International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:3, 2008

Social, Group and Individual Mind extracted from Rule Bases of Multiple Agents

P. Cermak

***Abstract*—**This paper shows possibility of extraction Social, Group and Individual Mind from Multiple Agents Rule Bases. Types those Rule bases are selected as two fuzzy systems, namely Mambdani and Takagi-Sugeno fuzzy system. Their rule bases are describing (modeling) agent behavior. Modifying of agent behavior in the time varying environment will be provided by learning fuzzy-neural networks and optimization of their parameters with using genetic algorithms in development system FUZNET. Finally, extraction Social, Group and Individual Mind from Multiple Agents Rule Bases are provided by Cognitive analysis and Matching criterion.

***Keywords*—**Mind, Multi-agent system, Cognitive analysis, Fuzzy system, Neural network, Genetic algorithm, Rule base.

I. Introduction

THIS paper describes how to determine Social, Group and Individual Mind from Rule bases of Multiple Agents in Multiagent system. Situation shows following Fig. 1.

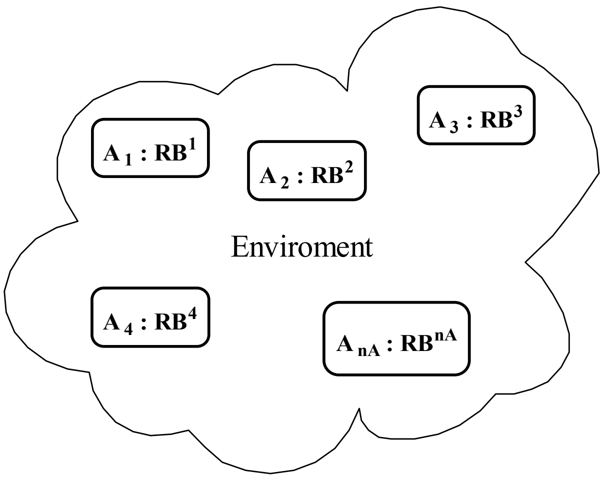

Fig. 1 Multiagent system

Behavior of each agent A is given by rule base RB, *nA* is number of Agent. For Agent behavior modeling we use two fuzzy systems namely Takagi-Sugeno and Mamdani fuzzy systems those systems we select especially for following reasons:

- Prevent oscillations
- Aprox. function between rules are nonlinear
- Rules we can expressed natural language like (small distance, middle distance, long distance)

The rule base of Mamdani fuzzy system is given by

$$\begin{aligned} &IF\ (x_1\ is\ A_{11})\ and\ (x_2\ is\ A_{21})\ and\ ...\ (x_n\ is\ A_{n1})\ THEN\ (y_1\ is\ C_1)\\ &IF\ (x_1\ is\ A_{12})\ and\ (x_2\ is\ A_{22})\ and\ ...\ (x_n\ is\ A_{n2})\ THEN\ (y_2\ is\ C_2)\\ &\vdots\\ &IF\ (x_1\ is\ A_{1r})\ and\ (x_2\ is\ A_{2r})\ and\ ...\ (x_n\ is\ A_{nr})\ THEN\ (y_r\ is\ C_r)\ , \end{aligned} \tag{1}$$

where x are inputs, A_{nr} are linguistic values of rules in antecedent and C_r are linguistic values of rules in consequent.

The rule base of Takagi-Sugeno fuzzy system is given by

$$\begin{aligned} &IF\ (x_1\ is\ A_{11})\ and\ (x_2\ is\ A_{21})\ and\ ...\ (x_n\ is\ A_{n1})\ THEN\\ &\qquad (y_1 = k_{01} + k_{11}x_1 + k_{21}x_2 + ... + k_{n1}x_n)\\ &IF\ (x_1\ is\ A_{12})\ and\ (x_2\ is\ A_{22})\ and\ ...\ (x_n\ is\ A_{n2})\ THEN\\ &\qquad (y_2 = k_{02} + k_{12}x_1 + k_{22}x_2 + ... + k_{n2}x_n)\\ &\vdots\\ &IF\ (x_1\ is\ A_{1r})\ and\ (x_2\ is\ A_{2r})\ and\ ...\ (x_n\ is\ A_{nr})\ THEN\\ &\qquad (y_r = k_{0r} + k_{1r}x_1 + k_{2r}x_2 + ... + k_{nr}x_n)\ , \end{aligned} \tag{2}$$

where x are inputs, A_{nr} are linguistic values of rules in antecedents and y_r are regression function of rules in consequent. Regression functions are linear combination of inputs x.

