PERGAMON

Robotics and Computer Integrated Manufacturing 16 (2000) 383–396

Robotics and Computer Integrated Manufacturing

www.elsevier.com/locate/rcim

Design candidate identification using neural network-based fuzzy reasoning

J. Sun[a], D.K. Kalenchuk[b], D. Xue[a], P. Gu[a,*]

[a]*Department of Mechanical and Manufacturing Engineering, The University of Calgary, 2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4*

[b]*Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Saskatchewan, Canada*

Received 17 December 1999; received in revised form 15 March 2000; accepted 3 May 2000

Abstract

Conceptual design has profound impact on success of a product design. Identification of the best conceptual design candidate is a crucial step as design information is not complete and design knowledge is minimal at conceptual design stage. This paper presents a method for design candidate evaluation and identification using neural network-based fuzzy reasoning. The method consists of the following steps: (1) acquisition of customer needs and ranking of their importance, (2) establishment of measurable metrics and their relations with customer needs, (3) development of design specifications and initial evaluation of design candidates, and (4) evaluation and identification of design candidates based on design specifications and customer needs using neural network-based fuzzy reasoning. A case study is given to show the effectiveness of the proposed method and associated algorithms. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Conceptual design; Design evaluation; Fuzzy reasoning; Neural network

1. Introduction

Development of a product undergoes a sequence of processes including conceptual design, embodiment design, detailed design, production process planning, manufacturing, assembly, and so on. With the advances in computer technologies, many of these design and manufacturing activities have been computerized to assist engineers such as computer-aided design (CAD), computer-aided manufacturing (CAM), computer-aided process planning (CAPP), and so on [1]. These computerized systems have significantly improved quality and productivity of design and manufacturing processes. However, most of the CAD systems focus on detailed stage. Very few computer-aided design systems are available to support conceptual design activities [2].

Conceptual design is a process to develop design candidate based upon design requirements. The design requirements are usually defined based upon customer needs, benchmarking of products from competitors and other analysis. These requirements are then translated into measurable technical attributes that are easily used for evaluating design candidates. Among all feasible candidates, the best candidate is selected for further development. Mapping from design requirements to design candidates and evaluation to these candidates, however, are non-trivial tasks, which require substantial research efforts.

The systematic study on conceptual design was started in 1970s [3]. Pahl and Beitz [3] defined design functional primitives such as gears and shafts, and stored them in libraries. At conceptual design stage, a design function is usually decomposed into a number of sub-functions. A design solution is accomplished by selecting design primitives to satisfy these sub-functions. Other systematic approaches for formulating conceptual design include the axiomatic design method [4] and quality function deployment (QFD) method [5]. In axiomatic design, mapping from functional requirements (FRs) to design parameters (DPs) is modeled by a matrix [4]. A design is evaluated in terms of independence of functional requirements and information content. The quality function deployment method is an approach to first identify customer requirements and their importance measures, and then translate these data into technical attributes and importance measures [5]. Design specifications are developed by comparing existing designs for improving the competitiveness of the new product.

* Corresponding author.
E-mail address: gu@enme.ucalgary.ca (P. Gu).

0736-5845/00/$ - see front matter © 2000 Elsevier Science Ltd. All rights reserved.
PII: S0736-5845(00)00017-X

At conceptual design stage, a number of design candidates are usually generated, which satisfy all design requirements. Therefore, identification of the best design candidate for further development must be carried out. Recent development in concurrent engineering allows product life-cycle performance measures to be used as criteria for evaluating design candidates [6]. Typical life-cycle performance measures used in these concurrent design systems include manufacturability [7–9], assemblability [8,10,11], serviceability [12,13], and recyclability [14]. However, it is considerably difficult to achieve these down-stream product development life-cycle performance measures accurately at the early conceptual design stage.

The research on computer-based automated design candidate generation and evaluation was started with the development of knowledge-based design systems [15]. In these systems, relations between functional requirements and design candidates are described by rules using the IF–THEN data structure. When the customer requirements are matched with the condition part of a rule, the result part of the rule is then executed to generate design candidates. Since it is difficult to develop systematic rules for conceptual design in general, case-based reasoning has been used to generate design concepts. Hashemian and Gu [16] reported the development of computer-assistant tool for conceptual design with several case retrieval techniques.

The recent advances in soft computing techniques, including fuzzy logic and neural networks, provide new tools for developing intelligent systems with the capabilities of modeling uncertainty and learning [17–21]. Applications of soft computing in conceptual design resulted in computerized systems. Xue and Dong developed a fuzzy-based design function coding system to identify design candidates from design functions [22]. Venugopal and Narendran utilized the Hopfield neural network to preserve design patterns [23]. The most appropriate design pattern is retrieved during conceptual design stage using the neural network. Kamarthi et al. employed a multi-layer feedforward neural network for modeling the relations to retrieve design data from requirements [24]. Back-propagation algorithm was adopted to train these relationships between input and output data sets. Bahrami et al. used fuzzy associative memory (FAM), a two-layer feedforward neural network, to describe the relationships between customer needs and design candidates [25]. Chang and Tsai utilized adaptive resonance theory (ART) neural network for identifying similar design candidates for design improvement [26]. Gu et al. [27,28] have used genetic algorithms, simulated annealing and fuzzy logic in module formation for life cycle engineering. These methods were successful and effective in module formation for engineering design.

Despite of this progress, the fuzzy and neural network-based design systems suffer from the following two major problems:

(1) The fuzzy logic-based intelligent design systems allow the designers to model the concepts and relations with uncertainty, therefore evaluating the design in a more effective manner. However, fuzzy reasoning requires considerable computation efforts due to the complexity of the fuzzy membership functions and large number of rules.
(2) In the intelligent systems developed using Hopfield, feedforward, and ART neural networks, the input and output data are described by conventional two-value logic without considering uncertainty. The Hopfield, FAM, and ART neural networks have difficulties to store a large number of design patterns.

The objective of this research is to develop a systematic approach to identify the best design candidate based upon customer requirements, thus providing the basis for implementing the next generation CAD systems with the conceptual design functions. A neural network-based fuzzy reasoning method is introduced for solving the problems of modeling uncertainty and improving computational efficiency in the process of identifying design candidate. The remaining of the paper is organized as follows: Section 2 gives an overview of fuzzy reasoning and neural networks, and introduces a neural network-based fuzzy reasoning model. Section 3 proposes a four-step design candidate identification method. Section 4 presents application of the neural network-based fuzzy reasoning for design candidate identification. Section 5 gives a case study example to show the effectiveness of the developed method. Section 6 summarizes this research.

2. Neural-network-based fuzzy reasoning

Identification of the design candidate is conducted through neural network-based fuzzy reasoning. In this section, we first introduce fuzzy sets, feedforward neural networks, and the neural-network-based fuzzy reasoning.

