

A Conflict Detection Method Based on Constraint Satisfaction in Collaborative Design

Kangkang Yang¹, Shijing Wu^{1*}, Wenqiang Zhao^{1,2}, and Lu Zhou¹

¹School of Power and Mechanical Engineering, Wuhan University, Wuhan, and

²Henan Pinggao Electric CO. Ltd., Pingdingshan, China

yangkk0503@hotmail.com, wsj@whu.edu.cn, 349402566@qq.com, 172877508@qq.com

Abstract

Hierarchical constraints and constraint satisfaction were analyzed in order to solve the problem of conflict detection in collaborative design. The constraints were divided into two sets: one set consisted of known constraints and the other of unknown constraints. The constraints of the two sets were detected with corresponding methods. The set of the known constraints was detected using an interval propagation algorithm, a back propagation (BP) neural network was proposed to detect the set with the unknown constraints. An immune algorithm (IA) was utilized to optimize the weights and the thresholds of the BP neural network, and the steps were designed for the optimization process. The results of the simulation indicated that the BP neural network that was optimized by IA has a better performance in terms of convergent speed and global searching ability than a genetic algorithm. The constraints were described using the eXtensible Markup Language (XML) for computers to be able to automatically recognize and establish the constraint network. The implementation of the conflict detection system was designed based on constraint satisfaction. A wind planetary gear train is taken as an example of collaborative design with a conflict detection system.

Category: Smart and intelligent computing

Keywords: Collaborative design; Conflict detection; Constraint satisfaction; Immune algorithm; Back propagation neural network

I. INTRODUCTION

Collaborative design is an important branch of computer supported cooperative work (CSCW) that consists of a group-working style that implements effective communication and cooperation to design complex products. Designers from different disciplines participate in the collaborative design process, and the variables of the designers are interrelated, interdependent and exert mutual restraint. Conflicts inevitably occur as a result of the differences in background knowledge, different views for an

issue and different standards of evaluation. Conflict detection is an important function of collaborative design since conflicts can reduce the efficiency of the design process.

Extensive research has been conducted to effectively detect conflicts in collaborative design. Yvars [1] considered that the design process can be modeled in the form of a constraint satisfaction problem (CSP). Meng et al. [2] described the CSP in a formal expression and explored the conflict detection problem. Hu et al. [3] investigated a method for conflict detection that is based on a vertical constraint network model. Slimani et al. [4] proposed

Open Access <http://dx.doi.org/10.5626/JCSE.2015.9.2.98>

<http://jcse.kiise.org>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 11 Mar 2015; Revised 27 May 2015; Accepted 28 May 2015

*Corresponding Author

eliminating conflicts by sharing and exchanging knowledge during the collaborative design process. Zhu and Song [5] put forward a conflict detection algorithm based on the design history in collaborative CAD design. Zhao et al. [6] and Xie et al. [7] used constraint verification and interval propagation algorithms to detect conflicts. Wang and Jin [8] proposed a consistency model for the operation sequence and conflict detection algorithm based on the geometry level. Xiong et al. [9] presented an approach for distributed conflict detection to support concurrent design.

The existing methods for conflict detection mainly use the interval propagation algorithm to solve the constraint network and to detect conflicts. However, the existence of massive implicit conflicts causes the exact ranges of some constraints to be difficult to determine. Conflicts cannot be detected comprehensively and accurately. Therefore, this paper utilizes hierarchical constraints and constraint satisfaction to construct a conflict detection method that provides a theoretical basis for conflict digestion.

II. CONSTRAINT ANALYSIS OF COLLABORATIVE DESIGN

There are many types of constraints in collaborative design, including design constraints, process constraints, manufacturing constraints, etc., and these are associated with product attributes that form a network and constitute the boundaries of the possible design solutions. Each design problem can be converted to a solution process based on the constraint network, and conflicts will occur when the constraints cannot be satisfied.

A. Analysis of Hierarchical Constraints

During the collaborative design process, design goals are mapped onto the product object tree by using a hierarchical structure. From top to the bottom, the tree consists

of the product, components, parts, and features. Fig. 1 shows the transmission diagram of a compound planetary gear train and the relationships between the object tree and the hierarchical constraints.

The constraints reflect the restrictive relationships between the product and the design purposes. When the collaborative design of a wind planetary gear train is taken as an example, the constraint network can be divided into layers consisting of the product, components, parts, and features. The product layer describes the product performance, weight and structure, such as the train power, transmission ratio, etc. The component layer describes the design requirements for the component and the constraints among the different parts, such as the design of a sub-gear train, a cabinet, etc. The part layer describes the design requirements of the parts, such as the gears, shafts, etc. The feature layer describes the design parameters of the parts, such as the geometry, dimensions, strength requirements, etc.

The interaction between the different levels of constraints can be reduced since a controllable network of hierarchical constraints can be set up. It then is more convenient to manage the relationships of the constraints when design attributes change. As shown in Fig. 1, if the number of teeth of the sun gear changes, the designers only have to modify the corresponding layer of that part. The constraints of the different layers are related, and when conflicts occur for a high-level constraint, the conflict in the lower levels can quickly be found. Conflicts can be detected by verifying the constraints from layer to layer, starting from the feature layer.

