where available training data do not cover the full range of regions in which the trained system is supposed to operate.

TABLE IX
COMPARISON OF TEST RESULTS BASED ON JUDGMENTAL OBSERVATIONS

Test number		1	2	3	4	5	6
Trained Output	H	0.9912	0.8249	0.3138	0.0275	0.0422	0.1264
	M	0.0064	0.1731	0.6765	0.5749	0.7329	0.2657
	L	0.0023	0.0020	0.0097	0.3976	0.2249	0.6078
Observed output	H	1.0000	0.8000	0.2000	0	0	0
	M	0	0.2000	0.8000	0.6000	0.8000	0.4000
	L	0	0	0	0.4000	0.2000	0.6000
Test number		7	8	9	10	11	12
Trained Output	H	0.0000	0.1596	0.0118	0.0171	0.0521	0.1606
	M	0.0000	0.4724	0.1981	0.3752	0.3266	0.5353
	L	1.0000	0.3680	0.7901	0.6077	0.6213	0.3041
Observed output	H	0	0	0	0	0	0
	M	0	0.6000	0.2000	0.4000	0.4000	0.8000
	L	1.0000	0.4000	0.8000	0.6000	0.6000	0.2000

F. Experimental Results Based on Judgmental Assessments

Table IX shows the comparison between the observed (transformed) assessments and the trained results. In this example, the error tolerance for the single-objective optimal training model of each subrule base is set to 0.0001, and the maximum iteration number is set to 60. The error tolerance for the multiobjective optimal training model is set to 0.001, and the maximum iteration is set to 60 to avoid dead loop in the optimal training process. The initial beliefs and weights are based on *Expert* 1.

The trained outputs in terms of belief degrees to the linguistic terms are very close to the observed outputs. This shows the capability of the BRB system to simulate real systems using mixed numerical values (input in this case) and judgmental information (output in this case).

G. Validation of Learning Results Based on Numerical Data

To validate the trained models, the available data are partitioned into a training data set and a test data set. The training data set is used for training the system parameters. The trained BRB system is then used to generate outputs for the test input data.

For illustration purposes, the first six sets of data, as shown in rows 1–6 in Table II, are used as the training data for parameter estimation. The initial beliefs and rule weights are given by *Expert* 1 (Column *Expert* 1 in Table IV). The initial attribute weights are all set to be equivalent. After training, the remaining six sets of data, as shown in rows 7–12 in Table II, are used for validation. The observed and simulated outputs are listed in Table X. The test results in Table X indicate that over 90% of the results are correct within the specified tolerance of 0.05.

TABLE X
OBSERVED AND SIMULATED VALUES OF THE OUTPUT

Test data set	7	8	9	10	11	12
simulated output	0.0073	0.4866	0.1419	0.162	0.251	0.3858
observed output	0	0.3	0.1	0.2	0.2	0.4

VI. Conclusion and Discussion

Several new optimization models were proposed in this paper for training the parameters of a BRB system that is initially built using expert knowledge. These optimal learning models provide a systemic mechanism to enhance the capability of the recently developed BRB inference methodology (RIMER) to simulate systems where both expert knowledge and partial input–output data are available. It can be used to handle a range of knowledge representation schemes, thereby facilitating the construction of various types of BRB systems. A numerical example for a hierarchical rule-based expert system was examined to demonstrate how these new models can be implemented.

There are several features in the proposed models. First, due to the use of belief rules, the RIMER method provides an analytical description of relationships between system's inputs and outputs that could be discrete or continuous; complete or incomplete; linear, nonlinear or nonsmooth; or their mixtures. This enables a BRB system to act as a generic functional mapping from system inputs to outputs and allows powerful learning techniques to be used for parameter training and system updating on the basis of both numerical data and human judgements. A hierarchical BRB system could be constructed to simulate complex systems. Second, a BRB system provides a non-black-box simulator that can explicitly represent expert's domain-specific knowledge as well as common-sense judgments and thus can avoid generating obviously irrational conclusions. Once new knowledge becomes available, the new learning models can be used to fine tune the system in order to achieve better performances in simulating a real system. Third, the new models are capable of accommodating input and output information that can be numerical or judgmental or both, thereby providing a flexible way to represent and a rigorous procedure to deal with hybrid information to arrive at rational conclusions. Finally but by no means least importantly, the new models can be used to process incomplete or vague information. It may be appropriate to claim that the unique feature of these training models is their capabilities to handle ignorance, as powered by the ER algorithm-based rule aggregation process, in order to support the development of BRBs with incomplete or mixed training information. The aforementioned features allow BRB systems, which are equipped with the new learning models, to be capable of simulating a range of real systems.