2.1. Fuzzy sets and fuzzy reasoning

The concept of fuzzy set was introduced in the mid-1960s to overcome the limitation of the conventional crisp sets, where the relation of a data and a set is described by either “in” or “out” [29]. In fuzzy set theory, the relation between a member and a set is represented by a membership function in the range of 0 to 1. Fig. 1 depicts the concepts of “tall” and “short” of human heights in conventional crisp sets and fuzzy sets, respectively.

Fuzzy logic was developed by representing antecedent and consequence parts of IF–THEN rules using fuzzy sets. In fuzzy reasoning, logical “*and*” and logical “*or*”

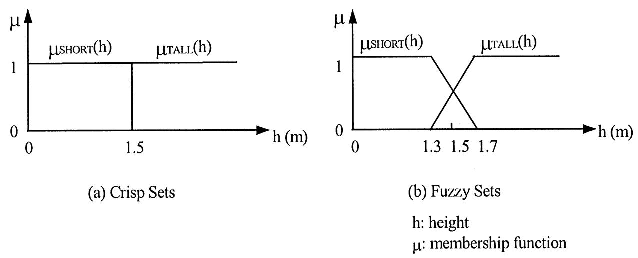

Fig. 1. Crisp sets and fuzzy sets.

operations produce the minimum and maximum membership function values. The output value can be obtained by calculating the center of gravity of the output membership function. A fuzzy reasoning example considering two rules is illustrated in Fig. 2. For each rule, the membership function measures for the two input variables, x_0 and y_0, are obtained first. The smaller value is selected as the measure to evaluate the matching of the rule. The result membership function of fuzzy reasoning considering only one rule is the minimum of the membership function at the THEN part of the rule and the rule matching measure. The result membership function, $\mu_c(z)$, considering all relevant rules is achieved by obtaining the maximum value of these result membership functions for these rules. The value of the output variable, z_0, is the center of gravity of the output membership function $\mu_c(z)$, calculated by Eq. (1)

$$z_0 = \frac{\int_{z_{\min}}^{z_{\max}} \mu_c(z) z \, dz}{\int_{z_{\min}}^{z_{\max}} \mu_c(z) \, dz}. \tag{1}$$

With an increase of number of rules and complexity of membership functions, the calculation of output variable values becomes difficult. Therefore, an improvement in computational efficiency is important. The recent advances in artificial neural networks provide a means to learn the fuzzy input/output relations by training the neural network, and use the neural network to obtain the fuzzy reasoning result in an efficient manner.

2.2. Feedforward neural network

A multi-layer feedforward neural network is an effective tool for modeling non-linear relationships between input and output data, when the mathematical expressions representing relations between these input and output parameters cannot be formulated easily [18]. A three-layer feedforward neural network is illustrated in Fig. 3. This neural network architecture consists of an input layer with n input nodes, a hidden layer with p-hidden nodes, and an output layer with q output nodes. The nodes at two adjacent layers are connected by arcs with weights u_{ij} ($i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, p$) and w_{jk} ($j = 1, 2, \ldots, p$; $k = 1, 2, \ldots, q$). A feedforward neural network is first trained to learn the input and output relations using correct data sets, and is then used to calculate output result when a set of input data are provided to the neural network.

Training of the feedforward neural network is conducted through adjusting the weight values of connection arcs by the back-propagation algorithm [18]. The back-propagation algorithm is an iterative gradient algorithm designed to minimize the errors between the actual output of a multi-layer feedforward neural network and the desired output. This algorithm consists of several steps given in Appendix A.

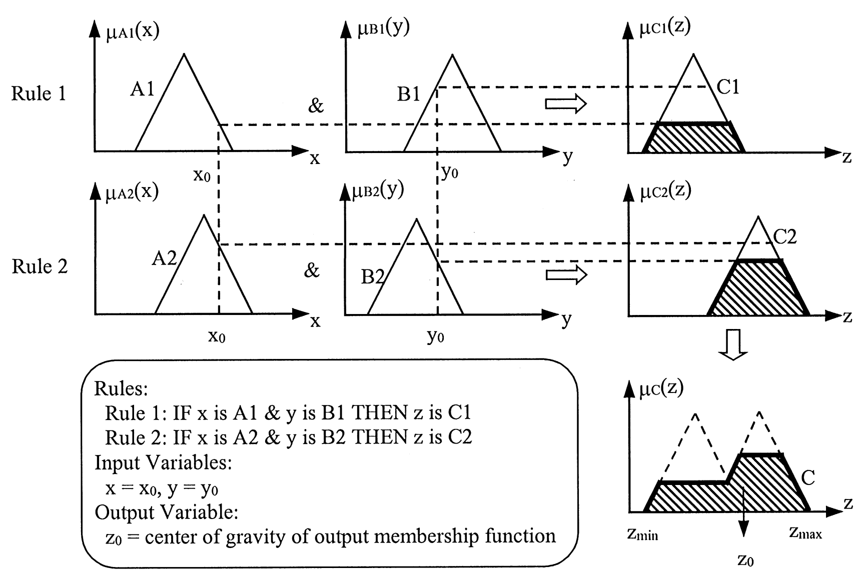

Fig. 2. Fuzzy reasoning.

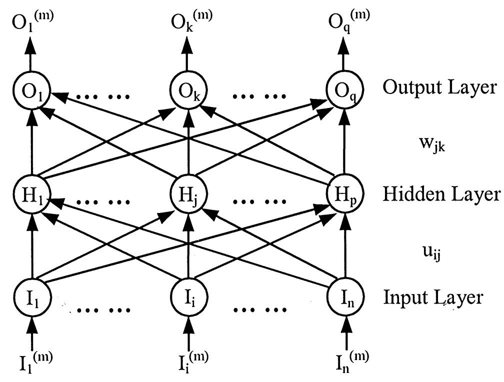

Fig. 3. A feedforward neural network.

2.3. Neural network-based fuzzy reasoning

In fuzzy reasoning, the relations between input fuzzy sets and output fuzzy sets are described by fuzzy rules. When input variable values are provided, output variable values are achieved using relations defined in fuzzy rules through fuzzy reasoning, as introduced in Section 2.1. Computational effort required to obtain output variable value, however, is a non-trivial task due to the complexity of fuzzy set membership functions and large number of fuzzy rules. In this research, a feedforward neural network is employed to model the non-linear relations between input and output variables in fuzzy reasoning to improve computational efficiency.

The feedforward neural network for fuzzy reasoning consists of three layers: an input layer, a hidden layer, and an output layer, as shown in Fig. 4. The input nodes are used to describe the fuzzy membership functions for different fuzzy sets. Since an input variable is associated with a number of fuzzy sets, these input nodes are organized in groups based upon these variables. In the example shown in Fig. 4, m input nodes and n input nodes are used for representing fuzzy set membership function measures when input variables x and y are assigned to x_0 and y_0. The output nodes are used to describe the output membership function $\mu_o(z)$ for output variable z. Each node corresponds to the membership function measure when z is assigned with a particular value. When the input data are provided, the output membership function can be calculated using the neural network. The output variable value, z_0, is the center of gravity of the output membership function.