B. Analysis of Constraint Satisfaction

The solutions of the constraint network can be expressed by CSP, according to the following equation [10]:

$$CSP = (X, D, C) \tag{1}$$

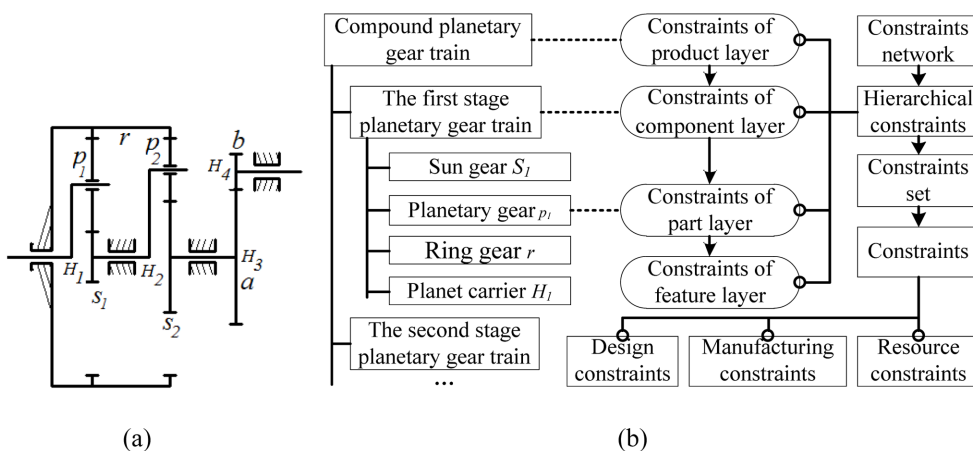


Fig. 1. Transmission diagram of compound planetary gear train (a) and its constraints (b).

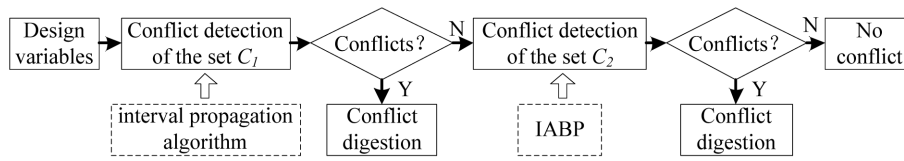


Fig. 2. Process of conflict detection. IABP: back propagation neural network optimized with immune algorithm.

where $X=\{x_1, x_2, \dots, x_n\}$ is the set of design variables, $D=\{d_1, d_2, \dots, d_n\}$ is the range of design variables, $C=\{c_1, c_2, \dots, c_n\}$ is the set of constraints. n is the number of constraints, c_i consists of two parts: $V(c_i)=\{c_{i1}, c_{i2}, \dots, c_{im}\}$ and $R(c_i)$, $R(c_i)$ is a subset of $d_{i1} \times d_{i2} \times \dots \times d_{im}$.

The solutions of the constraints can be expressed as:

$$K(X) = \{(X_1=x_1, X_2=x_2, \dots, X_n=x_n) | \forall c_i, \prod V(c_i)R(c) \subseteq R(c_i)\} \tag{2}$$

where $\prod V(c_i)R(c) \subseteq R(c_i)$ is the projection of the variable set $V(c_i)$ in the constraint set $R(c_i)$. If Eq. (1) has a solution, then there is no conflict. If there is no solution, then conflicts have occurred.

Collaborative design is a continuous process to discover and digest conflicts. So it is impossible to confirm that all possible constraints of the design variables have been met until the design is completed. As the design is further developed, the constraints that have been determined may constantly change, and new constraints will appear. However, some constraints cannot be specified, such as resource allocation, data sharing and data cooperating between different design departments. Therefore, the set of constraints C can be divided into one set C_1 with known constraints and another set C_2 with unknown constraints. Unknown constraints may be converted into known constraints during design, and the set of constraints can be written as:

$$\begin{cases} C = C_1 \cup C_2 \\ C_1 = f_1(X, D) \\ C_2 = f_2(X, D) \end{cases} \tag{3}$$

The solution for the conflicts in the collaborative design process can be transformed to a solution for Eq. (3) to make sure whether or not conflicts have happened.

III. DESIGN OF THE CONFLICT DETECTION MODEL

The set of constraints C_1 can be directly verified by using the interval propagation algorithm [11]. The implicit conflicts make it difficult to confirm some of the constraints in set C_2 , which cannot be solved using the interval propagation algorithm. The relationship between the constraints and conflicts is highly nonlinear, and the

influence of the constraints that has an impact on the conflict is different. A back propagation (BP) neural network can be used to approach the complex nonlinear system with arbitrary precision. This method has been successfully applied in multivariate nonlinear problems, such as fault diagnosis and life prediction [12, 13]. However, the BP neural network has disadvantages in that it exhibits a slow convergence rate, local extreme point and weak generative ability. The biological immune mechanism is used as a basis for the immune algorithm (IA), which is an improved genetic algorithm that combines immune theory with genetic algorithm development [14]. In order to retain the ability for a global random search, this method implements mechanisms that exist in biological immune systems, such as antigen recognition, antibody diversity, immune memory, antibody encouragement and restraint, antibody diversity keeping, etc. It avoids premature termination and guarantees that the result converges to the global minimum.

