A Novel Construction and Inference Methodology of Belief Rule Base

QINGSHUANG HU1, CHENGHAI LI2, YANLI LU2, AND SONG LI2

1Graduate School, Air Force Engineering University, Xi’an 710051, China
2Air and Missile Defense College, Air Force Engineering University, Xi’an 710051, China

Corresponding author: Song Li (li_song77@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61703426, and in part by the Young Talent Fund of University Association for Science and Technology in Shaanxi, China, under Grant 20190108.

ABSTRACT
The belief rule base model performs well in continuous systems but not in piecewise systems, and it suffers from a combinatorial explosion when there are a large number of model input referential parameters, resulting in a slow model optimization speed. Aiming at these problems, we do the following things. Firstly, different from the existing methods that improve model performance by improving optimization algorithm, we propose a distributed belief rule base model construction and inference methodology, which reduces the algorithm complexity of model construction and inference and improves the model accuracy by dividing the model into several small and independent subsets of belief rule base. Then we prove that the distributed belief rule base model can reduce the optimization algorithm complexity by using this methodology optimizes all subsets in parallel and independently during model construction, and reduce the inference cost of inactivated rules by adopting a hierarchical inference methodology. Then we prove that the proposed distributed belief rule base model performs well on both continuous and piecewise systems; also, construction efficiency and inference accuracy using the same optimization algorithm are higher than those of the traditional belief rule base model through experiments of nonlinear continuous function fitting and binary piecewise function fitting. Finally, we improve that the proposed distributed belief rule base model has a noticeable performance advantage in complex environments compared to the traditional model through the network situation prediction experiment and point out that the distributed belief rule base model is suitable in applications that require high real-time performance and high accuracy.

INDEX TERMS
Belief rule base, distributed model, independent optimization, parallel optimization, hierarchical inference, subset of belief rule base, network security situation prediction.

I. INTRODUCTION
The belief rule base (BRB) model was proposed by Yang [1] based on the D-S evidence theory, decision theory, fuzzy theory, and IF-THEN rule base. This model effectively uses expert’s qualitative knowledge and quantitative data and offers a convenient way to model complex problems. Therefore, it has been successfully applied to many applications, including the oil pipeline leak detection [2], equipment fault diagnosis [3], and network security situation prediction [4]. However, there are still some problems to be solved in the belief rule base model, such as long construction time when the model is large and poor performance for piecewise systems.

At present, the improvement of the belief rule base model mainly relates to the improvements in model optimization method and model structure. In terms of the model optimization method improvement, Chang et al. [5] proposed a genetic algorithm-based model parameter optimization method. Yang et al. [6] proposed an improved particle swarm parameter optimization algorithm. Xu et al. [7] proposed an optimization method based on parallel multi-population and redundant gene strategy. The above methods improve model optimization efficiency to a certain extent by applying the improved optimization algorithms to the model optimization stage. In terms of the model structure improvement, Chen et al. [8] incorporated the reference value of the premise attribute into the parameters to be trained, thus increasing the number of model optimization parameters. Li et al. [9] proposed a method to determine the premise attributes of the model by means of confidence of the K-means clustering, so as to improve model applicability to complex scenes. Liu et al. [10] proposed an extended confidence rule base method to convert the training data.
into rules directly. The mentioned methods determine some structural parameters of the model by a data-driven method, which has an auxiliary effect on efficiency improvement of model construction. However, all the above-mentioned methods assume that the rule scopes in a belief rule base apply to the entire model and thus belong to global optimization methods. The global optimization methods have high algorithm complexity and belong to serial optimization methods, which do not utilize the performance of multiple (core) processors and are unable to improve the optimization efficiency further.

Mapping the scopes of a belief rule base model to a multi-dimensional space shows that the scope of each rule in the model is limited to a continuous area around the rule and the scopes of neighboring rules intersect. Since the scopes of the rules are not independent of each other, a conventional belief rule base cannot be directly divided into smaller units using the existing methods. This paper aims to divide a rule at the boundary into multiple rules using the same antecedent attributes but different other parameters. The divided rules have the same coordinates in the multi-dimensional space, their scopes are independent of each other, and the sum of their scopes represents the scope of the original rule. After the boundary rule scopes are divided, the model scope can be divided into multiple independent sub-scopes, and the subset of belief rule base (SBRB) can be constructed. All the subsets of belief rule base constitute a distributed belief rule base (DBRB). The complexity of an optimization algorithm for a belief rule base model is approximately proportional to the third power of the number of parameters to be optimized. Thus, a distributed belief rule base can effectively reduce the algorithm complexity by reducing the scales of optimization units. In addition, the construction and inference of each subset in the distributed belief rule base can be performed independently. Thus, parallel optimization can be achieved to take advantage of the performance of multiple (core) processors in the model construction and use hierarchical inference as the inference methodology to reduce invocations of irrelevant rules, thereby improving model construction and inference efficiencies.

Our contributions could be summarized as follows:

1. We propose a construction and inference method of distributed belief rule base which can reduce the complexity of model construction and inferences by dividing a belief rule base into independent subsets.
2. We analyze the time complexity of belief rule base model and point out that a distributed belief rule base model can reduce the optimization algorithm complexity compared to the traditional ones.
3. We prove that distributed belief rule base model performs well on both continuous and piecewise systems; also, construction efficiency and inference accuracy using the same optimization algorithm are higher than those of the traditional belief rule base model through the experiments of nonlinear continuous function fitting and binary piecewise function fitting.
4. We prove that proposed distributed belief rule base model performs better in complex environments compared to the traditional model and is suitable in applications that require high real-time performance and high accuracy through the network situation prediction experiment.