The neural network is trained using available correct data sets through the back-propagation algorithm given in the appendix. Each correct data set is obtained by assigning values to input variables, calculating the membership function measures of input nodes, and obtaining output membership function using fuzzy reasoning. Since all the fuzzy rules should be involved in the training process, all these rules can be encoded in the neural network. When a set of input data is provided to the neural network, these encoded rules are activated to different degrees simultaneously. The output membership function generated by the neural network computation is the one considering all relevant fuzzy rules. The neural-network-based fuzzy reasoning serves as a tool for design candidate identification in this research.

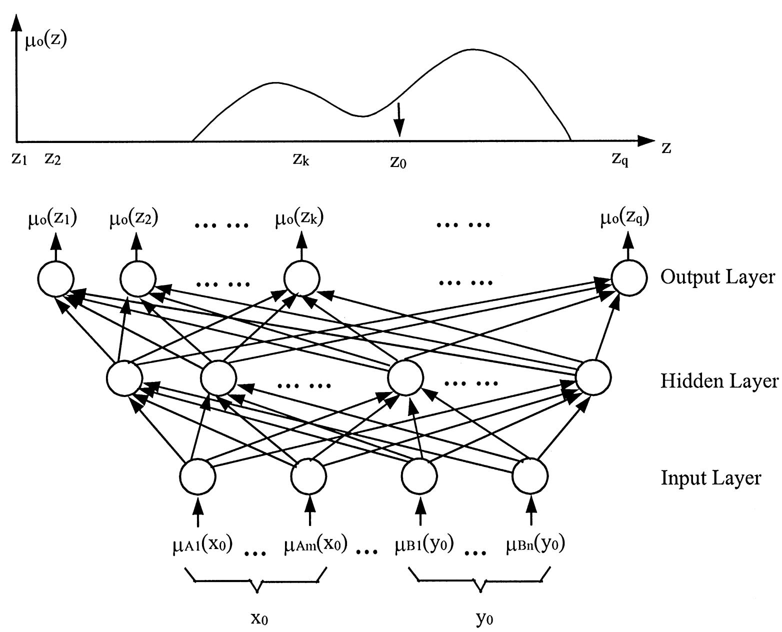

Fig. 4. A feedforward neural network for fuzzy reasoning.

3. A design candidate identification method

Conceptual design is a process to develop design candidates from design requirements for further product development. Design requirements are usually first defined by interpreting customer needs, benchmarking of existing products and engineering analysis. These needs are then translated into measurable technical attributes, usually called metrics. The relationships between customer needs and metrics need to be defined in order to evaluate design candidates. The best candidate should satisfy customer needs.

In this research, a four-step method is proposed for evaluation and subsequent identification of design candidates. These four steps are: (1) acquisition of customer needs and ranking of their importance; (2) establishment of measurable metrics and their relations with customer needs, (3) development of design specifications and initial evaluation of design candidates, and (4) evaluation of design candidates for selection of the best candidate.

The importance of customer needs, the relationships between customer needs and metrics, and the level of satisfaction of customer needs are modeled in fuzzy sets. Satisfaction of design specifications is described by conventional crisp set. Fuzzy rules are employed to model the knowledge to evaluate the level of satisfaction of design candidates using design specification, importance of customer needs, and relationships between customer needs and design metrics. Fuzzy reasoning is carried out using the trained feedforward neural network.

3.1. Acquisition of customer needs and ranking of their importance

A conceptual design is initiated from identifying the customer needs. These needs are generally gathered from market surveys and described by qualitative expressions such as “the cost should be low” and “the product should look good.” Improvement in product competitiveness can be reached by designing the product that can better satisfy the customer needs.

Among all the customer requirements, some are considered more important than the others. Therefore, improvement on these more important requirements has stronger influence on product competitiveness. Identification of importance measures of customer needs, however, is a complicated task. In this research, the pair wise comparison method is utilized [30].

Pair wise comparison starts with comparing the relative importance, or importance ratio, of two selected items. If n items are associated with n weights, $w_1, w_2, \ldots, w_n$, the relative importance, a_{ij}, considering the ith item and the jth item is obtained as

$$a_{ij} = \frac{w_i}{w_j}. \tag{2}$$

The pair wise ratios satisfy

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = n \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}. \tag{3}$$

Since an item is equally important as itself, the value of a diagonal element in the matrix is 1 ($a_{ii} = 1$), and values of the elements in the upper triangle of the matrix are the reciprocal values of the elements in the lower triangle of this matrix, only $n(n-1)/2$ times of comparisons are needed. For ease of explanation, this equation is described as

$$A\mathbf{w} = n\mathbf{w} \tag{4}$$

or

$$(A - nI)\mathbf{w} = \mathbf{0}, \tag{5}$$

where I is a $n \times n$ identity matrix. From this equation, it is apparent that n is an eigenvalue of A, and $\mathbf{w}$ is an eigenvector for eigenvalue n.

In a general case, we cannot give the precise values of w_i/w_j, but only estimates of them. The estimation errors result in inconsistency of the data in the pair wise ratio matrix. Saaty introduced a consistency index, CI, as a measure to evaluate the deviation from consistency of the pair wise ratios [30]. CI is calculated by

$$\mathrm{CI} = (\lambda_{\max} - n)/(n-1), \tag{6}$$

where $\lambda_{\max}$ is the maximum eigenvalue of A considering estimation errors. When values of the elements of a reciprocal matrix are generated randomly, the consistency index for this matrix is represented as RI. The average RI values for different orders of matrices are summarized in Table 1 [30].

The ratio of CI to RI for the same order matrices is called the consistency ratio (CR). A pair wise ratio matrix with consistency ratio less than 0.10 is considered as a good one to calculate the weights of the items.

In the process to identify importance measures of customer needs, customers are asked to specify how a particular need is more important than another one. The comparison values, a_{ij}, are defined on a scale of 1 to 9, as shown in Table 2.

