Evolving Fuzzy Neural Network Based Fire Planning in Rescue Firebrigade Agents

Behrouz Shahgholi Ghahfarokhi, Hamed Shahbazi, Mohammad Kazemifard, Kamran Zamanifar
Department of Computer Engineering, University of Isfahan, HezarJarib Street, Isfahan, Iran
shahgholi,rescue_esfahan, kazemifard, zamanifar@eng.ui.ac.ir

Keywords: RoboCupRescue, Firebrigade agent, Fire Selection, Evolving Fuzzy Neural Networks.

Abstract

RoboCupRescue Simulation System is a platform for designing and implementing various artificial intelligent issues. In rescue simulation environments, Firebrigades should select fire points in a collaborative manner such that the total achieved result is optimized. In this work, we are going to propose a new method for fire prediction and selection in Firebrigade agents. This method is based on Evolving Fuzzy Neural Networks to obtain a set of trained fuzzy rules as rule base of Firebrigades Fire Selection System to select targets autonomously.

INTRODUCTION

Disaster rescue is one of the most serious social issues which involve very large numbers of heterogeneous agents in the hostile environment. The intention of the RoboCupRescue project is to promote research and development in this socially significant domain at various levels involving multi-agent team work coordination, physical robotic agents for search and rescue, information infrastructures, personal digital assistants, a standard simulator and decision support systems, evaluation benchmarks for rescue strategies and robotic systems that are all integrated into a comprehensive systems in future. A generic urban disaster simulation environment is constructed on network computers. Heterogeneous intelligent agents such as Firebrigades, police forces, ambulances, victims, etc. conduct search and rescue activities in this virtual disaster world.

This problem introduces researchers advanced and interdisciplinary research themes. As AI/robotics research, for example, behavior strategy (e.g. multi-agent planning, real-time/anytime planning, heterogeneity of agents, robust planning, mixed-initiative planning) is a challenging problem. For disaster researchers, RoboCupRescue works as a standard basis in order to develop practical comprehensive simulators adding necessary disaster modules. A diverse spectrum of possibilities of this technology will contribute to the creation of the safer social system in the future.

Fire Selection Policy

As discussed above, Firebrigade agents are one of those saviors that are responsible for extinguishing fires in disaster space. Discussing about Firebrigades, there are a lot of limitations including limited vision and operation radius and dangers. Therefore, the knowledge of Firebrigades about fiery buildings will be limited to that radius and agents should earn their knowledge via communication and prediction. Also the rate of fire extinguishing is so much slower than the growing rate of fire and the amount of water is limited too. Each building in this simulation environment is described as an object with some attributes. The most important attributes of a building are: area of the building, area of its neighbors, kind of its material (i.e. wood), distance of that building to the refuge and fieriness which is the quantity of fire's destroying effect on that [5].

Considering these limitations, each Firebrigade in this simulation environment needs to have a fire selection policy that is able to predict the future of fieriness in buildings and choose the best ones as fast as possible. In [9] a decision tree based method has been proposed for fire selection in Robocup rescue environment. In the next subsection, we will describe Evolving Fuzzy Neural Networks as an emerging tool for the proposed fire selection policy.

Evolving Fuzzy Neural Network

EFNNs[1] are based on ECOS[2] frameworks for building on-line, adaptive intelligent systems that have

1. Evolving Fuzzy Neural Networks
2. Evaluating Connectionist Systems

both their structure and functionality evolving in time. The model is called evolving because of the nature of the structural growth and structural adaptation of the whole evolving connectionist system which it is a part of that. EFNNs evolve their structure and parameter values through incremental, hybrid supervised/unsupervised, on-line learning. New connections and new neurons are created during the operation of the system. EFNNs can accommodate new input data, including new features, new classes, and etc. through local element tuning. These characteristics are useful in cases where the number of features or classes is not determined.

EFNNs suggest a new neuro-fuzzy systemic approach that employs more sophisticated supervised/unsupervised, knowledge-based learning methods. The functionality of EFNNs can be fully utilized when EFNNs are used as elements of an ECOS framework for adaptive, intelligent, knowledge-based systems [4].