Those fuzzy systems will be used in composition of the fuzzy neural networks.

II. Building and Learning Single Rule Base

The basic idea of the composition method of the fuzzy neural networks (FNN) is to realize the process of fuzzy reasoning by the structure of neural network and to make the parameters of fuzzy reasoning be expressed by the connection weights of neural network.

A. FNN Mamdani

The FNN can be divided into premise parts and the consequence parts according to the structure of FNN rules. The premise and consequence parts of the FNN are connected in series for using the Back Propagation (BP) learning algorithm. The Fig. 2 shows the case where the FNN have two inputs and two membership function in each premise. The outputs of the units in B-layer are written as

This work was supported in part by the MSM 4781305903, IGSU 37/2006 and Czech Science Foundation GACR 201/04/0528.

Petr Cermak is with the Institute of Computer Science, Silesian University, 13 Bezruc Sq., 746 01 Opava, Czech Republic, (corresponding author to provide phone: +420 553 684 372; fax: +420 553 684 370; email: Petr.Cermak@fpf.slu.cz)

$$\mu(x,s_1,s_2,c)=\begin{cases} e^{-s_1(x-c)^2} & x<c \\ e^{-s_2(x-c)^2} & x\geq c \end{cases} \quad (3)$$

Then the membership functions in the premises are tuned them by modifying their parameters *s1*, *s2* and *c* through the learning with learning rate η_m. The grades of the membership functions are calculated in A – C layers and we using min-max procedure for calculating cut-values. Cut-values of rules are obtained as outputs of the units in D-layer and cutting membership function in layer E. In layer F we calculate intersection and in layer G defuzzification.

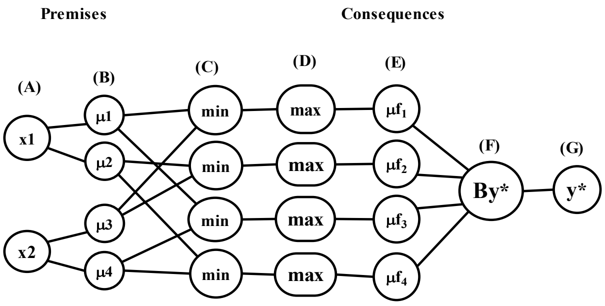

Fig. 2 Fuzzy neural network (Mamdani)

Structure identification is provided by genetic algorithm described in subchapter C. For antecedent and consequent parameters learning we use (BP) learning. First we must determine errors of output and hidden layers and back propagate through FNN. Error of output layer is given by

$$\varepsilon^{(G)}=\frac{\partial E}{\partial y^{(G)}}=y^*-y^0 \quad (4)$$

After back propagate errors into layer (B) and (E) we determine equation for modifying membership function parameters in antecedent

$$p_j^{(B)}(t+1)=p_j^{(B)}(t)-\eta_j^{(B)}\varepsilon_j^{(B)}\frac{\partial f_j^{(B)}\left(z_j^{(B)},p_j^{(B)}\right)}{\partial p_j^{(B)}} \quad (5)$$

and membership function in antecedent parameters by

$$p_j^{(E)}(t+1)=p_j^{(E)}(t)-\eta_j^{(E)}\varepsilon_j^{(E)}\frac{\partial f_j^{(E)}\left(z_j^{(E)},p_j^{(E)}\right)}{\partial p_j^{(E)}} \quad (6)$$

B. FNN Takagi-Sugeno

The FNN can be divided into premise parts and the consequence parts according to the structure of FNN rules. The premise and consequence parts of the FNN are connected in series for using the BP learning algorithm. The Fig. 3 shows the case where the FNN have two inputs and two membership function in each premise. The outputs of the units in B-layer are written in (3).