The calculated weights for customer needs are scaled to the range between 0 and 1 for representing the importance measures. Four fuzzy sets have been developed for

Table 1
Consistency indices for random reciprocal matrices with different orders

Orders	2	3	4	5	6	7	8
RI	0.00	0.52	0.90	1.12	1.24	1.32	1.41

Table 2
Scales for comparison of customer needs

a_{ij}	Comparison of the ith need and the jth need
1	The ith need is equally important as the jth need
3	The ith need is moderately more important than the jth need
5	The ith need is strongly more important than the jth need
7	The ith need is very strongly more important than the jth need
9	The ith need is extremely more important than the jth need
2, 4, 6, 8	These are intermediate comparison values
Reciprocals	These are values for inverse comparisons a_{ji}

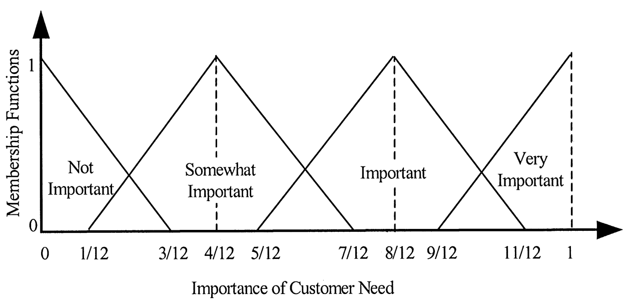

Fig. 5. Membership functions for modeling the importance of customer needs.

modeling the *importance of customer needs*: (1) Not Important, (2) Somewhat Important, (3) Important, and (4) Very Important, as shown in Fig. 5.

3.2. Establishment of measurable metrics and their relations with customer needs

Since customer needs are usually described by qualitative expressions, evaluation of design candidates in terms of customer satisfaction is difficult to be conducted directly. Therefore, translation of customer needs to measurable metrics is required.

The measurable metrics, usually called design attributes, are terms used by design engineers. Each metric is associated with a value and a unit. For instance, diameter and module are two metrics of a gear. Because a design candidate is described by metrics, evaluation of design candidates based on metrics can be easily accomplished.

To evaluate the design candidates based on customer needs, relationships between metrics and needs, or called the capability of metrics to measure needs, have to be developed. In this research, these relations are described by numbers between 0 and 1, with 1 indicating a perfect measure to the need and 0 indicating an impossible measure to the need. The relations are correspondent to four fuzzy sets representing the capability of metrics

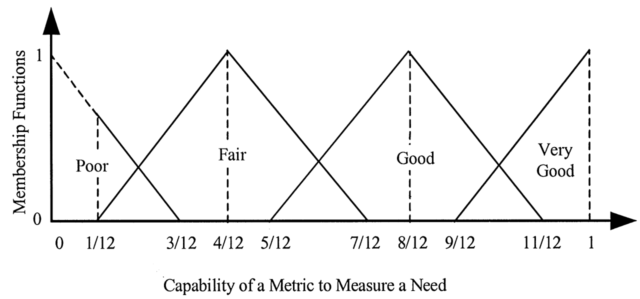

Fig. 6. Membership functions for modeling the capability of metrics to measure needs.

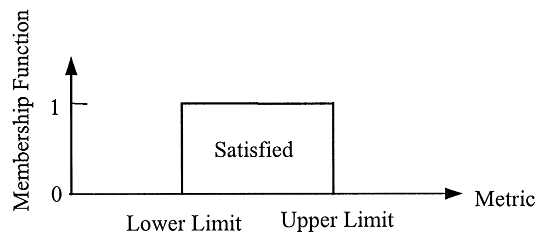

Fig. 7. Membership function for modeling the satisfaction of specifications.

to measure the needs: (1) Poor, (2) Fair, (3) Good, and (4) Very Good, as shown in Fig. 6. It is noted that when the relation is too weak ($<\frac{1}{12}$), the metric cannot measure a need properly.

3.3. Development of design specifications and initial evaluation of design candidates

Design specifications state the required measures of technical metrics. Each specification is usually described by lower and upper bounds. Specifications are developed based upon customer requirements, competitive analysis of similar products, and product testing. Evaluation results are represented by conventional crisp sets, as shown in Fig. 7. In this research, a crisp set is considered as a special case of fuzzy set, where the membership function measure can only be selected as 0 or 1.

3.4. Evaluation of design candidates based on customer needs using fuzzy reasoning

Evaluation of design candidates based upon customer needs is carried out using the three previously achieved measures: (1) satisfaction of metrics to specifications, (2) capability of metrics to measure needs, and (3) importance of customer needs. This concept is illustrated in Fig. 8.

Fuzzy rules are developed for evaluating design candidates with three defined measures. The IF part of a fuzzy

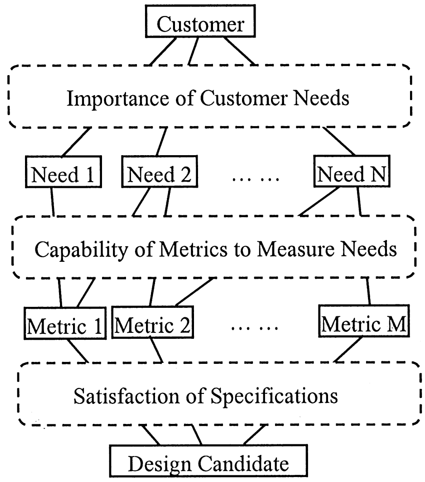

Fig. 8. Design candidate evaluation based upon satisfaction of customer needs.

rule is composed of three expressions linked by logical-and (&), representing the fuzzy sets corresponding to the three achieved measures. The THEN part of a rule describes the fuzzy set representing satisfaction of design candidate to customer needs. Each time, only one metric and one need are considered to evaluate a design candidate. Evaluation of the candidate considering all metrics and needs is based upon these individual evaluation results.

Satisfaction of customer needs is also described by a measure between 0 and 1. Five fuzzy sets are used to model this measure: (1) Very Poor, (2) Poor, (3) Fair, (4) Good, and (5) Very Good, as shown in Fig. 9.

Thirty-two rules have been developed to represent the fuzzy relations for design candidate evaluation, as summarized in Table 3. These rules describe the knowledge such as:

IF the metric satisfies the specification, and the metric has very good capability to measure the need, and the need is very important,
THEN the design candidate is very good to satisfy the customer need.
IF the metric satisfies the specification, and the metric has good capability to measure the need, and the need is somewhat important,
THEN the design candidate is good to satisfy the customer need.
IF the metric does not satisfy the specification, and the metric has very good capability to measure the need, and the need is very important,
THEN the design candidate is very poor to satisfy the customer need.

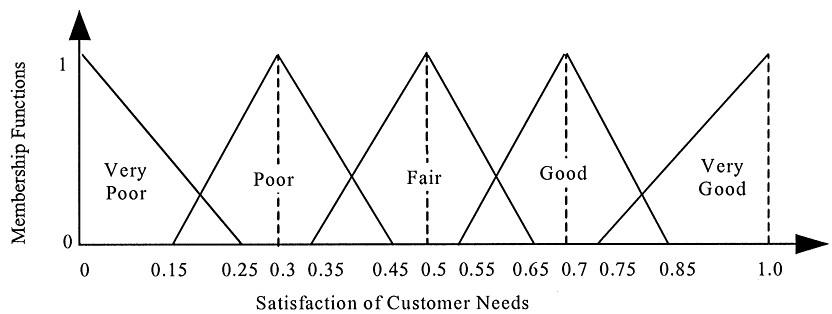

Fig. 9. Membership functions for modeling satisfaction of customer needs.