This paper uses an interval propagation algorithm to detect a set of constraints C_1 , while the set of constraints C_2 is simulated by using a BP neural network. IA is used to optimize the BP neural network's weights and thresholds (IABP). IABP not only improves the ability of generalization during mapping, but it also ensures the algorithm can rapidly convergence with a globally optimal solution and a strong learning ability. Finally, this algorithm can accurately detect whether or not there is a conflict. Fig. 2 shows the detection process for the collaborative design.

A. Conflict Detection Based on Interval Propagation Algorithm

1) Description of the Algorithm

According to the interval propagation algorithm, the answer for design variable x_i can be obtained by solving the set of constraints C as:

$$d'_i = f^{-1}[R(c_i), x_i] \tag{4}$$

So the new solution for the interval of x_i can be written as:

$$d''_i = d'_i \cap d_i \tag{5}$$

If Eq. (5) is not empty, d_i will be replaced by d''_i , and the feasible solution interval for all design variables x_i

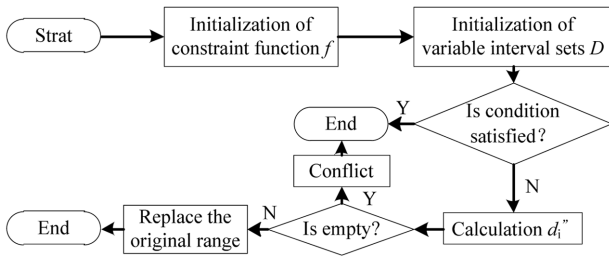


Fig. 3. Detection process based on the interval propagation algorithm.

can be written as:

$$D' = [d_1^n, d_2^n, \dots, d_n^n] \quad (6)$$

If d_i^n is empty, the constraint network is an unsolved problem, and so there must be a conflict in the collaborative design process. Fig. 3 shows the detection process based on the interval propagation algorithm.

2) Validation of Interval Propagation Algorithm

As is shown in Fig. 1, the first stage planetary gear train in the wind turbine gearbox consists of one sun gear, one ring gear, one planet carrier and three planetary gears. The interval propagation algorithm will be used to detect whether a conflict occurs when the adjacent planetary gear trains are installed. Fig. 4 shows the constraint network of the adjacent planetary gear trains without a conflict.

Z_1 is the teeth number of the sun gear, Z_2 is the teeth number of the planetary gear, Z_3 is the teeth number of the ring gear, and i is the transmission ratio. The detection process is given as follows.

a) Initialization of the constraint function:

$$f = \left\{ Z_1 \geq \frac{2\sqrt{3}-3}{3} Z_2 + \frac{4\sqrt{3}}{3}, Z_2 \leq \frac{\sqrt{3}Z_1-4}{2-\sqrt{3}}, Z_3 = Z_1 + 2 \cdot Z_2, \dots \right\}$$

b) Initialization of variable interval sets:

$$D = \{d_{z_1}, d_{z_2}, d_{z_3}, d_i\} = \{d_{z_1} = [17, 25], d_{z_2} = [20, 40], \dots, d_i = [6, 8]\}$$

c) Calculation process:

$$Z_1 \geq \frac{2\sqrt{3}-3}{3} Z_2 + \frac{4\sqrt{3}}{3} \Rightarrow d_{z_1}' = [8.5, +\infty] \Rightarrow d_{z_1}'' = [17, 25]$$

$$Z_2 \leq \frac{\sqrt{3}Z_1-4}{2-\sqrt{3}} \Rightarrow d_{z_2}' = [0, 94.96] \Rightarrow d_{z_2}'' = [20, 40]$$

...

$$i = 1 + \frac{Z_3}{Z_1} \Rightarrow d_i' = [3, 5.7] \Rightarrow d_i'' = \emptyset$$

The results indicate that the transmission ratio is unable to meet the constraint requirements. By analyzing the reasons for the conflict during the design process, the

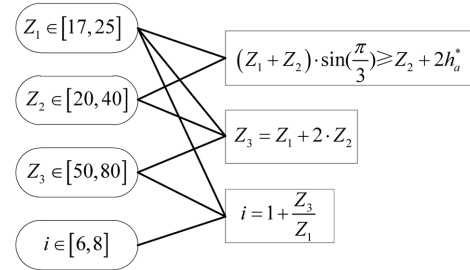


Fig. 4. Constraint network of adjacent planetary gear trains without a conflict.