The reports in [7] exhibit that the EFNN model not only gives the analogous performance compared with other advanced and complex neuro-fuzzy systems, but also provides the feature of the expeditious one pass parameter training which makes it highly suitable for the low power requirement.

In the next section we will propose our fire selection scheme based on fuzzy systems and afterwards, we will utilize EFNN for fire planning.

A FUZZY FIRE SELECTION SCHEME

The fuzzy logic has been used to solve many decision making problems. Because of intricacy of fire prediction and selection in Rescue Firebrigade agents, we are going to implement a novel fuzzy decision making system for fire selection in Firebrigades. The proposed fuzzy decision maker is encompassed in the dotted frame as shown in Fig. 1. The basic functions of the components employed in the scheme are described as follows:

Fuzzifier. The fuzzifier performs the fuzzification function that converts eight types of input data into suitable linguistic values which are needed in the inference engine. Notably, the input to the fuzzifier *DT* represents freshness, which is a measure of novelty of our knowledge about fire point, *F* denotes the fieriness degree, *TA* stands for the total area of burning building divided by maximum building area in the map and BC represents burning rate of the building. *FCP* stands for Firebrigade Collectivity Percent and will be acquired using Eq. 1:

$$FCP = \frac{|NC|}{M} \tag{1}$$

Where $|NC|$ is the number of Firebrigades in the nearest cluster set to a building and M is total number of Firebrigades. *RA* (Refuge Adjacency) is another input which is a criterion of distance from fire point to nearest and farthest refuges. *RA* obtains as follows:

$$RA = \frac{Dist_{\min}(\text{fire}-\text{point}, \text{refuges})}{Dist_{\max}(\text{fire}-\text{point}, \text{refuges})} \tag{2}$$

Another input is *FPP* which stands for the Fired Perimeter Percent and will be acquired using Eq. 2:

$$FPP = \frac{\textit{firy primeterof building}}{\textit{total premiterof building}} \tag{3}$$

Fuzzy rule base. The fuzzy rule base is composed of a set of linguistic control rules and the attendant control goals.

Inference engine. The inference engine simulates human decision-making based on the fuzzy control rules and the related input linguistic parameters. The max–min inference method is used to associate the outputs of the inferential rules [1, 2], as described later in this subsection.

Defuzzifier. The defuzzifier acquires the aggregated linguistic values from the inferred fuzzy control action and generates a non-fuzzy control output C, which represents that how good is the current fire point. The Mamdani defuzzification method is employed to compute the centroid of membership function for the aggregated output, where the area under the graph of membership function for the aggregated output is divided into two equal sub-areas [1, 2].

In our proposed fire selection system, properties of fiery buildings and their neighbors are used as inputs to fuzzy inference parts as indicated by dotted frame in Figure 1. Afterwards, agents will select the bugling with maximum criteria as the next fire point to be distinguished.

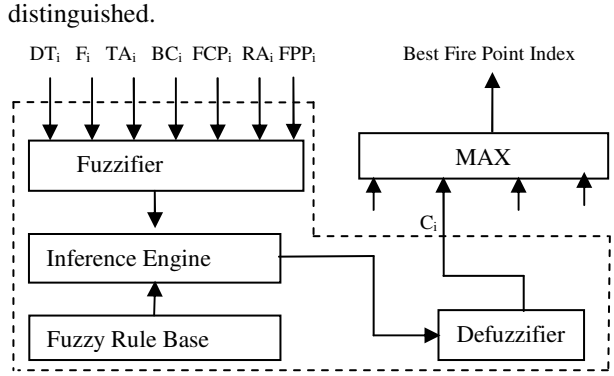

Figure 1. Fuzzy Fire Selection System

 ISBN: 1-56555-307-1

Fuzzy Membership Functions

After introducing the fuzzy predictor, we are going to determine membership functions. Some of fuzzy variables including BC, DT and also F are discrete variables which their membership functions are presented in Table 1.