Table 3
Fuzzy rules

		Capability of metrics to measure needs			
		Very good	Good	Fair	Poor
(a) *Fuzzy rules when metric satisfies the specification*					
Importance of needs	Very important	Very good	Very good	Very good	Good
	Important	Very good	Very good	Good	Good
	Somewhat important	Very good	Good	Good	Fair
	Not important	Good	Good	Fair	Fair
(b) *Fuzzy rules when metric does not satisfies the specification*					
Importance of needs	Very important	Very poor	Very poor	Very poor	Poor
	Important	Very poor	Very poor	Poor	Poor
	Somewhat important	Very poor	Poor	Poor	Fair
	Not important	Poor	Poor	Fair	Fair

When N customer needs and M metrics are used in design candidate evaluation, suppose the evaluation measure consider only the ith need and jth metric is described by S_{ij}, design evaluation measure considering the jth metric and all needs is calculated by

$$S_j = \frac{1}{n_j} \sum_{i=1}^{n_j} S_{ij} \quad (j = 1, 2, \ldots, M), \tag{7}$$

where n_j ($n_j < N$) is the number of involved needs in calculating S_j. Design candidate evaluation considering all needs and metrics is achieved using

$$S = \frac{1}{M} \sum_{j=1}^{M} S_j. \tag{8}$$

If multiple candidates are considered, the one with the highest evaluation measure is selected as the best candidate for further product development.

Calculation of the customer satisfaction measure is a complicated task. In this research, a neural network is employed for fuzzy reasoning to improve computational efficiency. The details of the neural network and fuzzy reasoning will be given in Section 4.

4. Neural network-based fuzzy reasoning for design evaluation

A feedforward neural network, as shown in Fig. 10, has been developed for fuzzy reasoning to evaluate design candidates. This neural network consists of three layers: an input layer, a hidden layer, and an output layer. The input layer has nine input nodes, representing the nine fuzzy membership functions (Figs. 5–7) used as antecedents of rules. These nine input nodes are organized in three groups, corresponding to the three input variables: (1) satisfaction of a metric to a specification, (2) capability of a metric to measure a need, and (3) importance of a customer need. The input nodes are summarized in Table 4. The output layer has 21 nodes, representing the membership function of the fuzzy set to measure the satisfaction of metrics to customer needs.

Training of the neural network is conducted using the correct input/output data sets obtained from the 32 fuzzy rules. For instance, the fuzzy rule

IF the metric satisfies the specification, and the metric has good capability to measure the need, and the need is somewhat important,

THEN the design candidate is good to satisfy the customer need.

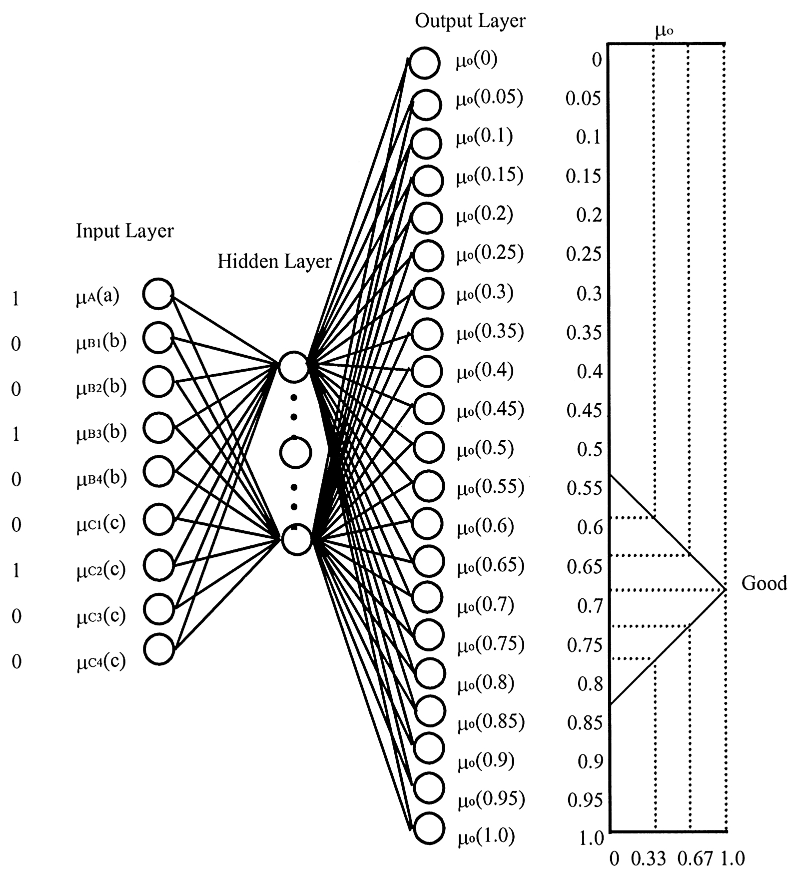

Fig. 10. Neural network for fuzzy reasoning to evaluate design candidates.

Table 4
Input variables and their corresponding nodes

Input variables	Input nodes	Fuzzy sets
Satisfaction of a metric to a specification: *a*	μ_A	Satisfied
Capability of a metric to measure a need: *b*	μ_{B1}	Poor
	μ_{B2}	Fair
	μ_{B3}	Good
	μ_{B4}	Very good
Importance of a customer need: *c*	μ_{C1}	Not important
	μ_{C2}	Somewhat important
	μ_{C3}	Important
	μ_{C4}	Very important

is correspondent to the training data:

Input data set: [1, 0, 0, 1, 0, 0, 1, 0, 0]
Output data set: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.33, 0.67, 1.00, 0.67, 0.33, 0, 0, 0, 0]

as shown in Fig. 10. Since 32 rules are used, 32 correct input/output data sets are obtained for training the neural network.

In neural network-based fuzzy reasoning for design candidate evaluation, first the nine membership functions for the 3 input variables are calculated using Figs. 5–7. These nine input measures are then provided to the neural network for calculating the output fuzzy set membership function. The output variable value is obtained as the center of gravity of the output membership function.

5. A case study

A case study of front suspension fork design of mountain bikes was used to demonstrate the effectiveness of the introduced design candidate identification method. The data provided in reference [31] were used for this case study. Six different front suspension fork design candidates for mountain bikes were selected and evaluated based upon the proposed method to identify the best one for further product development. One of these design candidates is shown in Fig. 11. The identification of the best design candidate was carried out in the following four steps.

Step 1: *Acquisition of customer needs and ranking of their importance*. The customer needs were acquired through informal survey of customers. In the mountain bike design, six needs were identified by mountain bike cyclists, as listed in Table 5.