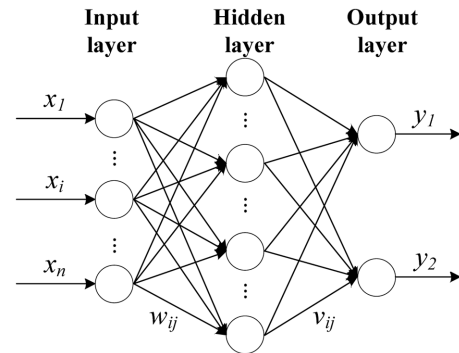


Fig. 5. Back propagation neural network structure.

number of teeth of the ring gear is found to not match the transmission ratio. In order to eliminate such conflicts, the designers should change the relationship between the number of teeth of the ring gear and the transmission ratio.

B. Conflict Detection Algorithm Based on IABP

1) Model of the BP Neural Network

Hsu et al. [15] found that a three-layer BP neural network can solve the random function fitting and approximation problem. As a result, a three-layer BP neural network was adopted in this paper. As shown in Fig. 5, the BP neural network has three layers: an input layer, a hidden layer and an output layer.

The input variables are composed by design variables X ($X = \{x_1, x_2, \dots, x_n\}$). The results of the conflict detection can be divided into two kinds: one with conflicts and another without conflicts. The expression of (0, 1) represents the result without conflicts, while (1, 0) means that conflicts have occurred. Therefore, the number of hidden layer neurons nodes can be given as [16]:

$$n_2 = \sqrt{n_1 + n_3} + \alpha \quad (7)$$

where n_1 is the number of input layer neurons nodes; n_3 is the number of output layer neurons nodes; α is a constant

that is randomly produced in the range [0, 1].

The dimensions of the variables and the goals are different during design. If the parameters are used to directly detect conflicts, the error in the precision of the BP neural network will decrease. Before the data is fed into the BP neural network, the data must be normalized within [0.1, 0.9], according to the following equation:

$$x'_i = 0.1 + \frac{x_i - x_{min}}{x_{max} - x_{min}} * (0.9 - 0.1) \quad (8)$$

where x_i represents the input variables for all i ; x_{min} is the minimum of the data, x_{max} is the maximum; and x'_i represents the input variables after normalization.

2) Learning Algorithm of BP Neural Network

The mean square error of the actual and the target output is taken as the training error function of the BP neural network, which is defined as:

$$E = \frac{1}{2mq} \sum_{j=1}^q \sum_{i=1}^m (y_{ij} - z_{ij})^2 \quad (9)$$

where m is the total number of samples, q is the number of output layer neurons nodes, y_{ij} is the actual output, and z_{ij} is the target output.

To improve the convergence speed and the generalization, an additional momentum method and adaptive training algorithm are used to train the BP neural network [17]. The weight matrix can be amended as follows:

$$w^{t+1} = w^t - \eta \left. \frac{\partial E}{\partial w} \right|_{w=w^t} + a\eta \left. \frac{\partial E}{\partial w} \right|_{w=w^{t+1}} \quad (10)$$

where a is a dynamic factor that is randomly produced among [0, 1], η is learning rate, t is the iterative steps, w^t is the weight matrix, and w^{t+1} is the weight matrix of the next generation.

3) Process of BP Neural Network Optimized by Immune Algorithm

To obtain better initial weights and thresholds, IA is used to optimize the weights and the thresholds of the BP neural network. Fig. 6 shows a flow chart of the BP neural network that has been optimized with IA.

a) Antibody encoding: Binary encoding is for the connection weights between the input layer and a hidden layer, the thresholds of the hidden layer, connecting weights between the hidden layer and the output layer and the thresholds of the output layer. Table 1 shows the individual code of the BP neural network. The number of

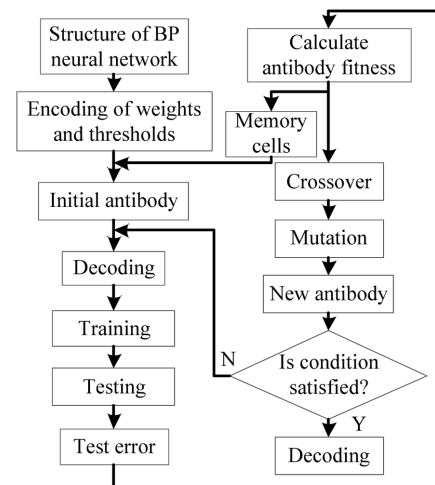


Fig. 6. Flowchart of back propagation (BP) network optimized by immune algorithm.

antibodies consists of n new antibodies and m memory cells. Therefore, the total number of initial antibodies is p , the sum of n and m .

b) Fitness function: To reduce the error, the fitness function can be defined as:

$$f(x_i) = \frac{1}{E(x_i) + \xi} \quad (11)$$

where $E(x_i)$ is the mean square error of the antigen and ξ is the penalty factor.

c) Concentration of the antibody: To ensure a diversity of antibodies, the concentration of the antibody is calculated using a method based on the Euclidean distance. The concentration of the antibody is calculated by:

$$V(x_i) = \frac{1}{e^{D_{x_i}}} \quad (12)$$

where D_{x_i} is the sum of all Euclidean distances of antibodies. The higher the concentration of the antibody, the more similarity there is between antibodies.