Table 1. Fuzzy membership functions for discrete fuzzy values

DT		F		BC	
Low	(0,1) (1,1) (2,0.5) (others,0)	Not-Burning	(0,1) (1,0.1) (others,0)	Slow	(2,1) (0,0.3) (1,0.3)
Medium	(2,0.5) (3,1) (4,1) (5,0.5) (6,0)	Burning	(0,0) (1,0.5) (2,1) (3,0.5) (others,0.1)	Fast	(0,1) (1,1) (2,0.3)
High	(5,0.5) (6,1) (others,1)	Burned	(3,0.2) (5,0.5) (6,0.8) (7,1)		

Figure 2 shows mapping of other continious fuzzy input and output variables into some appropriate linguistic terms and membership values, which are expressed by the values within the range of 0 and 1.

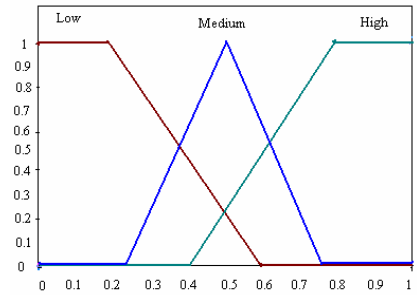

Figure 2. Fuzzy membership functions for normalized TA, RA, FCP and FPP variables

Fuzzy Inference and Inferential Rules

The input and output fuzzy sets are correlated to construct the inferential rules of the fuzzy fire selector. A part of these inferential rules have been listed in Table 2, which are correspondent with the manual observation.

As mentioned earlier, max-min inference method is used to associate the outputs of the inferential rules and Mamdani defuzzification method is employed for defuzzification phase. Fig. 3 illustrates the reasoning procedure for a two-rule Mamdani fuzzy inference system.

Although we generated a near exhaustive fuzzy rule base for derivation of fire selection criteria, our proposed rule base will be imperfect. In the next section, we are going to enhance the adaptation of our algorithm to a dynamically changing environment like rescue simulation environment using EFNN.

Table 2. The Inferential Rules

Rule	Inputs							Output
	DT	F	TA	BC	FCP	RA	FPP	C
1	Low	N-Burning	Low	Slow	Low	Low	Low	Low
2	Low	Burning	Low	Slow	Low	Low	Med	High
3	High	Burned	Low	Slow	Low	High	Low	Low
...								

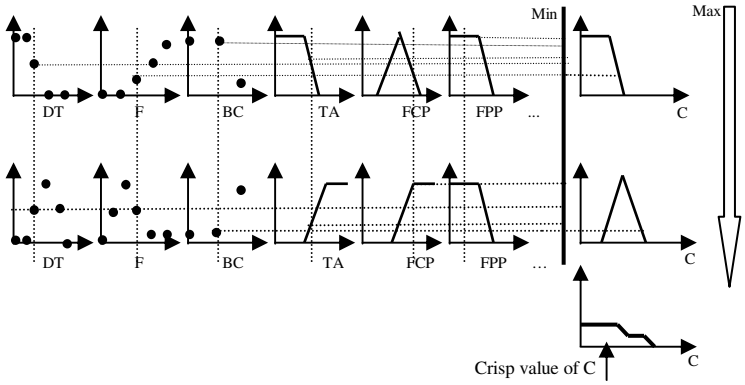

Figure 3. The reasoning procedure for Mamdani defuzzification method

FIRE SELECTION USING EVOLVING FUZZY NEURAL NETWORKS

The EFNN employs a feed forward neural network to process fuzzified data and defuzzifies the fuzzy data as the output. The proposed EFNN is mainly composed of five layers of neurons as shown in Figure 4. The fuzzy input layer carries out fuzzy quantization of the input variables, which are fed from the input layer. The rule layer contains rule nodes that can evolve through learning. Notably, these rule nodes reveal the prototypes of the input–output data mapping that can be graphically represented as associations of hyper-spheres between the fuzzy input and the fuzzy output spaces. Each node of the fuzzy output layer represents fuzzy quantization of the output variables, and then the defuzzification for the fuzzy output variables get done in the output layer.

Generally, an EFNN consists of five processing stages, which are network initiation, inputs feed forward, parameters tuning, node aggregation and pruning, and rule extraction, respectively. After each input vector is fed into the EFNN, the EFNN updates the parameters, evolves connections, aggregates and prunes nodes based on the output error during the last epoch if necessary. Then the EFNN propagates the signals forward, and computes the output error again[4].