The paper is organized as follows. In Section II, the representations of the rules in the belief rule base, the solution method, and the optimization approach of the model are introduced. In Section III, the construction, optimization, and inference method of the distributed belief rule base is introduced, and the algorithm complexity of the model is analyzed. In Section IV, first the feasibility of the proposed model is tested by the experiment of nonlinear continuous function fitting, then the model applicability to a piecewise function is evaluated by the experiment of piecewise function fitting, and finally, the model construction efficiency and inference accuracy are further verified by a practical application of the proposed model in network security situation prediction. In Section V, The comparison of the three optimization algorithms used in the experiments is analyzed. Finally, the conclusion is given in Section VI.

II. BELIEF RULE BASE

A. REPRESENTATION OF BELIEF RULE BASE

Belief rules were proposed by Yang [1] based on the traditional IF-THEN rule. The belief rules introduce a frame-work of belief distribution and weight parameters and express the output in the form of distribution of belief degrees. A belief rule base is composed of \(1 \leq M\) belief rules, and rule \(R_k\) can be expressed as:

\[
R_k : \text{if } (x_1 A_1^k) \land (x_2 A_2^k) \land \cdots \land (x_M A_M^k) \text{ then } \{D_1, \beta_1, k), (D_2, \beta_2, k), \cdots, (D_N, \beta_N, k)\}
\]

With a rule weight \(\theta_k\) and attribute weight \(\delta_1, \delta_2, \cdots, \delta_M\)

where \(A_i^k\) \((k = 1, \cdots, L; i = 1, \cdots, M)\) denotes the referential value of the \(i\)th input of the \(k\)th rule, \(L\) denotes the number of rules, \(M\) denotes the number of antecedent attributes, \(x_i\) represents the value of the \(i\)th antecedent attribute in the input, \(D_j\) \((j = 1, \cdots, N)\) denotes the \(j\)th evaluation grade of the consequent attributes, \(N\) denotes the number of evaluation grades, \(\beta_{j, k}\) represents the belief degree of the \(j\)th evaluation grade of the \(k\)th rule, and lastly \(\theta_k\) and \(\delta_i\) denote the weights of the \(k\)th rule and the \(i\)th antecedent attribute, respectively. If \(\sum_{j=1}^{N} \beta_{j, k} = 1\), then the \(k\)th rule is complete; otherwise, it is incomplete.

B. INFERENCE OF BELIEF RULE BASE

The inference of the belief rule base is implemented using the evidential reasoning (ER) algorithms [11]–[13]. The main idea is to combine activated rules using the ER algorithms and obtain the final output of a belief rule base system. The specific inference steps of the belief rule base are as follows.

\[
\sum_{j=1}^{N} \beta_{j, k} = 1
\]
Assume the input information is expressed as \( x = (x_1, x_2, \cdots, x_M) \), then the matching degree of \( x_i \) related to the referential value \( A_i^k \) is calculated by:

\[
d_i^k = \begin{cases} 
\frac{A_i^{l+1} - x_i}{A_i^{l+1} - A_i^l} & k = l(A_i^l \leq x_i \leq A_i^{l+1}) \\
1 - d_i^k & k = l + 1 \\
0 & k = 1, 2, \cdots, N (k \neq l, l + 1).
\end{cases}
\]

The activation weight of the \( k \)th rule is calculated by:

\[
w_k = \frac{\theta_k \prod_{i=1}^M (a_i^k)^{\delta_i}}{\sum_{l=1}^L \theta_l \prod_{i=1}^M (a_i^l)^{\delta_i}}. \tag{2}
\]

If \( w_k = 0 \), the \( k \)th rule is not activated; otherwise, the rule is activated. The equation of the combination of the activated rules is expressed as (3) and (4), as shown at the bottom of the page, where \( \hat{\beta}_j \) represents the belief degree of the evaluation consequent \( D_j \). Utility can be used to transform a consequent from the form of a belief degree to a numerical value. Assume \( \mu(D_j) \) denotes the utility of the evaluation grade \( D_j \), then the expected utility of the system output \( f(x) \) is expressed as:

\[
\mu(f(x)) = \sum_{j=1}^N \mu(D_j) \hat{\beta}_j. \tag{5}
\]

C. CONSTRUCTION AND OPTIMIZATION OF BELIEF RULE BASE

The process of the belief rule base construction is divided into two parts: establishing the initial model by using qualitative knowledge and optimizing the model by using quantitative knowledge. The setting of the parameters in the initial model is according to the qualitative knowledge of experts, and there are also some data-driven generation methods that can help experts to do it [14], [15].

Insufficient or erroneous expert’s understanding of a problem can result in poor accuracy of an initial belief rule base. The initial belief rule base model based on qualitative knowledge alone can be optimized using quantitative data. The process of the belief rule base model optimization is shown in Figure 1.

Model parameters are adjusted to minimize the error between the model output and the actual system output, thereby improving the model’s inference accuracy, which is given by:

\[
\min \{ \xi(P) \} \\
\text{s.t.} \ 0 \leq \theta_k \leq 1, \quad 0 \leq \delta_i \leq 1, \quad 0 \leq \beta_{i,k} \leq 1, \quad k = 1, 2, \cdots, L \\
0 \leq \beta_{i,k} \leq 1, \quad 0 \leq \beta_{i,k} \leq 1, \quad j = 1, 2, \cdots, M \\
N \sum_{j=1}^N \beta_{i,k} = 1, \quad k = 1, 2, \cdots, L
\]

where \( P \) represents the parameter vector of the model, and \( P = (\theta_1, \cdots, \theta_L, \beta_{1,1}, \cdots, \beta_{N,L}, \delta_1, \cdots, \delta_M) \). The objective function can be expressed by the mean absolute error (MAE) as follows.