d) Expected reproduction probability: The expected reproduction probability is determined by the fitness $f(x_i)$ and the concentration $V(x_i)$, which can be written by:

$$p(x_i) = \lambda \frac{f(x_i)}{\sum f(x_i)} + (1 - \lambda) \frac{V(x_i)}{\sum V(x_i)} \quad (13)$$

where λ is the reproduction probability. From the above

Table 1. Antibody encoding

Connecting weights between the input layer and hidden layer	Thresholds of the hidden layer	Connecting weights between the hidden layer and the output layer	Thresholds of the output layer
$n_1 \times n_2$	n_2	$n_2 \times n_3$	n_3

Table 2. Data samples

No.	S_1	p_1	S_2	p_2	r	a	b	m_1	m_2	Conflict
1	0.1000	0.5800	0.1000	0.5800	0.5000	0.3811	0.9000	0.9000	0.1000	Yes
2	0.1000	0.9000	0.1000	0.9000	0.7667	0.1000	0.7000	0.4429	0.5000	Yes
3	0.1000	0.9000	0.7857	0.7400	0.7667	0.2297	0.5000	0.4429	0.5000	Yes
4	0.1000	0.7400	0.1000	0.7400	0.6333	0.2730	0.9000	0.4429	0.5000	No
5	0.1000	0.2600	0.1000	0.2600	0.2333	0.7054	0.9000	0.9000	0.1000	Yes
6	0.1000	0.1000	0.1000	0.1000	0.1000	0.9000	0.9000	0.9000	0.1000	Yes
7	0.6333	0.6867	0.5571	0.6867	0.6778	0.6189	0.9000	0.4429	0.1000	No
8	0.6333	0.5267	0.5571	0.5267	0.5444	0.5973	0.5000	0.4429	0.5000	Yes
9	0.9000	0.7400	0.7857	0.7400	0.7667	0.5541	0.5000	0.1000	0.5000	Yes
10	0.9000	0.5800	0.7857	0.5800	0.6333	0.6838	0.5000	0.1000	0.5000	Yes
11	0.1000	0.7400	0.7857	0.5800	0.6333	0.4892	0.9000	0.4429	0.1000	No
12	0.2333	0.6867	0.9000	0.5267	0.6111	0.4676	0.5000	0.4429	0.5000	No
13	0.6333	0.3667	0.5571	0.3667	0.4111	0.5541	0.1000	0.4429	0.9000	Yes
14	0.9000	0.9000	0.7857	0.9000	0.9000	0.5757	0.9000	0.1000	0.1000	Yes

formula, we can conclude that when the concentration of the antibody is high, it is difficult to select the antibody with a high fitness; when the concentration of the antibody is low, it is easy to select the antibody with a high fitness. In this manner, the good individual is withheld, the choice of a similar antibody is reduced, and the variation is ensure in the individual.

e) Genetic manipulation: The basic purpose of an immune system is to recognize foreign cells and molecules. If the difference between the antibody and the antigen is smaller, the affinity between the antibody and the antigen is higher and recognition is therefore also more likely.

- Selection: a uniform stochastic selection method is used [18].
- Crossover: a one-point crossing method is used.
- Mutation: a uniform mutation is used for experimentation.

4) Validation of IABP

The designers have designed 14 schemes for a compound planetary gear train, as shown in Fig. 1, and the set of constraints C_1 have been verified. The number of teeth and the modulus of the gears are fed into the BP neural network as input variables. Table 2 shows the normalized data according to Eq. (9), in which Nos. 1–10 are training samples and Nos. 11–14 are testing samples. S_1 and p_1 are the number of teeth of the sun gear and the planetary gear in the first stage planetary gear train, respectively; S_2 and p_2 are the number of teeth of the sun gear and the planetary gear in the second stage planetary gear train, respectively; r is the number of teeth of the ring gear; a and b are the numbers of teeth of the gear pair; m_1 and m_2 are

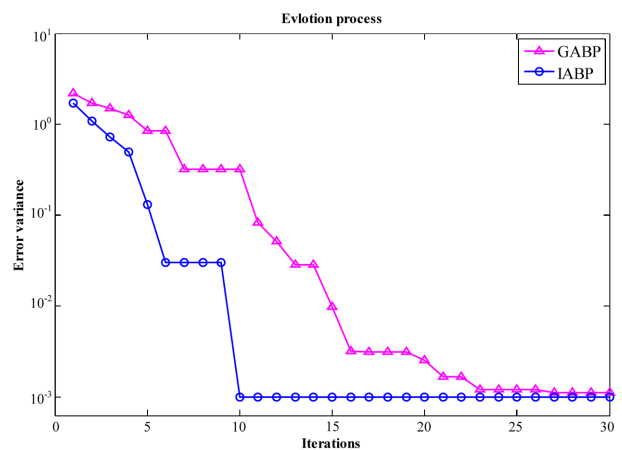


Fig. 7. Evolutionary curve of error. GABP: back propagation neural network optimized with genetic algorithm, IABP: back propagation neural network optimized with immune algorithm.

the modulus of the planetary gear train and the parallel shaft, respectively.

According to Eq. (7), the BP neural network uses a 9-12-2 structure with three layers. In the emulation, the max training time is of 1000, the training goal is 0.01, and the learning rate is 0.1.

