A Soft Case-based Reasoning System for Travelling Time Estimation

Lixing Wang
Department of Industrial and Systems Engineering
The Hong Kong Polytechnic University
Hong Kong
e-mail: lx.wang@polyu.edu.hk

Wai-Hung Ip
Department of Industrial and Systems Engineering
The Hong Kong Polytechnic University
Hong Kong
e-mail: mfwhip@inet.polyu.edu.hk

***Abstract*—Soft computing, which includes fuzzy logic, neural network theory, evolutionary computing and probabilistic techniques, is an emerging approach to computing. It resembles biological processes more closely than traditional techniques. Case-based Reasoning (CBR), which normally includes four different phases in the problem-solving cycle, is an effective methodology for solving these kinds of problems. Integration of soft computing techniques into a CBR system can significantly help people to solve problems with more accuracy. This study aims to develop a soft CBR System using fuzzy logic and neural network theory to estimate travelling time of vehicles. A simulated case is also studied to test the performance of the system. The results show that the soft CBR system is able to achieve accurate solutions.**

Keywords-Case-based Reasoning; Soft Computing; Fuzzy Logi;, Neural Network; Travelling Time.

I. INTRODUCTION

Vehicle travelling time is an important factor in logistics. Accurate prediction of travelling is very helpful both to goods delivery and to make vehicle schedules [4]. When vehicles are used to deliver consignments, it is necessary to give an accurate estimation of time they will take to reach the destination. It will be convenient to customers if the logistics company can supply an accurate arrival time. Customers can prepare to receive the consignments. Accurate transportation time estimation is also important for making schedules. Firstly, accurate estimation of time can help planner to know whether the consignment would arrive at the destination before the deadline. Furthermore, most optimization algorithms for vehicle routing problems are studied based on the information of vehicle travelling time. One challenge for the application of these optimization algorithms in industries is the accurate prediction of the arrival time of the consignments, since the travelling time is hard to predict for the logistics companies. The optimization results are also unfeasible if it is computed based on a inaccurate travelling time estimation. Consequently, logistics companies prefer making the schedules based on their experiences. Moreover, if some accidents which will affect the travelling time of vehicles happened, there is also a need to adjust the estimation of the travelling time.

This research aims to develop a soft CBR system to estimate the travelling time of vehicles. The system integrates soft computing techniques: fuzzy logic [11], neural network theory [5] and the concepts of CBR [1] together. With the help of the system, the travelling time can be estimated more accurate [5, 6]. This is because traditional CBR system can just forecast travelling time based on the cases in its database. But the conditions will change over time, such vehicle speed will increase, more vehicles will be on the road and so on. The traditional can just used outdated information to forecast, and it cannot achieve accurate results. The proposed system has the machine learning function. The system can update cases automatically. It can achieve more accurate results than traditional CBR system.

The rest of this paper is organized as follows: Section II contains a review of previous studies. In Section III, the system architecture is introduced. How integrate CBR and soft computing is also discussed in this section. In Section IV, a simulated case is computed to illustrate the system and to test the performance of the system. The final section draws conclusions from the research. The future work on this topic is also discussed.

II. LITERATURE REVIEW

Case-based Reasoning (CBR) is a designed model for expert systems [1]. It focuses on the reuse of experience [2]. Aamodt and Plaza [3] describe the traditional CBR model, which defines the problem-solving cycle in four different phases: Retrieval, Reuse, Revise and Return. Figure 1 illustrates the cycle.

The problem solving life cycle in a CBR system consists essentially of four parts:

1. RETRIEVE the most similar case or cases
2. REUSE the information and knowledge in that case to solve the problem
3. REVISE the proposed solution
4. RETAIN the parts of this experience likely to be useful for future problem solving

A new problem is solved by retrieving one or more previously experienced cases, reusing the case in one way or another, revising the solution based on reusing a previous case, and retaining the new experience by incorporating it into the existing knowledge-base (case-base).