\[
\xi(P) = \frac{1}{M} \sum_{m=1}^M |y_m - \hat{y}_m| \tag{7}
\]

III. CONSTRUCTION AND INFERENCE METHOD OF DISTRIBUTED BELIEF RULE BASE

In (1), which calculates the matching degree of an antecedent attribute of the input data during the inference process of a belief rule base model, \( x_i \) can be expressed in the form of the matching degree of two neighboring referential values of the \( i \)th input attribute. Each antecedent attribute in a rule acts only in the area between the antecedent attribute and the neighboring antecedent attributes, i.e., each rule acts only on the training data distributed in the neighboring area of the rule. The scope of each rule is bounded, and there is an intersection between the scopes of the neighboring rules. Therefore, when the rule scopes at the boundaries to be divided can be processed, the belief rule base can be divided into multiple independent subsets of belief rule base that constitute a distributed belief rule base.
A. CONSTRUCTION AND OPTIMIZATION OF DISTRIBUTED BELIEF RULE BASE

1) CONSTRUCTION OF INITIAL BELIEF RULE BASE

Based on empirical knowledge, experts first determine the belief rule base parameters, including the number of antecedent attributes, referential values of antecedent attributes, weights of antecedent attributes, rule weights, evaluation grades, and belief degrees of evaluation grades, and then construct an initial belief rule base. The total number of rules contained in the initial belief rule base is \( L \) given by:

\[
L = \prod_{i=1}^{M} J_i,
\]

where \( J_i \) denotes the number of the referential values contained in the \( i \)th antecedent attribute. Considering each input attribute of the belief rule base as one dimension in the space, the effective range of a model, represented by a region in an \( M \)-dimensional space, can be denoted as the model scope. A referential value of the input rule attribute is used as a coordinate to represent the rule as a referential point in the space. Using the planes, each of which contains a referential point and is perpendicular to a coordinate axis, the model scope can be divided into \( \tilde{L} \) small complementary regions, called the same-origin domains, as given by (9). The same-origin domain is the smallest unit that the model can be divided into. The input data in the same same-origin domains activate the same rules in inference, i.e., the vertices of the same-origin domains represent \( 2^M \) rules. The maximum scope of a rule denotes an area represented by the set of the same-origin domains whose vertices are the corresponding referential points of the rule, which is called the rule scope. According to the rule scopes, belief rules can be divided into boundary rules and internal rules. The rules with boundary antecedent attributes are called the boundary rules, and the rules entirely composed of non-boundary antecedent attributes are called the internal rules. The scopes of internal rules contain \( 2^M \) same-origin domains, and the scopes of the boundary rules contain \( 1 - 2^{M-1} \) same-origin domains.

\[
\tilde{L} = \prod_{i=1}^{M} (J_i - 1)
\]

Mapping a model scope to an \( M \)-dimensional space can offer a more intuitive interpretation of model parameters. Each rule in the rule base corresponds to a referential point in the rule scope. More specifically, the \( k \)th rule is the referential point corresponding to the coordinates \( (A_k^1, \cdots, A_k^M) \), \( \mu_k(D) \) denotes the model output of the \( k \)th referential point, \( w_k \) represents the effect of the \( k \)th referential point on the surrounding regions, and lastly, \( \delta_{i,k} (i = 1, \cdots, M) \) represents the relative effect of each dimension parameters at the \( k \)th referential point.

2) DIVISION OF SCOPES OF BELIEF RULE BASE SUBSETS

1) Division according to antecedent attributes when antecedent attributes are independent of each other

According to the antecedent attribute value distribution, the antecedent attribute value can be divided into several complementary subsets. The model scope can be divided into the corresponding subsets by a group of planes, each of which contains the boundary value of a subset and perpendicular to the dimension of the boundary value; \( \tilde{J}_i \) denotes the number of subsets of the \( i \)th antecedent attribute. One subset is selected from the subsets of each antecedent attribute, and the selected subsets are combined randomly. A total of \( T \) subsets of the belief rule are obtained, as given in (10). The scope of the \( t \)th belief rule is denoted as \( R_t = \tilde{R}_1 \cap \cdots \cap \tilde{R}_M \), where \( t_M \) represents the \( M \)th antecedent attribute subset in the \( t \)th subset of the belief rule base. The scopes of subsets of the belief rule base are mutually exclusive. Also, the scopes of subsets of the belief rule base formed by this division method are all convex polyhedrons composed of mutually perpendicular planes (rectangles in a two-dimensional space, or line segments in a one-dimensional space). The vertices of subsets of the belief rule base located within the model scope represent the intersections of \( 2^M \) neighboring subsets of the belief rule base, and the scope of each subset of the belief rule base around a vertex contains only one same-origin domain belonging to the vertex.

\[
T = \prod_{i=1}^{M} \tilde{J}_i
\]

2) Division according to the model attribute when antecedent attributes are dependent on each other

When the antecedent attributes of the model are dependent on each other, it is necessary first to comprehensively consider the mutual relationship between the attributes and group the rules according to actual effects of the rules and then establish subsets of the belief rule base from the grouped rules. The scopes of the belief rule base subsets formed by this method can appear in the space as convex polyhedrons (convex polygons in a two-dimensional space, or line segments in a one-dimensional space) or concave polyhedrons (concave polygons in a two-dimensional space, or line segments in a one-dimensional space). The vertices of the belief rule base subsets located within the model scope may represent the intersection of \( 2^M \) neighboring subsets of the belief rule base. The scopes of the belief rule base subsets around a vertex contain one or more same-origin domains belonging to the vertex.