In order to verify the superiority of IABP, both the IA and the genetic algorithm (GA) were used to optimize the BP neural network. The population size is 40, the maximum hereditary algebra is 30, the crossover rate is 0.7, the mutation rate is 0.01, the propagation rate is 0.95, the capacity for the memory base is 10, the generation gap is 0.95, and ξ is 0.001.

After training and testing, the error curve of the evolu-

Table 3. Test results

BP		GABP		IABP	
O_1	O_2	O_1	O_2	O_1	O_2
0.0158	0.9113	0.0071	0.9298	0.0053	0.9354
0.0163	0.9217	0.0063	0.9321	0.0067	0.9375
0.9138	0.0139	0.9368	0.0066	0.9349	0.0079
0.9196	0.0172	0.9287	0.0075	0.9311	0.0072

BP: back propagation, GABP: BP neural network optimized with genetic algorithm, IABP: BP neural network optimized with immune algorithm.

tion is seen in Fig. 7. As the number of iterations increases, IABP continuously causes the errors to decrease through a global search, reaching an optimal value at the 10th generation. However, the error of the BP neural network optimized with GA (GABP) is reduced, and the algorithm obtains an optimal value at the 27th generation. GABP can easily fall into the local minimum, greatly slowing down the convergence rate. IABP improves the global searching ability, overcomes the default of the local minimum point, and accelerates the convergence rate. Relative to GABP, the convergence rate for IABP increases by 63%.

By using the BP neural network, GABP and IABP, respectively, the results of the conflict detection are shown as Table 3. According to Eq. (9), the errors are $E_{BP} = 0.0248$, $E_{GABP} = 0.0163$, $E_{IABP} = 0.0151$. Relative to the BP neural network, the error of GABP is reduced by 34.3%, while that of IABP is 39.1%. The detection accuracy of IABP has been considerably improved with the lowest

error. Considering the faster convergence rate and the better global convergence ability, IABP is more suitable for conflict detection.

IV. DESIGN AND IMPLEMENTATION OF CONFLICT DETECTION SYSTEM

A. Description of the Constraints Set

In network-based collaborative design, data are exchanged across departments and platforms. However, the uniform expression of constraints is a prerequisite for conflict detection. The powerful ability of eXtensible Markup Language (XML) to describe and present data has been recognized as the standard for electronic data interchange in a multi-disciplinary domain [19-21]. In this paper, we express the constraints in the XML format. Considering the characteristics of the hierarchical constraint network, the design variables, goals and constraints are described in XML and are submitted to a Web server in the XML data format. Fig. 8 shows a description of the constraints based on XML and its detection process.

The set of constraints based on XML is a document with a tree structure that contains elements, attributes and texts. Design variables and goals are written into database according to the corresponding node names and fields of the SQL server. If a node contains child nodes, the values will likely be mapped onto the sub-table, and the child table and the parent table are associated by the primary key and ID. A hierarchical constraint network can be built through the constraints information and can be saved into a database. The constraint relationships are

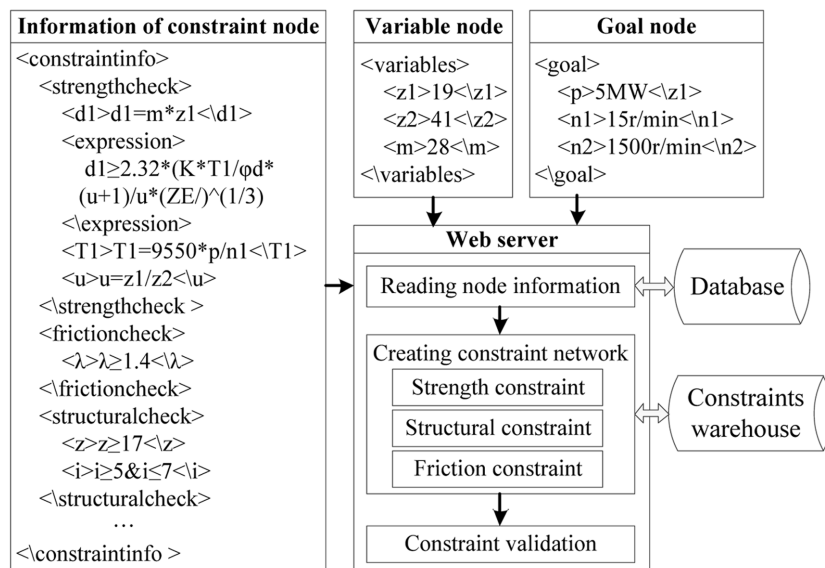


Fig. 8. Detection process of the constraint based on the eXtensible Markup Language (XML) format.