The BRB system methodology and the proposed optimal training models have been applied to develop knowledge-based systems using both numerical data and judgmental information, such as analyzing the safety of engineering systems [10], [11] and detecting leaks in oil pipelines [19]. In such applications, various types of information may be available, either measured using instruments or generated by trained experts or untrained consumers. The proposed methodology can provide a nonblack-box simulator with learning capability to support retrofit product design and manufacture, and faulty diagnosis. For instance, in the development of a BRB system for leak detection in oil pipelines [19], human experts specialized in fine tuning the parameters of a model for detecting oil pipeline leaks, provided initial incomplete knowledge to establish a preliminary BRB system. The initial system worked reasonably well but could not provide precise predictions when there were sizable leaks in the pipeline operating data. By using the proposed optimal learning methods to train the preliminary belief rules, the BRB system can learn from the online-collected pipeline operating data, the relationship between leak sizes, and the pipeline flow and pressure readings. For pipeline leak detection systems based on mass balance principles, identifying the relationship between flow difference and pressure changes is very important and involves intensive parameter tuning. It is a time-consuming process and pipeline specific. The self-learning capability of the BRB system can significantly reduce the parameter tuning time and improve the performance of the system.

However, the validation of training results needs to be extensively investigated for larger systems. Note that the local optimization models that are proposed in this paper are used to fine tune the parameters of a BRB that is initially built using expert knowledge. They are not yet equipped with the ability to automatically identify and rectify conflicting rules that may exist. In our current applications [10], [11], [19], experts are expected to inspect whether there are obvious irrational or conflicting rules based on their domain-specific knowledge. If conflicting rules are identified by experts or certain relationships among rules are required, then additional constraints could be added to the optimization models to avoid conflicts and to meet the requirements. For large systems or systems with little prior expert knowledge, however, there may be a need to develop new global optimization models to check inconsistency in a BRB.

Note that, for a complex real-world problem, prior knowledge may be limited, which may lead to the construction of an incomplete or even inappropriate initial BRB structure. If there are too many belief rules in an initial rule base, the training task may become too complicated to handle, or it is possible to result in overfitting. On the other hand, if there are too few rules in the initial rule base, this may lead to underfitting, and consequently, the system may not be able to achieve overall optimal performance. To achieve an overall optimal BRB, it may not be sufficient to just statistically tune parameters on given rules, but the structure of a BRB system may need to be adjusted as well. The final performance of a supervised BRB learning system therefore depends on both its system structure and system parameters. However, generating globally optimal system performance in general circumstances would also require the development of global optimization models.

The current local optimization models are constructed as conventional nonlinear single- or multiple-objective optimization problems. The numbers of variables, constrains, and objectives of such problems are analyzed as in the paragraphs following equation (12f) and problem (15). So, these problems can be readily solved using existing nonlinear optimization software such as the Matlab optimization tool box, *LINDO*, and Excel solver. Note that if global optimization models are developed in the future with the structure of BRBs also subject to changes, then there may be a need to deal with discrete and even nonsmooth optimization problems. To solve such problems, it would be beneficial to design hybrid algorithms by combining mathematical programming techniques with adaptive search or heuristic methods, such as genetic algorithms or simulated annealing.