3) RULE CONSTRUCTION AT SUBSET BOUNDARIES AND DISTRIBUTED BELIEF RULE BASE GENERATION

After the scopes of the belief rule base subsets have been divided, the boundary rules can be reconstructed using the same-origin domains at the subsets’ boundaries so that a subset of the belief rule base becomes a complete belief rule base. Rules at a boundary of belief rule base subsets in the distributed belief rule base have the same antecedent attribute value; thus, their coordinates in space are the same. When neighboring subsets of the belief rule base have the
same inference value at a boundary, the boundary rules of the belief rule base can be directly generated by transforming the internal rules of a traditional belief rule base. This type of boundary is defined as a continuous boundary. When the neighboring subsets of the belief rule base have different inference values at the boundary, the boundary rules of the belief rule base subsets with the same inference values as those of the original belief rule base can be directly generated by changing the scope of the internal rules of the belief rule base. Similarly, the boundary rules of the subsets of the belief rule base with different inference values from those of the original belief rule base or undefined at a boundary shall be determined according to the inference values or limit values of inference of the belief rule base subsets at the boundary. A distributed belief rule base has more rules than the traditional belief rule base. A distributed belief rule base constructed using the method of individual division based on each antecedent attribute contains $\hat{L}$ rules, whose number is given by (11). A distributed belief rule base constructed using the overall model division method contains $(L - \hat{L})$ rules.

$$\hat{L} = L + \sum_{i=1}^{M} J_i \prod_{j \neq i} (\hat{J}_j - 1) - \prod_{i=1}^{M} (\hat{J}_i - 1)$$  \hspace{1cm} (11)

4) TRAINING DATA ASSIGNMENT
Taking the input attribute values of the training data ($x_1, \ldots, x_M$) as coordinates, and representing the training data as points in the scope of the distributed belief rule base model, the training data can be assigned to the corresponding subsets of the belief rule base according to the subset scopes where the training data points are located.

5) MODEL OPTIMIZATION
Each subset of the belief rule base is used as an independent belief rule base in model parameter optimization. The optimization methods are the same as those used for the traditional belief rule bases, and they include the FMINCON [5], genetic algorithms [6], particle swarm algorithms [7], and gradient descent methods [16]. It should be noted that a distributed belief rule base model is fully optimized when all its subsets are fully optimized.

The process of contraction and optimization of a distributed belief rule base model is shown in Figure 2.

B. INFERENCE OF DISTRIBUTED BELIEF RULE BASE
The belief rules contained in subsets of a belief rule base are different, and their scopes are independent. Thus, only the subsets of a belief rule base where the training data are located need to be invoked during the model inference, which reduces the calculation amount for inactivated rules. The inference process is as follows:

① When a distributed belief rule base model receives the input data, the input data are assigned to the corresponding subsets of a belief rule base according to the antecedent attribute values of the input data.

② The data are then processed in the subsets of the belief rule base, which includes the matching degree calculation, rule activation weight calculation, and rule combination. The processing methods are the same as those of the traditional belief rule base model. The output of the subsets of the belief rule base where the training data points are located.

③ Only one subset of the belief rule base is activated for each set of the input data, and the output of the subset is used as the output of the distributed belief rule base for the provided input data.

The flowchart of the inference process of a distributed belief rule base is presented in Figure 3.

C. TIME COMPLEXITY ANALYSIS
Assume a belief rule base contains $L$ rules, and a parameter to be optimized is represented by $P = (\theta_1, \ldots, \theta_L, \beta_{11}, \ldots, \beta_{NL}, \delta_{11}, \ldots, \delta_{ML})$; the total number of parameters to be optimized is $n$, and it is given by (12). Similarly, the total number of parameters to be optimized for the $r$th subset in a distributed belief rule base is denoted as $n_r$, and it is given by (13).

$$n = L \times (1 + N + M)$$  \hspace{1cm} (12)

$$n_r = L_r \times (1 + N + M)$$  \hspace{1cm} (13)

There are many optimization algorithms available for the optimization of the belief rule base. If the particle swarm optimization algorithm is used in the time complexity analysis, the calculations in a single iteration include the particle fitness calculation, optimal solution search, particle velocity updating, and particle position updating [17]. The swarm optimization algorithm complexity is related to the number of particles $N_p$, the spatial dimension of particles $D_p$, and the maximum number of iterations $I_p \max$, and it is expressed as $O(I_p \max N_p O(f(D_p)))$, where $O(f(D_p))$ denotes the time complexity of the evaluation function.
complexity of the function to be optimized. When $I_{p,\text{max}}$ and $N_p$ are constant, the complexity of the particle swarm algorithm is proportional to the algorithm complexity of the function to be optimized. Similarly, the complexity of the other optimization algorithms, such as genetic algorithms and simulated annealing [18], is positively correlated with the function to be optimized. In each iteration, optimization algorithms perform $N_{\text{Data}}$ operations, including the inference and error calculation, where $N_{\text{Data}}$ represents the number of the training samples. Since the inference operation contains a large number of exponentiation and multiplication operations, the complexity of an optimization algorithm is approximately $O\left(N_{\text{Data}}n_p^2\right)$, where $n_p$ represents the number of parameters to be optimized.

The complexity of an algorithm for optimizing a traditional belief rule base is approximately $O\left(N_{\text{Data}}L_p^3\left(1+N+M\right)\right)$, while the complexity of an algorithm for optimizing a distributed belief rule base is related to the operating mode of the optimization process. When the subsets of a belief rule base are optimized in parallel, the time complexity of the optimization algorithm is the lowest, and it is expressed as $O\left(N_{\text{Data}}L_p^3\left(1+N+M\right)\right)$, where $L_p$ represents the number of belief rule base subset that has the most rules. However, when the subsets of a belief rule base are optimized in series, the time complexity of the optimization algorithm is the highest, and it is equal to $O\left(\sum_{t=1}^{T}N_{\text{Data},t}L_{t,\text{max}}^3\left(1+N+M\right)\right)$. Therefore, a distributed belief rule base model reduces the optimization algorithm complexity compared to the traditional belief rule base model.