Network Initiation

At the network initialization phase, the connection weights W_1 and W_2 as given in Fig. 4 are set to some predefined values based on the past experience of the network. W_1 represents the coordinates of the sphere center in the fuzzy input space, and W_2 the coordinates in the fuzzy output space. Triangular membership functions are used in the EFNN to simplify the computation of the similarity between the fuzzy input vector and the stored prototypes. Moreover, the number of the rule nodes and the connection patterns of W_1 and W_2 are settled beforehand based on the rule base given in Table 2.

Input Feed-Forward

When a new sample is fed as input, it is first fuzzified at the fuzzy input layer, and then a so called fuzzy distance between the output from the fuzzy input layer and the connections weights W_1 are calculated to determine if the input falls into the input receptive field of some specific rule node. The fuzzy distance between the two fuzzy membership vectors of input X_f and the connection weight of the j*th* rule node, $W_{1,j}$ is defined as follows:

$$FD(Xf, W1, j) = \frac{\left\| X_f - W_{1,j} \right\|}{\left\| X_f + W_{1,j} \right\|} \qquad (4)$$

Where $\left\| x_f - w_{1,j} \right\|$ denotes the sum of all the absolute values of a vector that is obtained after subtraction of X_f and $W_{1,j}$ vectors and $\left\| x_f + w_{1,j} \right\|$ is the sum obtained after summation of X_f and $W_{1,j}$ vectors. Here, $W_1 = [W_{1,1} \;\; W_{1,2} \;\; ... \; W_{1,N}]$, and N is the number of the rule nodes.

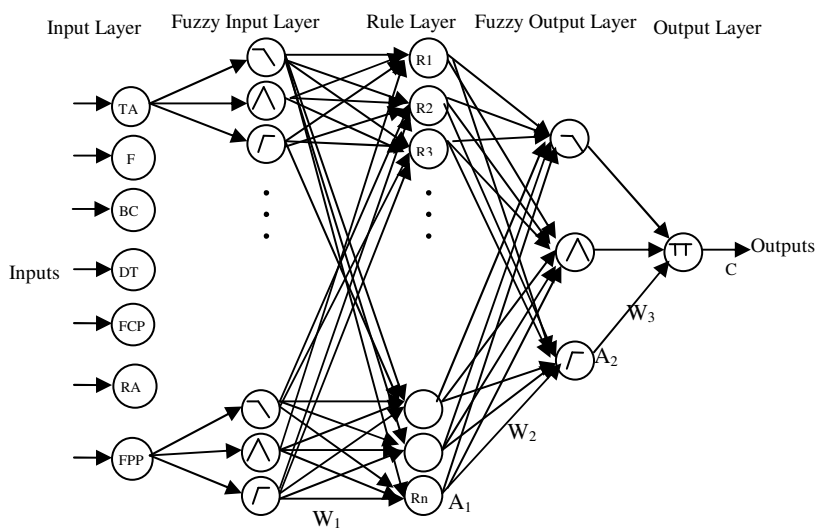

Figure 4. Architecture for EFNN-based Fire Selection System

 ISBN: 1-56555-307-1

We then select the rule node with the highest activation means the rule j with the lowest FD value. Activation set for rule layer is $A_1 = [A_{1,1} \ A_{1,2} \ ... \ A_{1,N}]^T$ where:

$$A_{1,j} = 1 - FD(X_f, W_{1,j}) \quad (5)$$

If the activation of the selected rule node is smaller than a predetermined sensitivity threshold, a new rule node is created and new connection weights are established for the new fuzzy input X_f and related fuzzy output pair, Y_f. So, new network has two added weights $W_{1,(N+1)}$ and $W_{2,(N+1)}$ where :

$$\begin{aligned} W_{1,(N+1)} &= X_f \\ W_{2,(N+1)} &= Y_f \end{aligned} \quad (6)$$