An initial description of a problem (top of figure) is regarded as a new case. It is used to RETRIEVE a case from the collection of previous cases. The retrieved case is combined with the new case - through REUSE - into a solved case, i.e. a proposed solution to the initial problem. Through the REVISE process this solution is tested for success, either application to the real world environment or evaluation by an expert, and repaired if failed. During

 ISBN: 978-1-61208-154-0

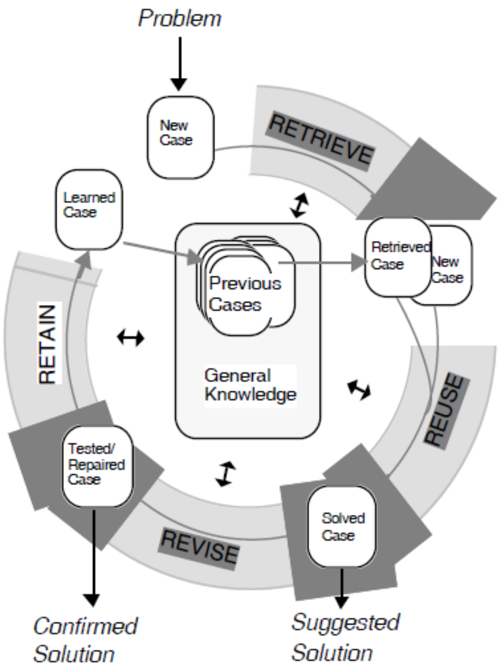

Figure 1. Case-based Reasoning Cycle (derived from [3])

RETAIN, useful experience is retained for future reuse, and the case base is updated by a new learned case, or by modification of some existing cases.

Sadek et al. [4] have successfully developed a prototype case-based reasoning system for real-time freeway traffic routing. CBR can solve new problems by reusing solutions of similar past problems. The result of their research has demonstrated that the prototype system can run in real-time and produce high quality solutions using case-bases of reasonable size. Shen et al. [5] proposed an approximate CBR model. This model uses the neural networks technology to process fuzzy inference with the two dualities of fuzzy logic and approximate reasoning. The self-organizing and self-learning procedure can be executed by modifying the weight. Maria and Maite [6] proposed retention and forgetting strategies to maintain the case base of the CBR system a certain scale by adding and removing cases. It has been proved in their research that the model they proposed can maintain the case base effectively. Anthony and Xun [7] successfully applied CBR system to handle planning applications in development control. The system helped the planner to reuse previous similar cases in making decision on the new applications. Castro et al. [8] developed a fuzzy system to improve CBR system in solving risk problems. Some rules are developed in the fuzzy system to search for the most suitable case not the most similar case. Passone et al. [9] incorporate domain-specific knowledge into a genetic algorithm to implement CBR adaptation. The research improved the adaptation phase in the CBR system. The improved system is suitable to deal with numerical modeling applications that require the substitution of a large number of parameter values.

These successful applications and relevant approaches encourage us to further research about CBR system for our problem: travelling time estimation. A soft CBR system has been developed. Fuzzy logic and neural network theory are applied in the system. In the next section, details about the system will be discussed.

III. METHODOLOGY

Figure 2 shows the system architecture used in this study. The system includes two parts. The first part separates the planned route of the vehicle into several segments. It helps users to find the same route from the case base. If there is not the same route, the most similar route will be selected using fuzzy logic and CBR. The other part of the system uses CBR to calculate the time of each segment of the planned route. The weightings of the CBR are trained using Neural Network theory. Finally, the case base is updated using some rule-based strategies.

A. Route Division

Firstly, route division part of the system will be introduced. When the user inputs the start location and the destination to the system, several routes will be generated. The user can choose one as the planned route. Then the system can help the user to estimate the time that needed for the vehicle from the start point to the destination. The first step is to divide the planned route of the vehicle into several segments using important traffic cross points. The Figure 3 shows the route segments of an example. If the vehicle is transported from point 1 to point 9, the planned route is 1-2-8-9. The system divides the route into three parts: 1-2, 2-8, 8-9. Then, it searches the case database for the same segments.

If the system cannot find the same route segment from the case base, the most similar route will be selected using fuzzy logic. Since road grade and city scale are two main factors to affect the vehicle speed, the fuzzy sets in this system include road grade (1, 2, 3 and 4), which indicates the designed vehicle speed on that road and city scale (super, big, middle and small), which indicates the population in that city. Figure 4 shows the membership functions. The sum of each similarity measure is the finally similarities of the route. The route with the biggest value is the most similar route.

There are two reasons for dividing a route into several segments. One is that the short route segments have relatively more similar routes than long route segments which can reduce the scale of the case base, so the search speed can also be increased. The other reason is that the number of same routes will be increased. Two vehicles may start from different places and also arrive at different places, but part of their routes may be the same.