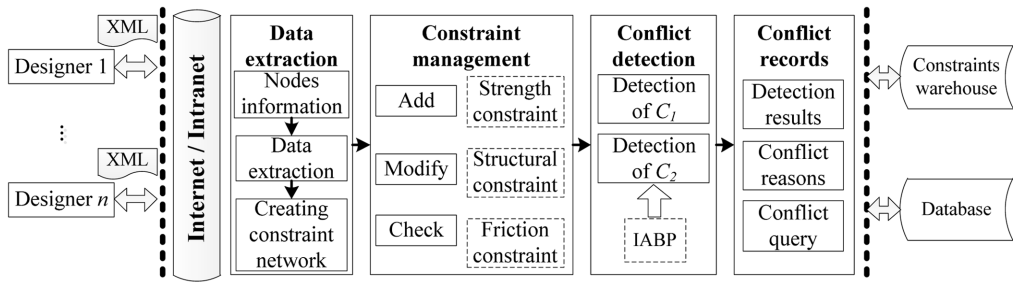


Fig. 9. Architecture of the conflict detection system. IABP: back propagation neural network optimized with immune algorithm.

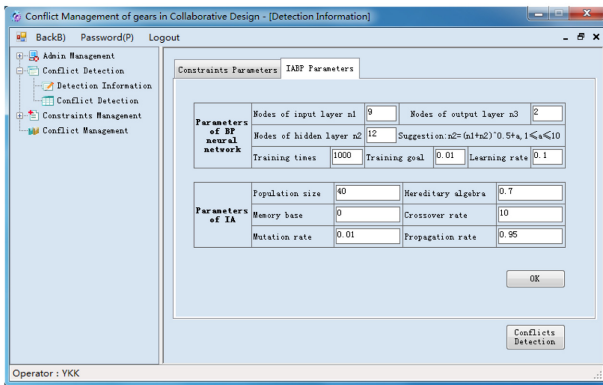


Fig. 10. Interface of the conflict detection system for the gearbox. BP: back propagation, IA: immune algorithm, IABP: BP neural network optimized with IA.

then validated to determine whether conflicts have occurred.

B. Design of the System Architecture

The conflict detection system uses the three-tier C/S architecture that is shown in Fig. 9. The designers access the Web server via the Internet/Intranet, and the system automatically detects conflicts and parameters, and the results of the detection are saved into the database as a repository.

1) Client layer: the designers exchange data from the Web server through the client with XML documents that include the design variables, the design goals, and constraints, etc.

2) Web server layer: there are four sub-modules, including data extraction, constraint management, conflict detection, and records. The data extraction module creates constraint relationships for the set C_1 by reading the XML document and saves the design variables, the design goals and other information into the database. The constraint management module manages the set of constraints by adding, modifying and checking the constraints in terms of the structural constraints, strength constraints, friction constraints, etc. The conflict detection module detects the set C_1 by constraint validation, and set C_2 is detected by calling the package GABP algo-

rihm. The conflict records module returns the results to the client and supports designers that conduct a query for conflicts.

3) Database layer: the system reads and stores the data, including the design variables, goals, constraints, and parameters trained by the IABP.

The system uses C# as the software platform, and the package detection algorithm as a computational engine that runs in the background. The designers input the XML documents and parameters of the conflict detection through the client, and the Web server extracts data and detects conflicts, and then returns the results that support the conflict detection process. Fig. 10 shows the conflict detection interface for the collaborative design of the gearbox.

V. CONCLUSION

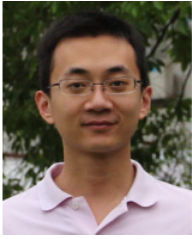
To address constraint satisfaction in collaborative design, this paper divided constraints into a known set of constraints C_1 and an unknown set of constraints C_2 . These are then detected separately, and set C_1 was detected using an interval propagation algorithm that was verified by a constraint network of the adjacent planetary objects while a BP neural network was proposed to detect set C_2 , the weights and thresholds that were optimized using IA. In the experiments for the comparison with the BP neural network optimized by GA, the convergent rate increased by 63%, which indicates that the BP neural network optimized by IA offers an improvement in the performance, speed of convergence and an ability to conduct a global search. When compared with GABP and the BP neural network, the detection accuracy of IABP considerably improves with the lowest error. To this end, the constraints are described by XML, so computers can automatically recognize and establish the constraint network. The collaborative design of a wind planetary gear train is used as an example to develop a conflict detection system in collaborative design based on the C# platform. The conflict detection method is proved to be feasible and effective, and it provides a solution for conflict detection in collaborative design.

ACKNOWLEDGMENTS

This research is sponsored by the National Nature Science Foundation of China (No. 51375350) and by the Fundamental Research Funds for the Central Universities (No. 2012208020205).