Fuzzy neural networks (FNNs) are a powerful tool to model systems with inaccurate input–output data. Indeed, it is claimed to be capable of modeling any nonlinear systems without requiring assumptions on functional structures among system inputs and outputs. However, it is in essence a black-box modeling approach, and its internal structure does not allow the explicit representation of subjective expert knowledge. As such, it entirely relies on the quality of data samples for its prediction performance. RIMER is not developed to compete against FNNs in situations where the latter one can perform well, for example, where a complete range of data samples of high quality is available. Rather, RIMER is designed to allow direct intervention of human experts in deciding the internal structure of a BRB. The optimization training models and suggested algorithms that are reported in this paper are used to fine tune a BRB system that is initially built using human knowledge. As such, they provide performance-enhancing mechanism for RIMER, which could be used wherever partial input–output data are available. The benefits of the strategy in allowing direct expert intervention were discussed in the last paragraph of Section V-D. It is possible that the construction of a BRB could be entirely based on human knowledge in the form of IF–THEN rules with beliefs and does not necessarily depend on the availability of numerical input–output data. On the other hand, it may be suggested that RIMER could also be used as a black-box simulator using input–output data without human intervention in structuring a BRB, which however requires further extensive research to ensure that statistically correct and intuitively meaningful belief rules can be generated.

ACKNOWLEDGMENT

The authors would like to thank the four anonymous referees for their constructive comments, which have helped to improve the quality of this paper to its current standard.

REFERENCES

[1] M. Campi, "Performance of RLS identification algorithms with forgetting factor: A Phi-mixing approach," *J. Math. Syst., Estim. Control*, vol. 4, no. 3, pp. 1–25, 1994.

[2] T. Coleman, M. A. Branch, and A. Grace, *Optimization Toolbox—For Use With MATLAB*. Natick, MA: The Mathworks Inc., 1999.

[3] A. P. Dempster, "A generalization of Bayesian inference," *J. R. Stat. Soc.*, vol. 30, no. 2, pp. 205–247, 1968.

[4] J. S. Dyker, P. C. Fishburn, R. E. Steuer, J. Wallenius, and S. Zionts, "Multiple criteria decision making, multiattribute utility theory: The next ten years," *Manag. Sci.*, vol. 38, no. 5, pp. 645–654, May 1992.

[5] J. Hodges, S. Bridge, and S. Y. Yie, "Preliminary results in the use of fuzzy logic for radiological waste characterization expert system," Mississippi State Univ., Mississippi State, MS, Tech. Rep. MSU-960626, Jun. 26, 1996.

[6] C. L. Huang and A. S. Masnd, *Multiple Objective Decision Making Methods and Applications. A State-of-Art Survey*. Berlin, Germany: Springer-Verlag, 1979.

[7] R. Isermann, K.-H. Lachmann, and D. Matko, *Adaptive Control Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[8] J. Koski and R. Silvennoinen, "Norm methods and partial weighting in multicriterion optimization of structures," *Int. J. Numer. Methods Eng.*, vol. 24, no. 6, pp. 1101–1121, 1987.

[9] G. P. Liu, J. B. Yang, and J. F. Whidborne, "Multiobjective optimization and control," in *Engineering Systems Modelling and Control Series*. Hertfordshire, U.K.: Res. Stud. Press Limited, 2002.

[10] J. Liu, J. B. Yang, J. Wang, H. S. Sii, and Y. M. Wang, "Fuzzy rule-based evidential reasoning approach for safety analysis," *Int. J. Gen. Syst.*, vol. 33, no. 2/3, pp. 183–204, Apr.–Jun. 2004.

[11] J. Liu, J. B. Yang, J. Wang, and H. S. Sii, "Engineering system safety analysis and synthesis using fuzzy rule-based evidential reasoning approach," *Qual. Reliab. Eng. Int.*, vol. 21, no. 4, pp. 387–411, 2005.

[12] L. Ljung and S. Gunnarsoon, "Adaptation, tracking and system identification—A survey," *Automatica*, vol. 26, no. 1, pp. 7–21, Jan. 1990.

[13] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidiscipl. Optim.*, vol. 26, no. 6, pp. 369–395, Apr. 2004.