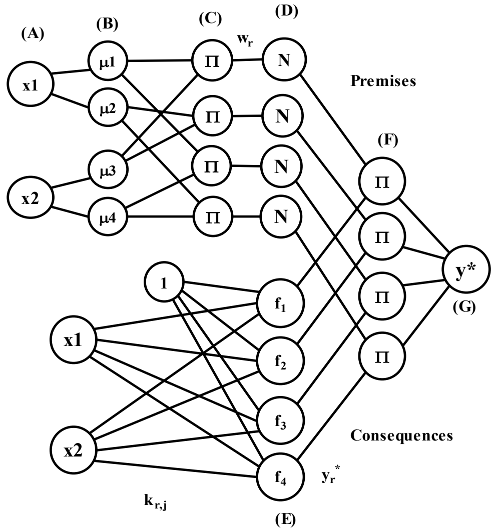

Fig. 3 Fuzzy neural network (Takagi Sugeno)

Then the membership functions in the premises are tuned them by modifying their parameters *s1*, *s2* and *c* through the learning with learning rate η_m. The grades of the membership functions are calculated in A – C layers. The truth-values of rules are obtained as outputs of the units in D-layer. This is given by the normalized products of the grades of the membership functions for each rule by formula

$$\alpha_r=\frac{w_r}{\sum_{r=1}^{n}w_r} \quad (7)$$

The consequents of FNRM rules are expressed by first-order linear equations. The inference method is given using formula

$$y^*=\sum_{r=1}^{n}\alpha_r f_r(x_1,x_2) \quad (8)$$

The Fig. 3 shows the consequence layer E. The inferred value of each rule y_r* is calculated as the output of a unit in E-layer. The global output *y** is calculated in F, G layers. The fuzzy rules are to be identified by modifying *j*-th regression coefficient of *r*-th rule k_{rj} through the learning process with learning rate η_c.

Two basic tasks identification of Takagi-Sugeno fuzzy model inside neural network are structural and parametric identification [1], [2], [3]. Algorithm structural and parametric identification is described in [6]

The primary learning of initial model is performed through ten learning steps providing parameter identification of rule consequents. The state of antecedents remains initial, including total partition of input fuzzy space. The result of primary model is next improved through a procedure of efficiency improvement. For consequent parameters learning we use (BP) learning. We must first determine errors of output and hidden layers and back propagate through FNN. Error of output layer is given by (4). After back propagate errors into layer (B) and (E) we determine equation for modifying membership function parameters in antecedent (5).

And linear regression coefficient in consequent

$$k_{q,r}(t+1) = k_{q,r}(t) - \eta^{(E)}\varepsilon_j^{(E)}\frac{\partial E}{\partial k_{q,r}} \tag{9}$$

where η(E) is learning coefficient in consequent, $k_{q,r}(t)$ are linear regression coefficients in Consequent and εj(E) back propagation errors.

C. Advanced Genetic Algorithm

For parameters optimization of fuzzy neural model and optimizing parameters of rule base has been used genetic optimization algorithm. GA appears from method of natural selection, when subjects with best adaptation to assigned conditions have most chance to survive, as in [4]. GA takes into account next natural mechanism – mutation that restricts risk of degradation, which is hold in local, extreme from optimization viewpoint. GA has iteration character. GA does not work with separate result in particular iterations, but with population. In each iteration GA works with several (generally a lot of results, standard value is hundreds) results, which are included in population and try ensured appearance still better results via genetic operations with these results. Generally scheme of GA:

$$GA = \left(N, P, f, \Theta, \Omega, \Psi, \tau\right) \tag{10}$$

where P is population containing N elements. Θ is parent selection operator, which select u elements from P, Ω is set of genetic operators, which include crossover operator Ω_c, mutation operator Ω_m and others problem-oriented or implementation-oriented specific operators, which all together generate v offspring from u parents. ψ is deletion operator, which removes v selected elements from actual population $P(t)$. v elements is add to new population $P(t+1)$ after it, τ is stop-criterion.