REFERENCES

1. P. A. Yvars, "A CSP approach for the network of product lifecycle constraints consistency in a collaborative design context," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 6, pp. 961-970, 2009.
2. X. L. Meng, H. Yi, Z. H. Ni, and Y. Liu, "Research on technologies of constraint-based conflict detection in collaborative design," *Computer Integrated Manufacturing Systems*, vol. 10, no. 11, pp. 1426-1432, 2004.
3. X. P. Hu, J. Wang, and Z. H. Xu, "Constraint network modeling and conflict detection in vertical collaboration design," *Mechanical & Electrical Engineering Magazine*, vol. 26, no. 5, pp. 44-47, 2009.
4. K. Slimani, D. F. Da Silva, L. Medini, and P. Ghodous, "Conflict mitigation in collaborative design," *International Journal of Production Research*, vol. 44, no. 9, pp. 1681-1702, 2006.
5. J. Zhu, X. D. Song, "A Conflict Detection Algorithm Based on the Design History in Collaborative CAD Design," *Computer & Digital Engineering*, vol. 5, no. 40, pp. 79-81, 2012.
6. H. S. Zhao, L. Tian, and B. S. Tong, "Constraint-based conflict detection and negotiation in collaborative design," *Computer Integrated Manufacturing Systems*, vol. 8, no. 11, pp. 896-900, 2002.
7. H. C. Xie, D. Chen, and X. Kong, "Constraint-based conflict detection in collaborative design," *China Mechanical Engineering*, vol. 13, no. 18, pp. 1590-1593, 2002.
8. D. Y. Wang and W. D. Jin, "Conflict detection algorithm in collaborative design," *Computer Applications*, vol. 27, no. 3, pp. 650-652, 2007.
9. Y. Xiong, W. H. Fan, and G. L. Xiong, "Research and realization of distributed conflict detection System," *Computer Engineering*, vol. 35, no. 20, pp. 23-27, 2009.
10. L. Jaulin, "Interval constraint propagation with application to bounded-error estimation," *Automatica*, vol. 36, no. 10, pp. 1547-1552, 2000.
11. M. Klein, "Supporting conflict resolution in cooperative design systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 6, pp. 1379-1390, 1991.
12. J. D. J. R. Avila, A. F. Ramirez, and C. Aviles-Cruz, "Non-linear system identification with a feedforward neural network and an optimal bounded ellipsoid algorithm," *WSEAS Transactions on Computers*, vol. 7, no. 5, pp. 542-551, 2008.
13. D. B. Yuan, X. Li, and N. Zhu, "Penetration depth forecast using BP neural network-based system," *Journal of Computational Information Systems*, vol. 10, no. 12, pp. 5001-5008, 2014.
14. Y. F. Zhang, Z. Y. Xiong, Y. Chen, G. Y. Li, and X. F. Geng, "Immune algorithms based on data processing in intrusion detection," *Journal of Computational Information Systems*, vol. 4, no. 1, pp. 293-300, 2008.
15. K. L. Hsu, H. V. Gupta, and S. Sorooshian, "Artificial neural network modeling of the rainfall-runoff process," *Water Resources Research*, vol. 31, no. 10, pp. 2517-2530, 1995.
16. Z. Wang, D. Miao, and Z. Du. "Modified particle filter algorithm for mobile robot simultaneous localization and mapping," in *Proceedings of International Technology and Innovation Conference 2009 (ITIC 2009)*, Xian, China, 2009, pp. 1-5.
17. D. Xu, F. F. Yap, X. Han, and G. L. Wen, "Identification of spring-force factors of suspension systems using progressive neural network on a validated computer model," *Inverse Problems in Engineering*, vol. 11, no. 1, pp. 55-74, 2003.
18. H. Gopalakrishnan and D. Kosanovic, "Operational planning of combined heat and power plants through genetic algorithms for mixed 0-1 nonlinear programming," *Computers & Operations Research*, vol. 56, pp. 51-67, 2015.
19. Y. Liu, D. Yang, S. Tang, T. Wang, and J. Gao, "Validating key constraints over XML document using XPath and structure checking," *Future Generation Computer Systems*, vol. 21, no. 4, pp. 583-595, 2005.
20. B. Huang, S. Yi, and W. T. Chan, "Spatio-temporal information integration in XML," *Future Generation Computer Systems*, vol. 20, no. 7, pp. 1157-1170, 2004.
21. S. C. Haw and C. S. Lee, "Extending path summary and region encoding for efficient structural query processing in native XML databases," *Journal of Systems and Software*, vol. 82, no. 6, pp. 1025-1035, 2009.



Kangkang Yang

Kangkang Yang is a Ph.D. candidate in School of Power and Mechanical Engineering at Wuhan University, China. His research focuses on computer aided engineering including collaborative design, multi-agent systems and database service.



Shijing Wu

Shijing Wu, Ph.D., is Professor in School of Power and Mechanical Engineering at Wuhan University, China. He has published more than 100 refereed papers in international leading journals and key conferences in the areas of computer aided design, nonlinear dynamics, gear transmission and power equipment. His current research interests include mechanical design theory, electro-hydraulic system, compound planetary gear train and equipment management technology. He is now the director of the Key Laboratory of Hubei Province For Water Jet Theory & new technology and a member of the Academic Committee of Wuhan University.



Wenqiang Zhao

Wenqiang Zhao is a part-time Ph.D. candidate in School of Power and Mechanical Engineering at Wuhan University, China. His research interests include mechanical design theory and methods. He is currently the deputy general manager of Henan Pinggao Electric CO. Ltd., China.



Lu Zhou

Lu Zhou is a postgraduate student in School of Power and Mechanical Engineering at Wuhan University, China. Her research areas include computer design and computer simulation of transmission system.