The importance measures of these customer needs were defined using the pair-wise comparison method, as

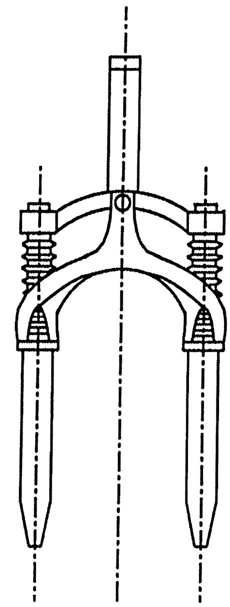

Fig. 11. A design of front suspension fork.

Table 5
List of customer needs

No.	Customer needs
1	Reduces vibration to the hands
2	Allows easy traversal of slow, difficult terrain
3	Enables high-speed descents on bumpy trails
4	Preserves steering characteristics of bike
5	Remains rigid during hard cornering
6	Is lightweight

Table 6
Pair wise comparison data

Customer needs	Customer needs					
	1	2	3	4	5	6
1	1	5	1	4	5	2
2	$\frac{1}{5}$	1	$\frac{1}{5}$	$\frac{1}{2}$	1	$\frac{1}{3}$
3	1	5	1	4	5	2
4	$\frac{1}{4}$	2	$\frac{1}{4}$	1	3	$\frac{1}{2}$
5	$\frac{1}{5}$	1	$\frac{1}{5}$	$\frac{1}{3}$	1	$\frac{1}{3}$
6	$\frac{1}{2}$	3	$\frac{1}{2}$	2	3	1

introduced in Section 3.1. The obtained importance ratio matrix is shown in Table 6. The eigenvalue of this matrix, λ_{max}, was calculated as 0.68. Therefore, the consistency index, CI, was obtained as 0.016 using Eq. (6). Since the consistency index for a 6-order random reciprocal matrix, RI, was 1.24 according to Table 1, the ratio of CI to RI was then calculated as 0.013. Because this ratio is much smaller than the acceptable ratio 0.10, the matrix

Table 7
Calculated importance measures of customer needs

Customer needs	Importance measures
1	1.00
2	0.18
3	1.00
4	0.32
5	0.17
6	0.54

Table 8
Design metrics

No.	Metrics	Units
1	Attenuation from dropout to handlebar at 10 Hz	dB
2	Spring preload	N
3	Maximum value from the Monster	g
4	Minimum descent time on test track	s
5	Maximum travel (26 inch wheel)	mm
6	Rake offset	mm
7	Lateral stiffness at the tip	kN/m
8	Total mass	kg

Table 9
Capability of metrics to measure customer needs

Customer needs	Metrics							
	1	2	3	4	5	6	7	8
1	0.8		0.7	0.2				
2		0.6						
3	0.3		0.5	0.8				
4					0.2	0.4		
5		0.7					0.6	
6								1.0

Table 10
Design specifications

Metric no.	Units	Lower limits	Upper limits
1	dB	10.0	None
2	N	480.0	800.0
3	g	0	3.5
4	s	0	13.0
5	mm	33.0	50.0
6	mm	37.0	45.0
7	kN/m	65.0	None
8	kg	0	1.4

was considered with a very good level of consistency. The corresponding eigenvector for the eigenvalue λ_{max} was calculated as (5.6, 1.0, 5.6, 1.8, 0.95, 3.0). The elements of this eigenvector were then scaled to the range between 0 and 1 for representing the importance measures of the customer needs. The obtained importance measures are listed in Table 7.

Step 2: *Establishment of measurable metrics and their relations with customer needs*. After the customer needs were established and their importance was ranked, the customer needs were then associated with metrics that enable the design engineers to understand the needs of the customers in technical terms. The eight metrics developed for the front suspension fork design are summarized in Table 8. The Monster is a shock test used by *Mountain Bike* magazine.

Two of these eight metrics, the maximum value from the Monster and the minimum descent time on the test track, were used for the standardized tests. The others are measurable properties of mountain bike suspension forks. The measurement of the maximum value from the Monster was made by obtaining the acceleration rate at the handlebar of a bike when the front wheel was subjected to an impulse disturbance.

The metrics were then associated with the customer needs by defining their relationships, as shown in Table 9. These relationships represented the capability of metrics to measure the customer needs. When a relationship was too weak ($<\frac{1}{12}$), the metric could not measure the need properly. It was impossible to evaluate the design candidate and the relationship was replaced by a blank in the table.

Step 3: *Development of design specifications and initial evaluation of design candidates*. The design specifications considering the 8 metrics were obtained by the design engineers and are given in Table 10. Each specification is described by a lower bound and an upper bound with units.

Six design candidates were selected for evaluation. The measures of metrics for the six design candidates are summarized in Table 11. When a metric satisfied the design specification, the evaluation of this metric in terms of design specification satisfaction was represented as 1, otherwise 0.

Step 4: *Evaluation of design candidates based on customer needs using fuzzy reasoning*. The evaluation of a design candidate in terms of customer need satisfaction was conducted by fuzzy reasoning using the developed feed-forward neural network. Each time, only one design metric and a customer need were considered in evaluation. The evaluation considering all the 8 metrics and 6 needs was carried out based upon these individual evaluations.

For instance, when the fourth metric, *minimum descent time on test track*, and the first need, *reduces vibration to the hands*, were considered for evaluating the first design candidate, the three input variables and their corresponding nine fuzzy membership function measures were calculated as shown in Table 12. These nine membership function measures were used as the input data for the

Table 11
Design candidates and their metric measures

Metric no.	Units	Design 1 *ST Tritrack*	Design 2 *Maniray 2*	Design 3 *Rox Tahx Quadra*	Design 4 *Rox Tahx Ti21*	Design 5 *Tonka Pro*	Design 6 *Gunhill Head Shox*
1	dB	8.0[a]	15.0	10.0	15.0	9.0[a]	13.0
2	N	550.0	810.0[a]	500.0	710.0	480.0	680.0
3	g	3.6[a]	3.2	3.7[a]	3.3	3.7[a]	3.4
4	s	13.0	11.3	12.6	11.2	13.2[a]	11.0
5	mm	28.0[a]	48.0	43.0	46.0	33.0	38.0
6	mm	41.5	39.0	38.0	38.0	43.2	29.0[a]
7	kN/m	59.0[a]	110.0	85.0	85.0	65.0	130.0
8	kg	1.409[a]	1.385	1.409[a]	1.364	1.222	1.100

[a]The metric value that does not satisfy the design specification.