B. CBR System

CBR sub-system is the core of the soft CBR system. The module is mainly designed using CBR. The weight coefficients of all factors which affect the process of degeneration are saved in a database. The design of the database is shown in Table I.

Different kinds of data have different weightings (w_i). The weightings need to satisfy the constraints:

$$\sum_{i=1}^{n} w_i = 1 \tag{1}$$

where, $0 \le w_i \le 1$ (i = 1, 2… …n)

When the vehicles finish their transportation tasks, they will give the information listed in the table to the backend system. The system will store the data in the case database and produce a new case_id for it. The case database is shown in Table II.

When the system begins to estimate the travelling time, it firstly searches for the case base. If one kind of data of the vehicle matches the same kind of data of a case, the system will record 1 as the value of x_i in the blank space of match_degree. Then, it multiplies the weighting of this kind of data by the match_degree ($w_i \times x_i$) to produce the result. After calculating all the types of data belonging to the case, the system adds all the results together. The sum (M) is the degree to which the case matches the problem that needs to be solved.

$$x_i = \begin{cases} 0, & \text{not match,} \\ 1, & \text{match,} \end{cases} \tag{2}$$

i = 1, 2……n,

$$M = \sum_{i=1}^{n} w_i x_i \tag{3}$$

where, M indicates the degree to which the case matches the problem.

After calculating all the cases of the same route, the system chooses the best match in the database and then uses the time consumed on this case as the predicted result of this segment. All the time needed by each segment added together is the time for the vehicle to arrive at the destination.

On the other hand, if all the degree to which the case matches the problem is less than 0.7 (the value is a coefficient and users can adjust it based on real conditions), the system will compute the deviation of the most existing similar route segments under the condition of the best match case and the