IV. EXPERIMENTAL VERIFICATION
In order to verify the algorithm proposed in this paper, the experiments of nonlinear continuous function fitting, binary piecewise function fitting, and network security situation prediction have been conducted. The experiments were performed on a workstation equipped with Inter (R) Xeon (R) Silver 4110 CPU@2.10 GHz (dual processors), 64 GB memory, and Windows 7 operating system. The proposed algorithm was implemented using MATLAB r2014b [19].

A. EXPERIMENT OF NONLINEAR CONTINUOUS FUNCTION FITTING
The belief rule base model is often used to deal with complex problems such as nonlinear systems [20]. In order to test the efficiency and accuracy of the proposed distributed belief rule base model in processing continuous functions, a commonly adopted nonlinear function was used for the testing; it is defined by (14) and plotted in Figure 4.

\begin{equation}
    f(x) = x \sin \left( x^2 \right), \quad 0 \leq x \leq 3
\end{equation}

Input $x$ was taken as an antecedent attribute of the belief rule base model. The left and right limits, as well as the three extrema of the function, were $[0,3]$ and $[1.3552,2.1945,2.8137]$, respectively, and they denoted the input attribute referential values of the model. The number of grades for the model output evaluation was set to three, corresponding to the grade utility values of $[-3,0,3]$. Accordingly, a belief rule base model framework containing five rules was constructed. The belief rule base model was divided into two belief rule base subsets with the antecedent attribute referential values of $[0,1.3552,2.1945]$ and $[2.1945,2.8137,3]$, which constitute the distributed belief rule base model framework. Each of the two subsets contained three rules. The belief degrees of the evaluation grades of the rules were initialized using the information transformation technique [14] of the rules to simulate the expert’s assignment, and the initial belief rule base was constructed. For the continuous function, the rule parameters of the initial distributed belief rule were the same as those of the initial belief rule base, so the parameters of the initial belief rule base were directly used to establish the initial distributed belief rule base model.

In order to optimize each region of the model, 70 datasets were selected according to the change rate of the function derivative in the interval of $[0,3]$ and used as training data after adding the white noise. In addition, 200 datasets were selected uniformly and used as test data to test model performance. The FMINCON optimization function was used to optimize the initial belief rule base model and initial distributed belief rule base model. The fitting of the two models to the function before and after optimization is displayed.
As shown in Figs. 5 and 6, in the continuous system, because the initial belief rule base model and the initial distributed belief rule base model had the same rule parameters and the same activated rules during each inference process, the two models provided the same inference results, but the number of rules involved in the calculation in the inference process of the distributed belief rule base model was equal to the number of rules in the subsets of the belief rule activated by the data, and thus the amount of calculation was less than that of the belief rule base. The initial (distributed) belief rule base model could roughly fit the experiment function and had high fitting accuracies near the referential values of the input attributes, but the fitting accuracy in the regions between the referential values was low. The fitted curves of the optimized distributed belief rule base model and the optimized belief rule base model were identical, with a greatly improved function fitting accuracy compared to the initial model. The fitting curve of the optimized distributed belief rule base model was not smooth enough at the boundary of the belief rule base subsets, and the fitting accuracy near the left side of the boundary was slightly lower than that of the optimized traditional belief rule base model. The fitting accuracy near the right side of the boundary was slightly higher than that of the traditional belief rule base model.

In order to evaluate the performance of the distributed belief rule base comprehensively, the particle swarm optimization algorithm and genetic algorithm were implemented respectively in the model optimization stage. The performances of the distributed belief rule base model were compared (the optimized models were denoted by PSO-DBRB and GA-DBRB) with the performance of the conventional belief rule base model (the optimized models were denoted by PSO-BRB and GA-BRB). In the particle swarm algorithm, the number of particles was set to 100, the maximum number of iterations was 1000, and the inertia weight was updated by decreasing it at each iteration [7] using the following expression:

$$w_k = \frac{w_{\text{start}}}{1 + \frac{k}{T_{\text{max}}}}$$

TABLE 1. Model performance comparison.

| Model          | Training data MAE | Testing data MAE | Construction time (s) |
|----------------|-------------------|------------------|-----------------------|
| Initial BRB    | 0.087             | 0.162            | -                     |
| FMINCON-BRB    | 0.033             | 0.037            | 196.73                |
| FMINCON-DBRB   | 0.031             | 0.036            | 16.13                 |
| PSO-BRB        | 0.173             | 0.296            | 115.48                |
| PSO-DBRB       | 0.075             | 0.105            | 18.12                 |
| GA-BRB         | 0.278             | 0.232            | 4783.01               |
| GA-DBRB        | 0.260             | 0.201            | 469.24                |

where $k$ denoted the current iteration, $w_{\text{start}}$ represented the initial inertia weight, and it was set to 0.9, and $T_{\text{max}}$ denoted the maximum number of iterations. The comparison of the model performance is given in Table 1.