[14] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA: Kluwer, 1999.

[15] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multiobjective Optimization*. New York: Academic, 1985.

[16] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton Univ. Press, 1976.

[17] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*. New York: Wiley, 1986.

[18] Y. M. Wang, J. B. Yang, and D. L. Xu, "Environmental impact assessment using the evidential reasoning approach," *Eur. J. Oper. Res.*, vol. 174, no. 3, pp. 1885–1913, Nov. 2006.

[19] D. L. Xu, J. Liu, J. B. Yang, G. P. Liu, J. Wang, I. Jenkinson, and J. Ren, "Inference and learning methodology of belief-rule-based expert system for pipeline leak detection," *Expert Syst. Appl.*, vol. 32, no. 1, pp. 103–113, 2006.

[20] J. B. Yang and M. G. Singh, "An evidential reasoning approach for multiple attribute decision making with uncertainty," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 1, pp. 1–18, Jan. 1994.

[21] J. B. Yang and P. Sen, "A general multi-level evaluation process for hybrid MADM with uncertainty," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 10, pp. 1458–1473, Oct. 1994.

[22] J. B. Yang and P. Sen, "Multiple attribute design evaluation of large engineering products using the evidential reasoning approach," *J. Eng. Des.*, vol. 8, no. 3, pp. 211–230, 1997.

[23] J. B. Yang and D. L. Xu, *Intelligent Decision System Via Evidential Reasoning*. Cheshire, U.K.: IDSL, 1999. Version 1.1.

[24] J. B. Yang, "Minimax reference point approach and its application for multiobjective optimization," *Eur. J. Oper. Res.*, vol. 126, no. 3, pp. 90–105, 2000.

[25] J. B. Yang, "Rule and utility based evidential reasoning approach for multi-attribute decision analysis under uncertainties," *Eur. J. Oper. Res.*, vol. 131, no. 1, pp. 31–61, 2001.

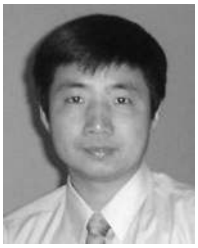
[26] J. B. Yang and D. L. Xu, "On the evidential reasoning algorithm for multiple attribute decision analysis under uncertainty," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 32, no. 3, pp. 289–304, May 2002.

[27] J. B. Yang and D. L. Xu, "Nonlinear information aggregation via evidential reasoning in multi-attribute decision analysis under uncertainty," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 32, no. 3, pp. 376–393, May 2002.

[28] J. B. Yang, J. Liu, J. Wang, H. S. Sii, and H. W. Wang, "A generic rule-base inference methodology using the evidential reasoning approach—RIMER," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 2, pp. 266–285, Mar. 2006.

[29] J. B. Yang, Y. M. Wang, and D. L. Xu, "The evidential reasoning approach for MCDA under both probabilistic and fuzzy uncertainties," *Eur. J. Oper. Res.*, vol. 171, no. 1, pp. 309–343, 2006.

[30] L. Z. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.



Jian-Bo Yang received the B.Eng. and M.Eng. degrees in control engineering from the North Western Polytechnic University, Xi'an, China, in 1981 and 1984, respectively, and the Ph.D. degree in systems engineering from Shanghai Jiao Tong University, Shanghai, China, in 1987.

He is currently a Professor of decision and system sciences and the Director of the Decision Sciences Research Centre in the Manchester Business School, University of Manchester, Manchester, U.K. He is also a Specially Appointed Professor of the Huazhong University of Science and Technology, Wuhan, China, and the Hefei University of Technology, Hefei, China. Prior to his current appointment, he was a Faculty Member with the University of Manchester Institute of Science and Technology (1998–2004, 1990), the University of Birmingham, Birmingham, U.K., (1995–1997), the University of Newcastle upon Tyne, Newcastle, U.K. (1991–1995), and Shanghai Jiao Tong University (1987–1989). In the past two decades, he has been conducting research in multiple-criteria decision analysis under uncertainty, multiple-objective optimization, intelligent decision support systems, hybrid quantitative and qualitative decision modeling using techniques from operational research, artificial intelligence, and systems engineering and dynamic system modeling, simulation, and control for engineering and management systems. His current applied research includes design decision-making, risk modeling and analysis, production planning and scheduling, quality modeling and evaluation, supply chain modeling and supplier assessment, and the integrated evaluation of products, systems, projects, policies, etc. His current research has been supported by the EPSRC, EC, DEFRA, SERC, HKRGC, NSFC, and industry. He has published widely and developed several software packages in these areas including the Windows-based intelligent decision system via evidential reasoning.