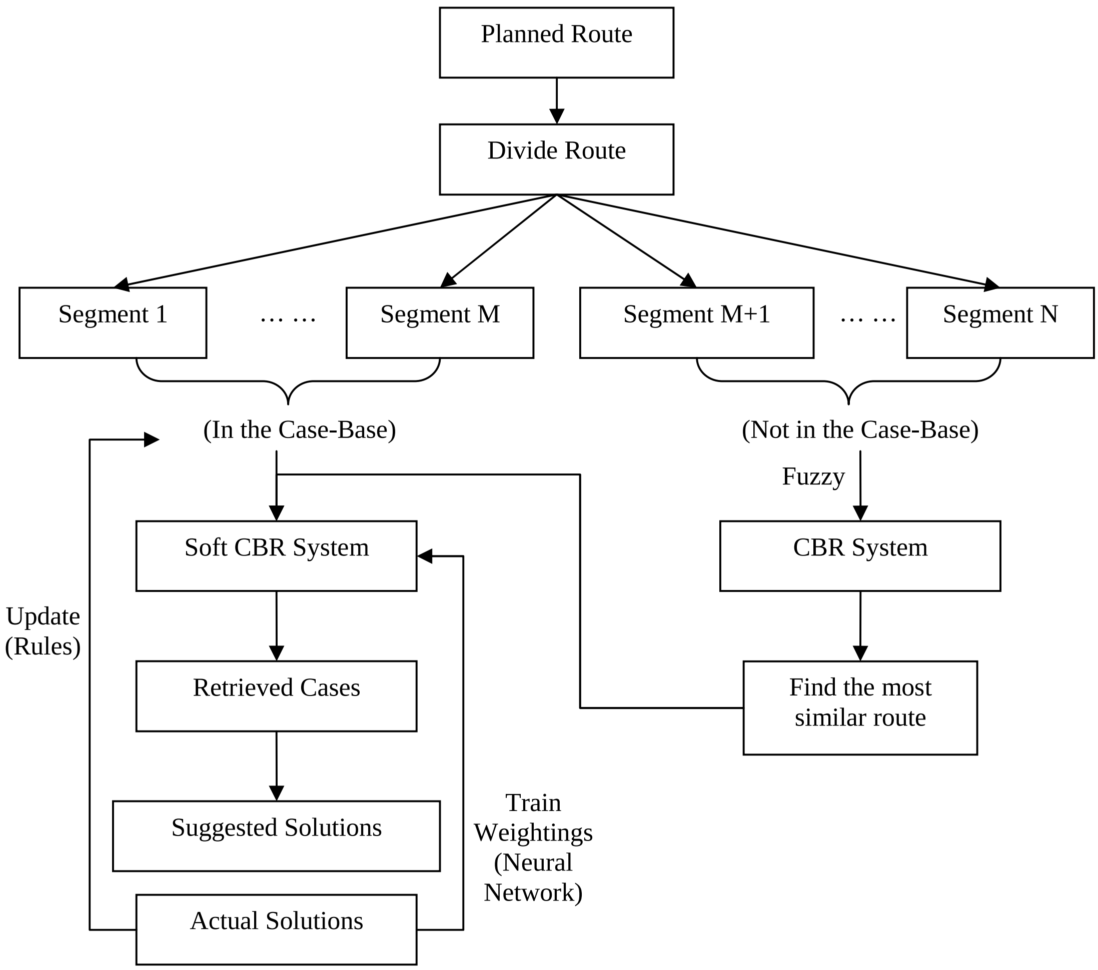

Figure 2. System architecture

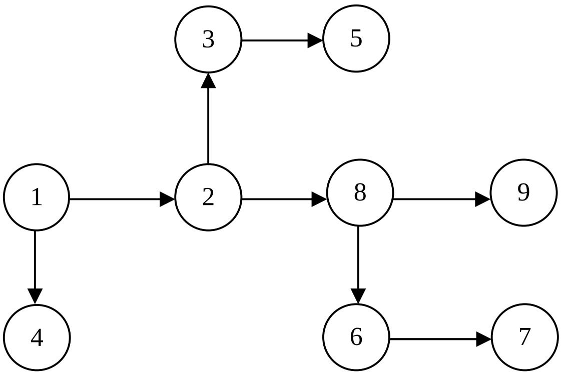

Figure 3. An example of route separation

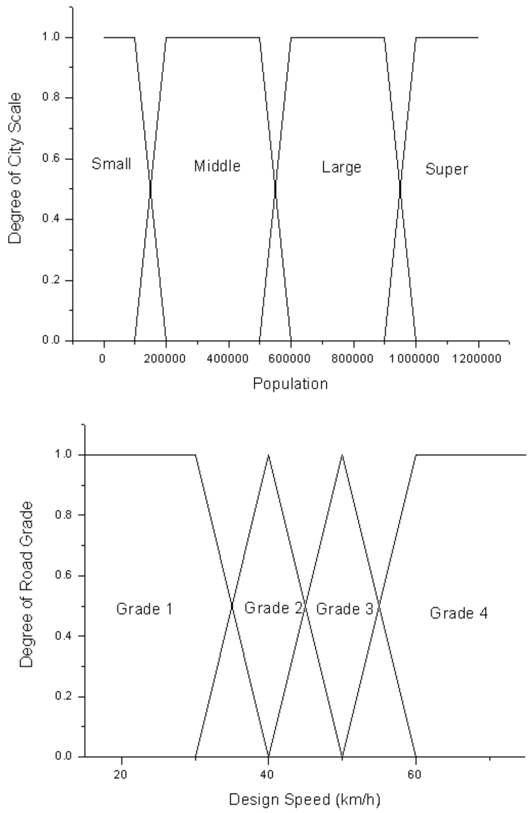

Figure 4. Membership Functions

condition of the unsolved problem. The deviation ratio will be set as the deviation ration of this unsolved problem and the best match case. Then the travelling time can be computed.

Finally, the total time spend for each segment is the time needed for consignments delivery.

In the CBR system, w_i is used to represent the weighting of each factor in a case for calculating the travelling time. Sometimes, the weightings in the database are not accurate. They are need to be adjusted. Which situations the system needs to adjust will be introduced in Part C of this section. And in this part, the adjustment methodology will be discussed firstly. Neuron network is applied to train these weightings. The details will be introduced as follows.

Let $X = \{x_1, x_2, ..., x_n\}$ be a set of n vectors, where the components of each vector represents the match degree of Case j . w_i is the coefficient. The value is decided by specific segments. Different segments has different set of w_i . A simple single layer neural network is used to train their values [10].

Step 1: Initialization

Set initial weights w_i and threshold θ to random numbers.