Parent selection operator Θ and genetic operators Ω have stochastic character, deletion operator ψ is generally deterministic.

Refs [5] discuss selection of suitable GA. We need choose such GA for practical intention that will be able to work with smallest population and nevertheless will be converge to global minimum with sufficient probability at minimum numbers of iteration steps. We adjudicate GA with sexual reproduction and GA without sexual reproduction during selecting of suitable GA.

GA without sexual reproduction: Organized selection is use for parent's selection. By crossing is created number of members matching to half population. Population of next generation (iteration) is create from best offspring and parents (Size of population stay constant during running GA), offspring needn't be inserted into population when its value of fitness function isn't better than value of current population.

GA with sexual reproduction: Members contain one chromosome. Population is divided into two parts according to sex in this version of GA. Pertinence to sex is determined by sexual gene.

It was selected GA with sexual reproduction contains one chromosome with restricted lifetime parameter 5 iterations, uniform crossing and genes of type 3/3/5 after considered properties of each GA. There are selected Gray code for encrypt parameters to chromosome which is useful by reason of bypassing so-called Hamming barrier. Selection strategy dominant male was used in contrast to [6].

Chromosome is compound of genes; each gene presents 1 bit of value, which is represented by chromosome. However, gene contains more than 1 bit (using redundancy encrypt). Bit values inside gene are mapped on outside value of gene (0 or 1) via specific map function when border between 0 and 1 is not crisp, but exists so-called "shade zone" where value carried by gene is determined randomly. The advanced GA used for two optimizing tasks:

1. Optimizing parameters of FNN
1) Levels coefficients for structure identification
2) Number iteration of structure identification (niter)
3) Learning coefficients of consequents
4) Learning coefficient of membership function

2. Optimizing behavior of rule bases
1) Numbers of rules
2) Structures of Terms in antecedents and consequents
3) Shapes of membership functions
4) Parameters of Terms in antecedents and consequents

Cost function of GA is absolute error of fuzzy neural network is given by:

$$AvgE = \frac{\sum_{k=1}^{K}\left|y^0(k) - y^*(k)\right|}{K} \tag{11}$$

D. The FuzNet

A developed programme tool FUZNET was based on fuzzy neural network technology including the new identifying procedures presented [3]. This fuzzy neural network environment appears to be suitable for fuzzy non-linear model identification as well as introduces the programme FUZNET including its application in real system modeling.

This system is extension of the neural network (NN) development system based on the Back Propagation algorithm. Figure Fig.4. shows the FUZNET structure:

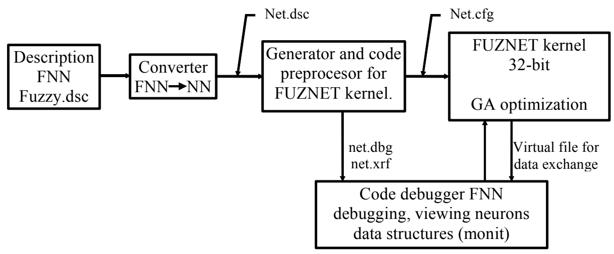

Fig. 4 The FUZNET

Now we divide the FUZNET structure description to three parts: pseudo code generator and code pre-processor, FUZNET kernel, and additional blocks. The pseudo-code generator contains two executables. Generator and pre-processor create all files, which are necessary for the adaptation, testing and debugging process. FUZNET kernel contains the function set, which allows adaptation, test and initialize/retrieve operation with FNN parameters. Additional blocks are implemented, a block for parameters optimization of system FUZNET with using genetic optimization algorithm [3], [6]. The last block online learning is a block with algorithm mentioned in [7].