Jun Liu received the B.Sc. and M.Sc. degrees in applied mathematics, and the Ph.D. degree in information engineering and control from the Southwest Jiaotong University, Chengdu, China, in 1993, 1996, and 1999, respectively.

He is currently a Lecturer in the School of Computing and Mathematics, Faculty of Engineering, University of Ulster at Jordanstown, Newtownabbey, U.K. Prior to his current appointment, he was a Research Associate in the Manchester Business School, University of Manchester, Manchester, U.K. He was also a Postdoctoral Staff Member at the Belgian Nuclear Research Center (SCK–CEN), Mol, Belgium (from March 2000 to February 2002). His research is focused on algebraic and nonclassical logic (many-valued logic and fuzzy logic), automated reasoning theory and methods, fuzziness and uncertainty theory and applications, intelligent decision support systems and evaluation analysis, information fusion, and data combinations. His current research interests include intelligent decision analysis and support under uncertainties for design decision making and risk and safety analysis in engineering fields.

Dong-Ling Xu received the B.Eng. degree in electrical engineering from Hunan University, Changsha, China, in 1983 and the M.Eng. and Ph.D. degrees in system control engineering from Shanghai Jiao Tong University, Shanghai, China, in 1986 and 1988, respectively.

She is currently a Lecturer in decision and system sciences in the Manchester Business School, University of Manchester, Manchester, U.K. She was a Senior Software Engineer in an IT company from 1995 to early 2001, a Postdoctoral Research Associate at the University of Newcastle upon Tyne, Newcastle, U.K., from 1993 to 1995, and a Lecturer and then an Associate Professor at East China University, Shanghai, from 1988 to 1993. She has developed several interactive web-based decision support tools and helped to develop a Windows-based intelligent decision system. These tools are used in a wide range of assessment activities such as safety and risk assessment, organizational self-assessment in quality management, product evaluation, financial investment policy assessment, and supplier evaluation. In the past 15 years, she has worked in or in close collaboration with industry, applying advanced theory and technology to product quality modeling, fault diagnosis, and risk management. She has published a book and more than 50 papers, many in high-quality international journals including the *European Journal of Operational Research*, the *Journal of Multiple Criteria Decision Analysis*, and *Decision Support Systems*.

Dr. Xu has published papers in the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS.

Jin Wang received the B.Sc. degree in marine automation from Dalian Maritime University, Dalian, China, in 1983 and the M.Sc. degree in marine engineering and the Ph.D. degree (via staff registration) in marine safety engineering from the University of Newcastle upon Tyne, Newcastle, U.K., in 1989 and 1994, respectively.

He is currently a Professor of marine technology in the School of Engineering, Liverpool John Moores University, Liverpool, U.K. He has been involved in marine and offshore safety research for the past 14 years with support from the EPSRC, EU, HSE, etc. His research interests include safety- and reliability-based design and operations of large marine and offshore systems, probabilistic and nonprobabilistic safety analysis and decision making, port safety assessment, and analysis of safety-critical systems.

Hongwei Wang (M'04–A'05) received the Ph.D. degree in systems engineering from the Huazhong University of Science and Technology, Wuhan, China in 1993.

He is currently with the Huazhong University of Science and Technology, where he is a Professor and the Head of the Department of Control Science and Engineering, and the Director of the Institute of Systems Engineering. His research interests include supply-chain management, decision-making and simulation of logistics management, and decision support systems.