Table 12
Input data considering the fourth metric and the first need for the first design candidate

Variable names	Variable values	Fuzzy sets	Membership functions
Satisfaction of the metric to the specification: a	1	Satisfied	1.00
Capability of the metric to measure the need: b	0.2	Poor	0.20
		Fair	0.45
		Good	0.00
		Very good	0.00
Importance of the customer need: c	1.0	Not important	0.00
		Somewhat important	0.00
		Important	0.00
		Very important	1.00

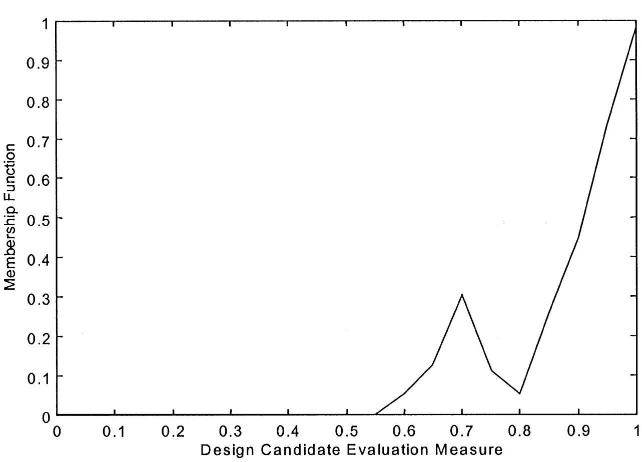

Fig. 12. Membership function of the output fuzzy set.

trained neural network. The output membership function, for the fuzzy set representing satisfaction of the design in terms of customer needs, was calculated by this neural network (see Fig. 12). The evaluation measure was obtained by calculating the center of gravity of the output membership function. The evaluation measure considering the fourth metric and first need for the first design candidate was obtained as 0.89.

In the same way, the evaluation measures considering other metrics and needs for the first design candidate were obtained, as shown in Table 13. As mentioned earlier, when a metric and a need had no relationship, or had a very weak relationship ($<\frac{1}{12}$), it was considered impossible to achieve the evaluation measure. The correspondent place to describe the evaluation result was filled with a blank. For each metric, the average evaluation measure, S_j ($j = 1, 2, \ldots, M$), considering all the customer needs was then obtained using Eq. (7). The evaluation considering all metrics and needs, S, was calculated as 0.38 by averaging all the S_j measures using Eq. (8).

When an evaluation measure was greater than 0.5, the design was considered good, otherwise poor. Since the evaluation measure for the first design candidate was 0.38, this candidate was considered as a poor design.

The evaluation measures of the six selected candidates considering the eight metrics are summarized in Table 14. From this table, the ranking of these six design candidates was also identified. The fourth design candidate was considered as the best, due to its highest evaluation value. Therefore, this candidate was selected for further development.

Table 13
Evaluation to the first design candidate[a]

Customer needs	Metrics							
	1	2	3	4	5	6	7	8
1	0.06		0.07	0.89				
2		0.68						
3	0.07		0.06	0.95				
4					0.31	0.70		
5		0.70					0.25	
6								0.06
Average measures S_j	0.065	0.69	0.065	0.92	0.31	0.70	0.25	0.06

[a]Evaluation result $S = 0.38$.

An analysis of the design evaluation results was carried out and the results of the analysis are given below:

(1) The evaluation measures of design candidates are primarily determined by specification satisfaction. When a metric satisfies a specification, the measure is generally high (> 0.5), otherwise low (< 0.5). When most of the design specifications are satisfied, the candidate is considered good. For instance, among all the 6 design candidates shown in Table 14, only the design 4 satisfies all the specifications, therefore providing the highest evaluation measure.
(2) The importance measures of customer needs and relationships between metrics and needs also influence the final evaluation results. For instance, both metrics 5 and 6 are related to need 4 with measures of 0.2 and 0.4, respectively. Considering design 2, even both of these two metrics satisfy the design specifications, the evaluation measure considering metric 6 is better than the evaluation measure considering metric 5. In the same way, if a need is considered more important than the other, the evaluation measure considering the first need is higher than the measure considering the second need.
(3) The evaluation measures also provide guidelines for design modifications and redesigns. The measures below 0.5 indicate poor metric values that generally do not satisfy the customer needs. Therefore, these metric values should be modified. When a number of metrics do not satisfy design specifications, the one with the lowest evaluation measure should be changed first due to its strongest influence on the poor evaluation measure of the design. Even for the metrics that satisfy specifications, the metrics with evaluation measures near 0.5 should also be adjusted to improve the overall evaluation measure for the design candidate.

6. Conclusions

This paper introduced a design method for design candidate identification based upon neural network-based fuzzy reasoning. The best design candidate was identified through four steps: (1) acquisition of customer needs and ranking of their importance measures, (2) establishment of measurable metrics and their relations with customer needs, (3) development of design specifications and initial evaluation of design candidates based on design specifications, and (4) evaluation of design candidates based on customer needs. A case study example was given to show the effectiveness of the introduced design candidate identification method.

As in the conceptual design stage, information is incomplete and design knowledge is limited. The design activities are mainly creative in nature. Designers usually generate a large number of design candidates in order to find the best one. The identification of the design candidate is a challenging task. The proposed method is effective in design candidate identification through fuzzy reasoning, when the customer needs, design metrics, and

Table 14
Design candidate evaluation results

Metrics	Design 1	Design 2	Design 3	Design 4	Design 5	Design 6
1	0.065[a]	0.935	0.935	0.935	0.065[a]	0.935
2	0.69	0.29[a]	0.69	0.69	0.69	0.69
3	0.065[a]	0.935	0.065[a]	0.635	0.185[a]	0.635
4	0.92	0.92	0.92	0.92	0.085[a]	0.92
5	0.31[a]	0.69	0.69	0.69	0.69	0.69
6	0.70	0.70	0.70	0.70	0.75	0.23[a]
7	0.25[a]	0.71	0.71	0.71	0.71	0.71
8	0.06[a]	0.86	0.06[a]	0.86	0.86	0.86
Scores	0.38[a]	0.76	0.60	0.77	0.51	0.71
Ranking	6	2	4	1	5	3

[a]Evaluation measures less than 0.5, representing the metrics that are poor to satisfy the customer needs.

their relations are expressed with uncertainties. As the method can also identify the metrics that result in poor evaluation measures, therefore, it provides the guidelines for improving the design to better satisfy customer needs. The feedforward neural network is an efficient tool to achieve the fuzzy reasoning results. Therefore, the introduced method is a new approach for conceptual design. It can be used for developing the future computer-based design systems with conceptual design functions.

Acknowledgements

This research is jointly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through research grants: OGP 0185788 and OGP 0105754, and Atomic Energy Canada Ltd. through an Industrial Research Chair grant.

Appendix A. Back-propagation algorithm used to train the feedforward neural network

Step 1: *Initialize weights and thresholds*. Assign small random values to all the connections u_{ij} and w_{jk}, threshold values of all the hidden nodes Θ_j ($j = 1, 2, \dots, p$), and threshold values of all the output nodes Γ_k ($k = 1, 2, \dots, q$).