III. Cognitive Analysis Rule Bases of Multiple Agents - CAMAS

A. Cognitive Analysis

Offered cognitive analysis is based on Question Answering Mechanism. With this analysis we can determine how *r*-th rule from rule base RB^i is **consistent** with other RB^j rule base.

Lets we have as vector $\mathbf{x}_r^i$ that contain all linguistic values antecedent part r-th rule from rule base RB^i

$$\mathbf{x}_r^i = \left(A_{1r}^i, A_{2r}^i, \ldots, A_n^i\right) \tag{12}$$

and question fuzzy system with RB^j to response on $\mathbf{x}_r^i$

$$By_{r,j}^{i\;*} = \text{Response}\left(RB^j, \mathbf{x}_r^i\right). \tag{13}$$

And check consistency with consequent part *r*-th rule C_r^i from RB^i as follows

Version Mamdani:

$$Cons_{r,j}^i = \frac{\int_0^1 By_{r,j}^{i\;*} \cap C_r^i}{\int_0^1 By_{r,j}^{i\;*}}. \tag{14}$$

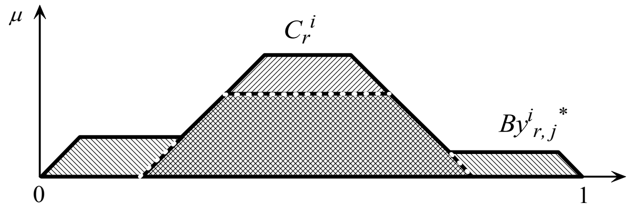

Fig. 5 Intersection response of *i*-th fuzzy system and consequent of *r*-th rule

Version Takagi-Sugeno:

$$Cons_{r,j}^i = \frac{E_r^i}{\sum_{r=1}^{R} E_r^i}. \tag{15}$$

Determining rule error E_r^i is described in [6], [7].

We must first define Consistency matrix and matching criterion.

$$CMA^i = \begin{bmatrix} Cons_{1,1}^i & Cons_{1,2}^i & \cdots & Cons_{1,nA}^i \\ Cons_{2,1}^i & Cons_{i,2}^2 & \cdots & Cons_{i,nA}^2 \\ \cdots & \cdots & \cdots & \cdots \\ Cons_{nR_i,1}^i & Cons_{nR_i,2}^i & \cdots & Cons_{nR_i,nA}^i \end{bmatrix}, \tag{16}$$

where nR_i is number of rules in rule base RB^i.

Rows determine consistency for each rules and columns consistency for all rules basis. Matching criterion must decide if consistency is satisfied or unsatisfied. The satisfy we define as equation

$$CMAS_{r,j}^i = \begin{cases} Cons_{r,j}^i \geq \varepsilon_j & 1 \\ Cons_{r,j}^i < \varepsilon_j & 0 \end{cases} \tag{17}$$

Dependency matrix *i*-th rule from base RB^i to all rule basis of MAS is defined as

$$DMAS^i = \begin{bmatrix} DMAS_{1,1}^i & \cdots & DMAS_{nR_i,nA}^i \\ \cdots & \cdots & \cdots \\ DMAS_{nR_i,1}^i & \cdots & DMAS_{nR_i,nA}^i \end{bmatrix}. \tag{18}$$

Dependency Histogram defines numbers of dependencies i-th rule RB^i to each rule bases of MAS.

$$DMASN^i = \begin{bmatrix} \sum_{j=1}^{nA} DMAS^i_{1,j} \\ \cdots \\ \sum_{j=1}^{nA} DMAS^i_{nR_i,j} \end{bmatrix}. \qquad (19)$$

B. Social Mind

Social Mind of i-th Agent is given by set of rules from RB^i consistent for all rule basis of MAS as follows

$$\mathrm{M}^i_{Social}\left\{R_{i,r}\ ;\ DMASN^i[r] = nA, r = (1...nR_i)\right\}, \qquad (20)$$

and rule set of Social Mind for all possible Agents is given by

$$\mathrm{M}_{Social} = \left\{\mathrm{M}^i_{Social}\ ,\ i = (1...nA)\right\}. \qquad (21)$$