The model performance comparison showed that when the same optimization function was used for optimization, the average fitting accuracies of the distributed belief rule base model to the training and test data were higher than those of the traditional belief rule base model. This was even more apparent when the particle swarm optimization algorithm was used, where the optimization accuracy was more sensitive to the number of optimization parameters. In terms of efficiency, the distributed belief rule base model was constructed much faster than the traditional belief rule base model, saving the time by about 84%–92%. In addition, the comparison of the models under different optimization conditions has shown that the FMINCON function belongs to the global optimization function, so it can effectively avoid falling into local optimum and has a faster training speed for a small number of parameters to be solved, but the function portability is poor. The particle swarm optimization algorithm has a certain probability of obtaining the optimal solution, but the optimization performance is slightly worse than that of the FMINCON function, and it is greatly affected by the number of optimization parameters. Also, it is easier to find the optimal solution when there are fewer optimization parameters. In this example, the particle swarm optimization algorithm’s optimization time is not effectively shortened, but the algorithm portability is improved. The results show that the genetic algorithm is inferior to the previous two algorithms, but is better in portability.
The experimental results presented in this section demonstrate that the distributed belief rule base model has high accuracy and construction efficiency in continuous systems. Consequently, the proposed distributed belief rule base model construction and inference method is feasible in continuous systems.

### B. EXPERIMENT OF PIECEWISE FUNCTION FITTING

The traditional belief rule base optimization algorithms generally assume that scopes of rules in a rule base are equal to the scope of the entire model, resulting in the poor fitting performance of piecewise functions. This section uses a binary piecewise function with interrelated input to test how well the distributed belief rule base model performs in the case of piecewise function. The piecewise function is given by (16) and plotted in Figure 7.

\[
z = \begin{cases} 
  x^2 + y^2 & 0 \leq x \leq 1, -1 \leq y \leq 0, \\
  -x^2 + y^2 & -1 \leq x \leq 0, -1 \leq y \leq 0, \\
  (x + y)^2 & 0 \leq y \leq 1.
\end{cases}
\]

The input of the function denoted as \((x, y)\) was used as model input antecedent attributes. According to the value ranges of the antecedent attributes and the boundary conditions of the function, the referential values of the antecedent attributes \(x\) and \(y\) were both set to \([-1, -0.5, 0, 0.5, 1]\); the number of evaluation grades of the output was set to three, and the utility values were \([-1, 1, 4]\). A belief rule base framework containing 25 belief rules was constructed. The input antecedent attributes of the belief rule base model were dependent on each other; thus, according to the relationship between the model input antecedent attributes, the belief rule base model was divided into three subsets having the antecedent attribute referential values of \([x \in [0, 0.5, 1], y \in [-1, -0.5, 0]], [x \in [-1, -0.5, 0], y \in [-1, -0.5, 0]],\) and \([x \in [-1, -0.5, 0, 0.5, 1], y \in [0, 0.5, 1]],\) thereby constructing a distributed build belief rule base model framework. The subsets contained 15, 9, and 9 belief rules. Their scopes are shown in Figure 8.

The scope boundaries of SBRB3 and SBRB1, and of SBRB1 and SBRB2 were both continuous boundaries, so the inference values of the two subsets of the belief rule base at a boundary were the same. The boundary rules of the belief rule base subsets were formed by directly transforming the internal rule at the boundary inside the belief rule base. The scope boundary of SBRB3 and SBRB2 was a discontinuous boundary; SBRB3 was defined at the boundary, but SBRB2 was not. In this case, the boundary rules of SBRB3 could be directly formed by transforming the internal rules of the belief rule base, while the boundary rules of SBRB2 could be determined based on the limit inference values of SBRB2 at the boundary.

In region \([x \in [-1, 1], y \in [-1, 1]],\) according to the changing rate of the partial derivative of the function, 441 datasets with added white noise were used as the training data, and 1764 datasets were uniformly selected as the test data and used to test model performance. By using an information transformation technique of the rules to simulate expert’s assignment, the results and belief degrees of the rules were initialized, and the initial belief rule base model and the initial distributed belief rule base model were established. These two models were optimized using the FMINCON function. The comparison of the fitting of the two initial models to the function is shown in Figure 9, while the comparison of the fitting performance of the optimized models to the function is shown in Figure 10.

As shown in Figs. 9 and 10 the fitted surface of the initial belief rule base model was smooth in all regions, and the fitting was poor at the boundaries of function segments; in the continuous segments of the function, the fitting of the area near the rules was good, and the fitting of the area between the rules was poor. The fitted surfaces of the initial distributed belief rule model were smoother inside the subsets.
TABLE 2. Model performance comparison.

| Model          | Training data MAE | Testing data MAE | Construction time (s) |
|----------------|-------------------|------------------|----------------------|
| Initial BRB    | 0.1321            | 0.1435           | -                    |
| Initial DRBR   | 0.0942            | 0.0962           | -                    |
| FMINCON-BRB    | 0.0284            | 0.0285           | 4899.88              |
| FMINCON-DRBR   | 0.0109            | 0.0125           | 520.58               |
| PSO-BRB        | 0.2425            | 0.2563           | 749.24               |
| PSO-DRBR       | 0.0379            | 0.0459           | 190.31               |
| GA-BRB         | 0.1593            | 0.1675           | 19245.26             |
| GA-DRBR        | 0.0711            | 0.0735           | 2079.31              |

Consequently, the distributed belief rule base model is efficient in processing piecewise functions.

C. EXPERIMENT OF NETWORK SECURITY SITUATION PREDICTION

In order to test the performance of the proposed algorithm in an actual application scenario, a network security situation prediction experiment was conducted to test the performance of the model with an office network as an experimental subject. The structure of the experimental network is displayed in Figure 11.

Changes in the network security follow a certain pattern, which can reflect attacker’s attempts to a certain extent, which is why time series methods have been generally used to predict the network situation [21], [22]. In the experiment, the situation values of the first \( n \) time intervals were used as the input, and the changing trend of the situation values was determined to predict the network security situation in the next time interval. The number of input parameters in the network security situation prediction model is closely related to prediction model attributes, network attributes, and situation prediction accuracy requirements. In the situation prediction models based on the Markov model (MM) or hidden Markov model (HMM) [23], [24], the number of input parameters is usually set to one. However, in the situation prediction models based on artificial neural networks, such as support vector machine, and Bayesian network [25]–[28], the number of input parameters is usually set to 3–5. In order to balance the accuracy and rapidity of prediction, the number of model input parameters was set to four in this work.