On the other hand, when the activation of the selected rule node is larger than the sensitivity threshold, it is passed forward to the next layer to compute the output of Fuzzy Output Layer A_2 as follows:

$$A_2 = satlin\ (W_2 \bullet A_1) \quad (7)$$

Where $W_2 = [W_{2,1} \ W_{2,2} \ ... \ W_{2,N}]$ and satlin() represents the saturating linear transfer function. Similarly, a new rule node will be created if the following fuzzy output error is larger than a predefined threshold value,

$$FE_{out} = \| A_2 - y_f \| \quad (8)$$

At last, crisp output value Y_c can be derived by Eq. 8 where W_3 denotes the connection weights between the fuzzy output layer and the output layer,

$$Y_c = W_3 \bullet A_2 \quad (9)$$

Parameter Tuning

The training process of the network involves the updating of the connection weights W1 and W2, the learning rate and the sensitivity thresholds for each rule node. W1 is adjusted via unsupervised learning based on the similarity between the fuzzy input vector X_f and the stored prototypes W1,j for the jth rule node as follows,

$$W_{1,j}(t+1) = W_{1,j}(t) + \eta_j\ (W_{1,j} - X_f) \quad (10)$$

And W2 is updated according to the Widrow–Hoff least mean square (LMS) algorithm [3] that minimizes the fuzzy output error,

$$W_{2,j}(t+1) = W_{2,j}(t) + \eta_j\ (A_2 - Y_f).A_{1,j} \quad (11)$$

In both equations, η_j stands for the learning rate of the j*th* rule node. Note that η_j can be expressed as $\eta_j = 1/ACC_j$ where ACC_j is the accumulated number of accommodated examples for the jth rule node. The sensitivity threshold for the rule node, which has referred in Eqs. (5), (6) or Eq. (7), is given by

$$S_j\ (t+1) = S_j(t) + FD\ (W_{1,j}(t+1), W_{1,j}(t)) \quad (12)$$

The training of the above parameters will be reinitiated whenever the ratio of the cumulative routing-miss to n successive incoming routing requests is larger than a predefined threshold, where n denotes the number of the training patterns for this network.

Rule Node Aggregation

After certain number of training samples has been presented, some neurons and connections may be pruned or aggregated. If the fuzzy distance as given in Eq. (3) for every two out of K nodes is less than some threshold for both connections W1 and W2, the K nodes can be aggregated into one single rule node with the following connection weights and sensitivity threshold,

$$\begin{aligned} W_{r,agg} &= \frac{\sum_{i=1}^{K} W_{r,i}}{K}, r = 1,2 \\ S_{agg} &= 1 - \underset{i \in 1..K}{Max}(\ FD(W_{1,agg}, W_{1,i})\ ) \end{aligned} \quad (13)$$

Where $W_{1,\ agg}$, $W_{2,\ agg}$ and S_{agg} represent the connections and sensitivity threshold for the aggregated node.

Rule Extraction and Insertion

Every rule node in the network can generate a fuzzy rule from W_1 and W_2 connections. We assume that there exists a fuzzy rule in the network like Figure 5.

Rj : IF DT is ‘‘low’’ with a degree of 0.45, and is ‘‘medium’’ with a degree of 0.65,
AND F is ‘‘Burning’’ with a degree of 0.67, and is ‘‘Burned’’ with a degree of 0.39,
AND BC is ‘‘Slow’’ with a degree of 0.44, and is ‘‘Fast’’ with a degree of 0.68,
AND TA is ‘‘low’’ with a degree of 0.14, and is ‘‘medium’’ with a degree of 0.68,
AND FCP is ‘‘medium’’ with a degree of 0.64, and is ‘‘high’’ with a degree of 0.18,
AND FPP is ‘‘low’’ with a degree of 0.14, and is ‘‘high’’ with a degree of 0.68, …
THEN C is ‘‘low’’ with a degree of 0.35, and is ‘‘medium’’ with a degree of 0.88

Figure 5. A fuzzy rule extracted from EFNN network

In Figure 5 the number after each fuzzy label represents the degree to which the centers of the input and the output hyper-sphere belong to the respective membership function. The degrees associated with the premise and the consequent parts of the rule are the connections weights from W1 and W2, respectively.