C. Individual Mind

Individual mind i-th Agent is given by set of rules from RB^i consistent only with RB^i

$$\mathrm{M}^i_{Individual} = \begin{Bmatrix} R^i_r\ ;\ DMASN^i[r] = 1 \wedge DMAS^i_{r,i} = 1, \\ r = (1...nR_i) \end{Bmatrix} \qquad (22)$$

D. Group Mind

$$\mathrm{M}_{Group} = \begin{Bmatrix} R^i_r\ ;\ DMAS^i_{r,j} = 1 \forall j \in GROUP, \\ r = (1...nR_i), j = (1...nA) \end{Bmatrix} \qquad (23)$$

GROUP is set of indexes all agents in group.

For example we define Multi agent system with 3 Agents:
$MAS = \{A1, A2, A3\}$
Each agent behavior described by rule base

$$A_1 : RB^1 = \left\{R^1_1, R^1_2, R^1_3\right\}$$
$$A_2 : RB^2 = \left\{R^2_1, R^2_2, R^2_3\right\}$$
$$A_3 : RB^3 = \left\{R^3_1, R^3_2, R^3_3\right\}$$

Dependency matrixes are set in example follows

$$DMAS^1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix},\ DMAS^2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix},$$
$$DMAS^3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

dependency histograms with using (18)

$$DMASN^1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, DMASN^2 = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}, DMASN^3 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$$

and each types of Mind

$$\mathrm{M}_{Social} = \left\{R^1_3, R^2_1\right\}$$
$$\mathrm{M}^1_{Individual} = \left\{R^1_1\right\}, \mathrm{M}^2_{Individual} = \left\{R^2_2\right\}, \mathrm{M}^3_{Individual} = \left\{R^3_2\right\}$$
$$\mathrm{M}^{A1,A2}_{Group} = \left\{R^1_2, R^2_3\right\}, \mathrm{M}^{A2,A3}_{Group} = \left\{R^3_3\right\}, \mathrm{M}^{A1,A3}_{Group} = \left\{R^3_3\right\}$$

IV. Conclusion

This work will be useful in study emergent time varying processes on multiagent systems and especially for building the supervisor system, which can make analysis of behaviour whole multiagent system and rule bases of course. This system will be select the best strategies and groups of agents for effectively solving given Tasks or Subtasks. For experiments will be used Khepera and Koala robot platform. Part of this work will be used in studying some phenomena in astrophysical systems.

References

[1] Takagi,T., Sugeno,M.: Fuzzy Identification of Systems and Its Application to Modelling and Control, IEEE Trans., MAC, 15, No.1, 116-132.
[2] Pokorny.M., Čermak P.: A Non-linear Regression Analyze Through Neuro-fuzzy Modeling, The 3rd Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty, Osaka, Japan, 30-31 October, 2000.
[3] Cermak,P., Pokorny,M.: An Improvement of Non-Linear Neuro-Fuzzy Model Properties , Neural Network World, ICS AV CR, Praha, 2001, ISSN 1210-0552, pp. 503-523.
[4] Goldberg,D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publ. Comp., INC., 1989, ISBN 0-201-15767-5
[5] Roupec, J.: Genetic optimization algoritms in fuzzy regulators, dissertation thesis, Brno, 1999.
[6] Cermak,P., Chmiel,P.: Parameters optimization of fuzzy-neural dynamic models, 23rd International Conference of the North American Fuzzy Information Processing Society June 27-30, 2004 Banff, AB, Canada, ISBN 0-7803-8377-X, pp. 762-767.
[7] Cermak,P.: Online Learning of Neural Takagi-Sugeno Fuzzy Model, 24rd International Conference of the North American Fuzzy Information Processing Society June 22-25, 2005 Ann Arbor, Michigan,USA, ISBN 0-7803-9188-8, IEEE Catalog No. 05TH8815C.