1) DATA ACQUISITION

As the third stage of the network security situation awareness, the network security situation prediction should use network security situation assessment to generate the required network security situational values [29]. The vulnerability of network equipment includes the https remote buffer overflow vulnerability (opensslRBOF), local buffer overflow vulnerability (mysqlLBOF), remote denial of service vulnerability (chargenDoS), and FTP buffer overflow vulnerability.
An attacker can use the Internet to attack an office network. The network security assessment equipment is responsible for assessing network situations, and it obtains the values of the network security situation related parameters according to certain indicators, such as the number of attacks, attack type, and attack severity level. The assessment time interval was one day. The security situations of the experimental network running continuously for 104 days were recorded and used to construct the network security situation prediction time series, as shown in Figure 12.

A sliding window method [33] with a window size of five, where each time sliding backward for one time unit, was used to generate 100 sets of data samples. The first 90 sets of samples were used as training samples, and the remaining ten sets of samples were used as test samples. The sample input consisted of the network security situation values in the four previous time intervals, namely, \( x(t-3), x(t-2), x(t-1), \) and \( x(t) \). The output denoted the predicted security situation values of the network in the next time interval, i.e., \( x(t+1) \). The input and output of some samples are given in Table 3.

### TABLE 3. Input and output for some samples.

| No. | Input data       | Output data |
|-----|------------------|-------------|
|     | \( x(t-3) \)     | \( x(t-2) \) | \( x(t-1) \) | \( x(t) \) | \( x(t+1) \) |
| 1   | 0.44             | 0.60        | 0.58         | 0.72       | 0.46         |
| 25  | 0.56             | 0.40        | 0.57         | 0.57       | 0.56         |
| 50  | 0.58             | 0.49        | 0.46         | 0.54       | 0.53         |
| 75  | 0.29             | 0.49        | 0.47         | 0.42       | 0.40         |
| 100 | 0.50             | 0.51        | 0.38         | 0.52       | 0.34         |

The network situation sequence is a continuous function; thus, the inference outcomes of the two initial models are the same. The plots in Figs. 14 and 15 show that the initial model constructed using only qualitative knowledge was inaccurate in the fitting on training data and in predicting the test data. After optimization, the inference accuracies of the two models have been effectively improved, and the inference accuracies of the distributed belief rule base model on the training and test data were higher than those of the traditional belief rule base model.

In order to test the performance of the models for different optimization methods, the variable-weight particle swarm optimization algorithm and genetic algorithm were used in the model optimization stage to generate the optimized models. The performances of the two models before and after the optimization are shown in Table 4.

The performance comparison of the two models shows that for the same optimization function, compared to the traditional belief rule base, the distributed belief rule base improved the average prediction accuracy by 71%–95% and reduced the construction time by 85%–99%.

Since the network security situation sequence is acquired by the existing network security tools, the measurement errors are inevitable. Due to uncertainties of network attack targets and subjective uncertainties of behaviors of the attack based on experts’ knowledge of the network and used to establish the initial belief rule base model. The reference value of each input attribute in the initial belief rule base model was \([0.0,0.44,0.5,0.58,1]\). Supposing that the input attributes in the model were independent, the reference values were evenly divided into two subsets: \([0.0,0.44,0.5]\) and \([0.5,0.58,0.1]\), constituting the initial distributed belief rule base that contained 16 subsets. The FMINCON function was used to optimize the two models. The fitting performances of the two models to the training data before and after the optimization are shown in Figure 13. The prediction accuracy of the model for the test data is shown in Figure 14.

This section assumes that the network situation sequence is a continuous function; thus, the inference outcomes of the two initial models are the same. The plots in Figs. 14 and 15 show that the initial model constructed using only qualitative knowledge was inaccurate in the fitting on training data and in predicting the test data. After optimization, the inference accuracies of the two models have been effectively improved, and the inference accuracies of the distributed belief rule base model on the training and test data were higher than those of the traditional belief rule base model.

In order to test the performance of the models for different optimization methods, the variable-weight particle swarm optimization algorithm and genetic algorithm were used in the model optimization stage to generate the optimized models. The performances of the two models before and after the optimization are shown in Table 4.
targets, the network situation prediction error cannot be eliminated, and thus the model prediction accuracy on test data is generally lower than the fitting accuracy on training data. However, network situation changes show certain statistics trends, which presents the overall predictability. Thus, by combining qualitative knowledge of experts and quantitative data collected from the network, good prediction results can be achieved, which provides network management personnel with a helpful reference in maintaining and upgrading network security.

V. DISCUSSION

In order to comprehensively evaluate the performance of the distributed belief rule base in different optimization functions, the FMINCON function, the particle swarm optimization algorithm and genetic algorithm were implemented respectively in the model optimization stage of the three experiments in Section IV. The following conclusion can be drawn based on the comparison of the three optimization algorithms.

1. The performance of the FMIINCON function is very sensitive to the number of training parameters; namely, when the number of training parameters is large, the optimization speed and accuracy of this optimization method are significantly reduced.

2. The optimization efficiency of the particle swarm algorithm increased with the number of parameters, but the increase in the particle dimension makes it more difficult for the models to find optimal solutions.

3. The genetic algorithm is less sensitive to the number of optimization parameters than the other two optimization algorithms, but the optimization performance is relatively poor.