Step 2: Activation

Activate the perceptron by applying inputs $x_i(p)$ and desired output $Y_d(p)$, which means the actual travelling time. Calculate the actual output at iteration p=1

$$Y(p) = step[\sum_{i=1}^{n} x_i(p) w_i(p) - \theta] \qquad (4)$$

where n is the number of the perceptron inputs, and formula (4) is a step activation function.

Step 3: Weight training

Update the weights of the perceptron

$$w_i(p+1) = w_i(p) + \Delta w_i(p) \qquad (5)$$

where $\Delta w_i(p)$ is the weight correction at iteration p.

The weight correction is computed by the delta rule:

$$\Delta w_i(p) = \alpha \times x_i(p) \times e(p) \qquad (6)$$

$$e(p) = Y_d(p) - Y(p) \qquad (7)$$

Step 4: Iteration

Increase iteration p by one, go back to step 2 and repeat the process until convergence.

Then w_i can be determined.

TABLE I. THE WEIGHTING COEFFICIENTS DATABASE

Weighting (w_i)	**Factor**	**Match_Degree** (x_i)
w_1	*Weather*	1/0
w_2	*Workday/Holiday*	1/0
w_3	*Time_Period*	1/0
w_4	*Vehicle_Type*	1/0
w_5	*Driver*	1/0
w_6	*Products Weight*	1/0
……	……	1/0
w_n	*Fuel_level*	1/0
	Sum	$\sum w_i x_i$

TABLE II. THE CASE DATABASE FOR MONITORED CONTAINERS

Case_ID	Time_Consumed (min)	Match_Degree
A01A0201	36	0.5
A02B0301	44	0.7
B03B0501	15	0.9
A02C0801	24	0.3
C08C0901	17	0.5
C08C0601	32	0.7

C. Updating Case Base

Everything is in course of continuous movement change, and development in the world. With the developing of vehicles and cities, the travelling time needed from the same start location and destination under the same condition will be different. The old cases in the database will be unsuitable for new arriving cases. Consequently, it is necessary to update the old case base, when the value of deviation is too big. The case base is updated using the rules listed below. Firstly, the system will check whether the segment calculated is the same route with the unsolved problem. If it is the most similar route, the actual segment and its result will be set as a new case in the case base. For the same route segment, if the new case can find a previous case from the case base, which $M = 1$, the old case will be replaced by the new case. If the results are different, the weighting of the case will be trained again. If $M \neq 1$, the case will also be added into the case base.

Rule 1: If the segment is the same as the actual segment and the match degree is not less than 0.7, then go to Rule 3;

Rule 2: If the segment is the same as the actual segment and the match degree is less than 0.7, then train the weighting and add the case into the case base.

Rule 3: If $M = 1$, then go to Rule 5

Rule 4: If $M \neq 1$, then train the weighting using neural network and add the case into the case base.

Rule 5: If (Result_New-Result_Old) / Resul_Old >5% or (Result_New-Result_Old) / Resul_Old < -5%, then train the weighting of this segment and add the case into the case base.

IV. CASE STUDY

In this section, a case study is used to illustrate how to apply this system to calculate the travelling time. Since it is difficult to collect adequate data from real transportation system, a simulated case is studied to describe the operating procedure of the system.

YT is a simulated logistics company in mainland China. The company plans to transport consignments from Shanghai to Beijing. A suggested route can be generated by the system based on Google map. The routes include: 311 Guangfu Road in Shanghai to G2 Highway, G2 Highway, S29 Highway, G25 Highway, G18 Highway, G3 Highway, S30 Highway, Jingjing Highway, Jingjing Highway to 14 Hepingli in Beijing. All the route segments can be found in the case base except the first one and the last one. The first one, the route from 311 Guangfu Road in Shanghai to G2 Highway, will be used to illustrate the operation of the Fuzzy part of the system. In contrast, another route G2 Highway, which can be found from the case base, will be used to illustrate the Neural Network part of the system as an example.