Step 2: *Present input and desired output data*. Provide the mth correct input data $I^{(m)} = (I_1^{(m)}, I_2^{(m)}, \dots, I_n^{(m)})$ and output data $O^{(m)} = (O_1^{(m)}, O_2^{(m)}, \dots, O_q^{(m)})$ to the feedforward neural network.

Step 3: *Calculate actual outputs*. Obtain the hidden node values and output node values using

$$H_j = f\left(\sum_{i=1}^{n} (I_i^{(m)} u_{ij}) - \Theta_j\right) \quad (j = 1, 2, \dots, p), \quad \text{(A.1)}$$

$$O_k = f\left(\sum_{j=1}^{p} (H_j w_{jk}) - \Gamma_k\right) \quad (k = 1, 2, \dots, q), \quad \text{(A.2)}$$

where, $f(\ )$ is the sigmoid function $f(x) = (1 + e^{-x})^{-1}$.

Step 4: *Adapt weights*. Computer the errors at the output nodes and hidden nodes using

$$d_k = d_k^{(m)} = O_k(1 - O_k)(O_k^{(m)} - O_k) \ (k = 1, 2, \dots q). \quad \text{(A.3)}$$

$$e_j = H_j(1 - H_j) \sum_{k=1}^{q} w_{jk} d_k \ (j = 1, 2, \dots, p). \quad \text{(A.4)}$$

Adjust the connection weights using

$$\Delta w_{jk} = \alpha H_j d_k \quad (j = 1, 2, \dots, p; k = 1, 2, \dots, q), \quad \text{(A.5)}$$

$$\Delta u_{ij} = \beta I_i^{(m)} e_j \quad (i = 1, 2, \dots, n; j = 1, 2, \dots, p), \quad \text{(A.6)}$$

where α and β are learning rates in the range of 0 to 1.

Step 5: *Calculate the total error*. The error for the mth correct data set is calculated by

$$E^{(m)} = \sqrt{\frac{\sum_{k=1}^{q} d_k^2}{q}}. \quad \text{(A.7)}$$

The total error considering all the M sets of correct data is obtained by

$$E = \sum_{m=1}^{M} E^{(m)}. \quad \text{(A.8)}$$

If E is less than a pre-defined small number ε, the training should be stopped. Otherwise, the training should go to Step 2.

References

[1] Singh N. Systems approach to computer-integrated design and manufacturing. New York: Wiley, 1995.
[2] Gu P. Recent development in design theory and methodology research. Keynote lecture, Proceedings of the International Conference on Manufacturing Sciences, June 10–12, Wuhan, China, 1998. p. 21–6.
[3] Pahl G, Beitz W. Engineering design: a systematic approach. Berlin: Springer, 1988.
[4] Suh N. The principles of design. Oxford: Oxford University Press, 1990.
[5] Akao Y. Quality function deployment: integrating customer requirements into product design. Cambridge, MA: Productivity Press, 1990.
[6] Kusiak A. Concurrent engineering: automation, tools, and techniques. New York: Wiley, 1993.
[7] Bralla JD, editor. Handbook of product design for manufacturing. New York: McGraw-Hill, 1986.
[8] Boothroyd G, Dewhurst P, Knight W. Product design for manufacture and assembly. New York: Marcel Dekker, 1994.
[9] Xue D. A multi-level optimization approach considering product realization process alternatives and parameters for improving manufacturability. J Manufacturing Systems 1997; 16(5):337–51.
[10] Andersen MM, Khler S, Lund T. Design for assembly. Bedford, UK: IFS Publications Ltd, 1983.
[11] Yan X, Gu P. Assembly/disassembly sequence planning for life-cycle cost estimate. Proceedings of the 1995 ASME Winter Annual Meeting, MED-Vol. 2-2/MH-Vol-3–2. 1995. p. 935–56.
[12] Gu P, Singh D. Product life-cycle serviceability analysis for supporting engineering design. I. activity based cost analysis of life-cycle services. Eng Design and Automation 1997;3(3): 255–74.
[13] Singh D, Gu P. Product life-cycle serviceability analysis for supporting engineering design. II. serviceability analysis for product redesign. Eng Design Automation 1997;3(3):275–97.
[14] Zhang HC, Kuo TC, Lu H, Huang SH. Environmentally conscious design and manufacturing: a state-of-the-art survey. J Manufacturing Systems 1997;16(5):352–71.
[15] Ohsuga S. Toward intelligent CAD systems. Computer-Aided Design 1989;21(5):315–37.
[16] Hashemian M, Gu P. Representation and retrieval of design knowledge for conceptual design. Proceedings of 3rd CIRP Workshop on Design and Implementation of Intelligent Manufacturing Systems, Tokyo, 1996. p. 85–94.

[17] Klir GJ, Clair St.U, Yuan B. Fuzzy set theory: foundations and applications. Englewood Cliffs, NJ: PTR Prentice Hall, 1997.
[18] Simpsom PK. Artificial neural systems: foundations, paradigms, applications, and implementations. Oxford: Pergamon Press, 1990.
[19] Du TCT, Wolfe PM. Implementation of fuzzy logic systems and neural networks in industry. Computers in Industry 1997;32:261–72.
[20] Monostori L, Barschdorff D. Artificial neural networks in intelligent manufacturing. Robotics and Computer-Integrated Manufacturing 1992;9(6):421–37.
[21] Huang SH, Zhang HC. Neural-expert hybrid approach for intelligent manufacturing: a survey. Computers in Industry 1995;26: 107–26.
[22] Xue D, Dong Z. Coding and clustering of design and manufacturing features for concurrent design. Computers in Industry 1997; 34:139–53.
[23] Venugopal V, Naredran TT. Neural network model for design retrieval in manufacturing systems. Computers in Industry 1992; 20(1):11–23.
[24] Kamarthi SV, Kumara SRT, Yu FTS, Ham I. Neural networks and their applications in computer design data retrieval. J Intelligent Manufacturing 1990;1(2):125–40.
[25] Bahrami A, Lynch M, Dagli CH. Intelligent design retrieval and packing system: application of neural networks in design and manufacturing. Int J Production Res 1995;33(2):405–26.
[26] Chang CA, Tsai CY. Using ART1 neural networks with destructive solid geometry for design retrieving systems. Computers in Industry 1997;34:27–41.
[27] Gu P, Hashemian M, Sosale S. An integrated design methodology for life cycle engineering. Ann CIRP 1997;46/1:71–4.
[28] Gu P, Hashemian M, Sosale S, Zhang Z. Module formation algorithms for product life cycle configuration design. Proceedings of the International Conference of Pacific Rim Management. Vancouver, August 14–16. 1998. p. 146–51.
[29] Zadeh LA. Fuzzy sets. Information and Control 1965;8:338–52.
[30] Saaty TL, Vargas LG. The analytic hierarchy process. New York: McGraw-Hill, 1980.
[31] Ulrich KT, Eppinger SD. Product design and development. New York: McGraw-Hill, 1995.