For insertion of fuzzy rules (as are in table 1), a new rule node, r_i, will be inserted in rule layer such that the connection weights W1(rj) and W2(rj) of the rule node represent this rule. It means that only for corresponding linguistic terms in rule, values of W_1(rj) and W_2(rj) are one and for others are zero [4].

SIMULATION RESULTS

Studying the proposed method, a set of simulations has been accomplished on different maps in RCRSS[3]. The simulation results have been compared using decision tree based fire selection [9], our classic fuzzy fire selection scheme and EFNN-based fire selection scheme. EFNN-based method has been trained on different maps in a set of iterations. Acquiring training samples, each agent will apply one training sample including input/output pair to the EFNN every 10 cycles. These training sample pair is obtained from selecting one fiery building using current EFNN-based fire selection scheme and computing local criteria after 10 cycles. The local criterion is the proportion of fiery area around the selected building. These simulations are acquired withdrawing the effects of other agent types such as Police Forces in multi-agent system. RCRSS score has been chosen for comparing efficiency of above methods which are shown in table 3. These values are the averaged scores among a number of iterations per each disaster space map.

3 .RoboCup Rescue Simulation System

Table 3. Comparing different fire selection methods on VC, Kobe and Foligno maps.

Fire Selection Methods	Dtree-Based Fire Selection	Fuzzy Fire Selection	EFNN-Based Fire Selection
Kobe	72.38	69.35	78.03
VC	46.56	45.89	56.69
Foligno	25.38	24.12	32.56

Simulation results illustrate that the EFNN-based method is more effective in fire selection than fuzzy fire selection method and previous Dtree-based method. The fastness of EFNN-base method will become slower as the number of rules increase. Nevertheless, the above results show that EFNN is appropriate when the number of trained rules is rational.

CONCLUSIONS AND FUTURE WORKS

In this paper, two methods have been presented for fire selection problem in rescue simulation environment which are preliminary fuzzy-based scheme and EFNN-based fire selection method. Both of these methods are used for predicting the future of fire points and selecting the best fire points in a multi-agent environment. Simulation results show that EFNN is more efficient disrespecting the number of rules. In this method, the effect of other types of agent is not considered and so we are going to improve the proposed method by considering the interaction between other types of agents.

REFERENCES

[1] Pedrycz, W., Gomide, F. 1998. *An Introduction to Fuzzy Sets: Analysis and Design (complex adaptive systems).* MIT Press, Cambridge, MA.

[2] Buckley, J. J., Eslami, E., Esfandiar, E. 2002." An Introduction to Fuzzy Logic and Fuzzy Sets (advances in soft computing)". Physica Verlag .

[3] Principe J., Euliano, N., Lefebvre, W.2000." Neural and Adaptive Systems: fundamentals through simulations". John Wiley & Sons, Inc., New York.

[4] Kasabov, N. 2001." Evolving Fuzzy Neural Networks for Supervised/ Unsupervised On-line Knowledge-based Learning". IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, vol. 31, no. 6 , 902–918.

[5]Takeshi Morimoto. 2002. How to Develop a RoboCupRescue Agent for RoboCupRescue Simulation System version 0. 1st edition .

[6] Woodford, B. J., Kasabov, N. K.2001." Ensembles of EFuNNs: An Architecture for a Multimodule Classifier". The 10th IEEE International Conference on Fuzzy Systems, Vol. 3 , 1573–1576.

[7] Abraham, A.2001. Neural Fuzzy Systems: state-of-the-art Modeling Techniques , In: Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence. Jose Mira and Alberto Prieto (Eds.), Springer-Verlag, Germany, ISBN 3-540-42235- 8, Granada, Spain. 269–276.

[8] Futschik, Matthias, E., Reeve, A., Kasabov, N. 2003. "Evolving Connectionist Systems for Knowledge Discovery From Gene Expression Data of Cancer Tissue". Artificial Intelligence in Medicine, Vol. 28, No. 2. 165–189.

[9] Shahbazi, H., Zafarani, R. 2006." Priority Extraction Using Delayed Rewards in Multi Agents Systems: A Case Study in RoboCup". In Proceedings CSICC'06, Iran,571-574.