TABLE III. CASE INFORMATION

Weather	Sunny
Workday/Holiday	Workday
Time_Period	8:45
Vehicle_Type	Truck_JF_15T
Driver	00012
Products Weight	15T

TABLE IV. CASE INFORMATION

Weighting (w_i)	Case_2314	Match_Degree (x_i)
0.20	*Weather*	1
0.30	*Workday/Holiday*	1
0.30	*Time_Period*	1
0.05	*Vehicle_Type*	1
0.05	*Driver*	1
0.10	*Products Weight*	1
	Sum	1

How the system calculates the time spends in the first route segment will be introduced. The fuzzy sets of this case are city scale and road grade. The city scale is measured by the population, which is more than 10 million. And the designed road grade is grade 2. Through computing by fuzzy logic, the most similar route segment is found. Then the system sets the most similar route segment as the unsolved new case and input it into the CBR system. Considering weather, driver and other factors listed in table I, the most similar case can be found. Then, formula (8) can be used to estimate the travelling time of this route segment. The method is the same as the calculation of the travelling time for the segment: G2 Highway.

$$t = t_s \cdot \frac{L}{L_s} \tag{8}$$

where, t_s is travelling time of the similar case, L_s is the route length of the similar case. t is the travelling time of the unsolved route segment. L is the route length of the unsolved route segment.

For the route segment that can be found in the case base, G2 Highway is used to illustrate. The G2 Highway in this case is shown in the table III. The most similar case that $M = 1$, Table IV shows the details. Since the $M = 1$, the travelling time of this similar case is regarded as the estimated time of this route segment, When all the segments are calculated, the time added together will be the total estimated time. After simulation, the division of real travelling time and the estimated result is bigger than 5%. Consequently, the weightings of this case need to be adjusted. Neural network is applied. After adjustment, the case base is updated using the new result of the G2 Highway.

V. Conclusion and future work

The main contribution of the study is to propose a soft case-based reasoning system to estimate travelling time of vehicles. The system is developed based on the traditional CBR system. It integrates Fuzzy logic and neural network techniques. Consequently, the system has the machine learning function. It can update itself automatically. The system can firstly divide the planned route into several segments. Then it helps users to search for the most similar case with each segment. If the deviation of the estimated time and real travelling time is huge, the weightings will be trained using neural network and old case will also be replaced. The system combines soft computing and case-based reasoning techniques to estimate travelling time. With the help of the system, the arrival time can be predicted more accurately. A simulated case is used to test the system. The results showed the system can be applied to estimate the travelling time. The accuracy increased with the help of neural network and fuzzy logic after some training. However, it is difficult to collect adequate data from real transportation system. It is hoped the system will be applied in the real transportation system in the future, and then its performance can be evaluated.

Acknowledgment

The authors wish to thank the Research Committee of The Hong Kong Polytechnic University for support in this research work.

References

[1] I. Watson, “Applying Case-Based Reasoning”, Morgan Kaufmann, San Francisco, CA, 1997.

[2] D. W. Aha, “The omnipresence of case-based reasoning in science and application”, Knowledge-Based Systems, vol. 11, pp. 261–273, 1998.

[3] A. Aamodt, and E. Plaza, “Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches”, AI Communications vol. 7, pp. 39-59, 1994

[4] A. W. Sadek, B. L. Smith, and M. J. Demetsky, “A prototype case-based reasoning system for real-time freeway traffic routing”, Transportation Research Part C, vol. 9 pp. 353-380, 2001.

[5] Z. Shen, H.C. Lui, and L. Ding, “Approximate Case-Based Reasoning on Neural Networks”, International Journal of Approximate Reasoning, vol. 10, pp. 75-98, 1994

[6] S. Maria, and L.S. Maite, “Adaptive case-based reasoning using retention and forgetting strategies”, Knowledge-Based Systems vol. 24, pp. 230-247, 2011

[7] A.G.O. Yeh, and X. Shi, “Case-based reasoning (CBR) in development control”, JAG, vol. 3(3), pp. 238-251, 2001

[8] J.L. Castro, M. Navarro, J.M. Sanchez, and J.M. Zurita, “Introducing attribute risk for retrieval in case-based reasoning”, Knowledge-Based Systems, vol. 24, pp. 257-268, 2011.

[9] S. Passone, P.W.H. Chung, and V. Nassehi, “Incorporating domain-specific knowledge into a genetic algorithm to implement case-based reasoning adaptation”, Knowledge-based Systems, vol. 19, pp. 192-201, 2006.

[10] M. Negnevitsky, Artificial Intelligence: A Guide to intelligent systems (Second Edition) (Ch. 6 168-174),2005

[11] J. Ma, S. Chen and Y. Xu, “*Fuzzy logic from the viewpoint of machine intelligence*”, Fuzzy sets and systems, vol. 157, pp. 628-634, 2006.