VI. CONCLUSION

A distributed belief rule base model represents an improvement of a traditional belief rule base model. This model can be constructed by using comprehensively qualitative knowledge and quantitative data. It reduces the complexity of model construction and inferences by dividing a belief rule base into independent subsets. It can also deal with the problems in continuous and piecewise systems. It has high construction efficiency and inference accuracy and is suitable for applications that require high real-time and accuracy performances.

Compared with the traditional model, the construction of distributed belief rule base model has more steps such as the construction of belief rule base subsets and the allocation of training data, which puts forward higher requirements for the programming. In addition, the results of the distributed belief rule base model may be discontinuous at the segmentation, so the current boundary rule generation method could be further improved. The next step is to explore the construction and optimization methods of belief rule base subsets further and expand the application scope of the distributed belief rule base model.

REFERENCES

[1] J.-B. Yang, “Rule and utility based evidential reasoning approach for multattribute decision analysis under uncertainties,” Eur. J. Oper. Res., vol. 131, no. 1, pp. 21–61, May 2001.

[2] Y. You, J. Sun, J. Jiang, and S. Lu, “A new modeling and inference approach for the belief rule base with attribute reliability,” Int. J. Speech Technol., vol. 50, no. 3, pp. 976–992, Mar. 2020.

[3] Z. Zhou, Z. Feng, C. Hu, G. Hu, W. He, and X. Han, “Aeronautical relay health state assessment model based on belief rule base with attribute reliability,” Knowl.-Based Syst., vol. 197, Jun. 2020, Art. no. 105869.

[4] G.-Y. Hu and P.-L. Qiao, “Cloud belief rule base model for network security situation prediction,” IEEE Commun. Lett., vol. 20, no. 5, pp. 914–917, May 2016.

[5] R. Chang, Y. S. Bai, and Q. Meng, “Study on optimization of system learning of belief rule base,” J. North China Univ. water Resour. Hydropower, Natural Sci. Ed., vol. 36, no. 4, pp. 72–78, Aug. 2015.

[6] H. Yang, P. Z. Wu, and J. L. Ni, “Parameter Training algorithm based on Improved Particle Swarm belief rule base,” Comput. Eng. Des., vol. 38, no. 2, pp. 400–404, Feb. 2017.

[7] X. B. Xu, W. Zhu, X. J. Xu, “Belief rule base optimization method based on parallel multi-population and redundant gene strategy,” Acta Automata Sinica, to be published, doi: 10.16383/j.aas.c190580.

[8] Y.-W. Chen, J.-B. Yang, D.-L. Xu, Z.-J. Zhou, and D.-W. Tang, “Inference analysis and adaptive training for belief rule based systems,” Expert Syst. Appl., vol. 38, no. 10, pp. 12845–12860, Sep. 2011.

[9] B. Li, H. W. Wang, and B. Yang, “Confidence K-mean clustering algorithm for structure recognition of belief rule base,” Syng. Eng., vol. 29, no. 5, pp. 85–91, May 2011.

[10] J. Liu, L. Martinez, A. Calzada, and H. Wang, “A novel belief rule base representation, generation and its inference methodology,” Knowl.-Based Syst., vol. 53, pp. 129–141, Nov. 2013.

[11] J. B. Yang, J. Liu, J. Wang, H.-S. Sii, and H.-W. Wang, “Belief-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, Feb. 2016.

[12] 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.

[13] Y. Song, J. Zhu, L. Lei, and X. Wang, “Self-adaptive combination method for temporal evidence based on negotiation strategy,” Sci. China Inf. Sci., vol. 63, no. 11, Nov. 2020, Art. no. 210204, doi: 10.1007/s11432-020-3045-5.

[14] N. N. Chen, X. T. Gong, and Y. Y. Fu, “Extended belief rule-based reasoning method based on an im-proofed rule activation rate,” CAAI Trans. Intell. Syst., vol. 14, no. 6, pp. 1179–1188, Nov. 2019.

[15] B. Zhang, Y. Zhang, G. Hu, Z. Zhou, L. Wu, and S. Lv, “A method of automatically generating initial parameters for large-scale belief rule base,” Knowl.-Based Syst., vol. 199, Jul. 2020, Art. no. 105904.

[16] M. T. Orváño-Barreras, C. N. Sánchez, M. Rivera, M. Acevedo, and R. Velázquez, “Gradient descent-based optimization method of a four-bar mechanism using fully Cartesian coordinates,” Appl. Sci., vol. 9, no. 19, Oct. 2019, Art. no. 4115.

[17] A. Faza and A. Al-Mousa, “PSO-based optimization toward intelligent dynamic pricing schemes parameterization,” Sustain. Cities Soc., vol. 51, Nov. 2019, Art. no. 101776.
Q. Hu et al.: Novel Construction and Inference Methodology of BRB

QINGSHUANG HU was born in Shandong, China, in 1996. He received the B.S. degree from Air Force Engineering University, in 2018, where he is currently pursuing the M.S. degree. His research interests include evidence theory, fuzzy sets, and network security situation prediction.

CHENGLAI LI was born in Shandong, China, in 1966. He received the Ph.D. degree from Air Force Engineering University of China. He is currently a Professor with the Air Force Engineering University of China. His research interests include evidence theory, embedded systems, and network security.

YANLI LU was born in Xi’an, China, in 1979. She received the Ph.D. degree from Air Force Engineering University (AFEU), in 2009. She is currently a Lecturer with the College of Air and Missile Defense, AFEU. Her research interests include intuitionistic fuzzy sets and evidential reasoning.

SONG LI was born in Anhui, China, in 1977. He received the Ph.D. degree from Air Force Engineering University (AFEU). He is currently an Associate Professor with the College of Air and Missile Defense, AFEU. His main research interests are command automation information processing and air defense auxiliary decision.