Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Fuzzy Sets and Systems 142 (2004) 467–488

FUZZY sets and systems

www.elsevier.com/locate/fss

An intelligent fuzzy agent for meeting scheduling decision support system

Chang-Shing Lee*, Chen-Yu Pan

Department of Information Management, Chang Jung University, Tainan 711, Taiwan

Received 20 November 2002; received in revised form 25 April 2003; accepted 7 May 2003

Abstract

An *Intelligent Fuzzy Agent* for *Meeting Scheduling Decision Support System* is proposed in this paper. The proposed *Intelligent Fuzzy Agent* including *Meeting Negotiation Agent*, *Fuzzy Inference Agent* and *Genetic Learning Agent* can search and decide the suitable meeting time for the specified meeting in an organization. When a meeting host requests a meeting, the *Meeting Negotiation Agent* immediately sends the invitees' names to *Meeting Scheduling Decision Support System* for retrieving their schedules from *Group Calendar Data Base*, then *Meeting Scheduling Decision Support System* will compute the possible meeting time and respond the results to *Meeting Negotiation Agent*. Moreover, the *Fuzzy Inference Agent* infers the adequate meeting time based on the information provided by *Meeting Negotiation Agent* and *Personalized Knowledge Base*, and sends the computing result back to *Meeting Scheduling Decision Support System*. The meeting host will decide the final meeting time based on *Meeting Scheduling Decision Support System* and announce the meeting information by various devices including PDA, WAP, FAX, or E-mail. Furthermore, the invitees' decisions for attending the meeting or not will be stored into *Meeting Information Knowledge Base*, then the *Genetic Learning Agent* will adjust the *Personalized Knowledge Base* for the next meeting. By the experimental results, the proposed *Intelligent Fuzzy Agent* can work efficiently and effectively for *Meeting Scheduling Decision Support System*.

Keywords: Fuzzy inference systems; Decision support system; Genetic algorithm; Meeting scheduling; Agent

1. Introduction

An agent is a program that performs unique tasks without direct human supervision. Ferber [7] gives another definition of an agent such as "an agent is capable of acting in an environment

* Corresponding author. Tel.: 886-6-2785123+2059; fax: 886-6-2785657.
E-mail addresses: leecs@mail.cju.edu.tw, leecs@cad.csie.ncku.edu.tw (C.-S. Lee).

0165-0114/$ - see front matter
doi:10.1016/S0165-0114(03)00201-X

and can communicate directly with other agents". An intelligent agent is more powerful than an agent because of the reasoning and learning capabilities [4]. There are many meeting scheduling approaches that have been proposed. For example, Sen [14] proposes a software system that uses intelligent meeting-scheduling agents negotiating with other agents without compromising their user-specified constrains. The author uses a distributed approach with intelligent agents to design and develop efficient meeting scheduling. But the approach lacks of invitee's behavior learning and reasoning mechanism. Sugihara et al. [15] propose a meeting scheduler for office automation. They consider the priorities on persons and meetings in a real office environment, and use a heuristic algorithm for timetable rearrangement. Haynes et al. [8] propose an automated meeting scheduling system that utilizes user preferences. They highlight the usage of user preferences and priorities by the scheduling agent with regards to time of day, day of week, status of other invitees, topic of the meeting, and so on. But they do not investigate learning user preferences, either from observed behavior of the user or from querying other agents. Jeong et al. [9] focus their research on how the meeting-scheduling agent can reduce failures when there is no common time-slot. They solve the failure condition with utilizing the cooperation and the rescheduling strategy.

Besides, there are other meeting-scheduling systems that have been proposed. For example, Mynatt et al. [13] present a calendar's system, Ashir et al. [1] propose a multi-agent decision system, Bergenti et al. [2] describe an agent-based computer-supported system, and Lee et al. propose a fuzzy decision agent for meeting scheduling [10]. The common drawback of the proposed systems is lack of personalized learning invitees' behavior. On the other hand, Lin et al. [11] propose the fuzzy neural network architecture for control and decision system. Cordon et al. [3,5,6] propose a knowledge base generating and learning approach for a fuzzy system. The knowledge base contains a database (DB) and a rule base (RB). The proposed approach can automatically learn the DB and RB based on genetic algorithm.

In this paper, we propose a novel genetic-based fuzzy decision agent for meeting scheduling support system based on the learning mechanisms introduced in [3,5,6,11]. The proposed Intelligent Fuzzy Agent (IFA) contains a Meeting Scheduling Agent (MNA), a Fuzzy Inference Agent (FIA) and a Genetic Learning Agent (GLA) to carry out the suitable meeting time selection. Moreover, the Meeting Scheduling Decision Support System (MSDSS) will process the meeting information and send the selected suitable meeting time to the meeting host for deciding the final meeting time. In addition, the Group Calendar Data Base (GCDB) and Meeting Information Knowledge Base (MIKB) will store the personal calendar and personal meeting behavior. The details will be described in the following sections. The rest of this paper is organized as follows. In Section 2, the architecture of Meeting Scheduling Decision Support System is proposed. Section 3 introduces the Intelligent Fuzzy Meeting Agent. The experimental results are shown in Section 4. Finally, the conclusions and discussions are given in Section 5.

2. Meeting scheduling decision support system

In this section, we will brief describe the functionality of the meeting scheduling decision support system including the intelligent fuzzy agent. The intelligent fuzzy agent contains the meeting negotiation agent, fuzzy inference agent and genetic learning agent to perform the reasoning and

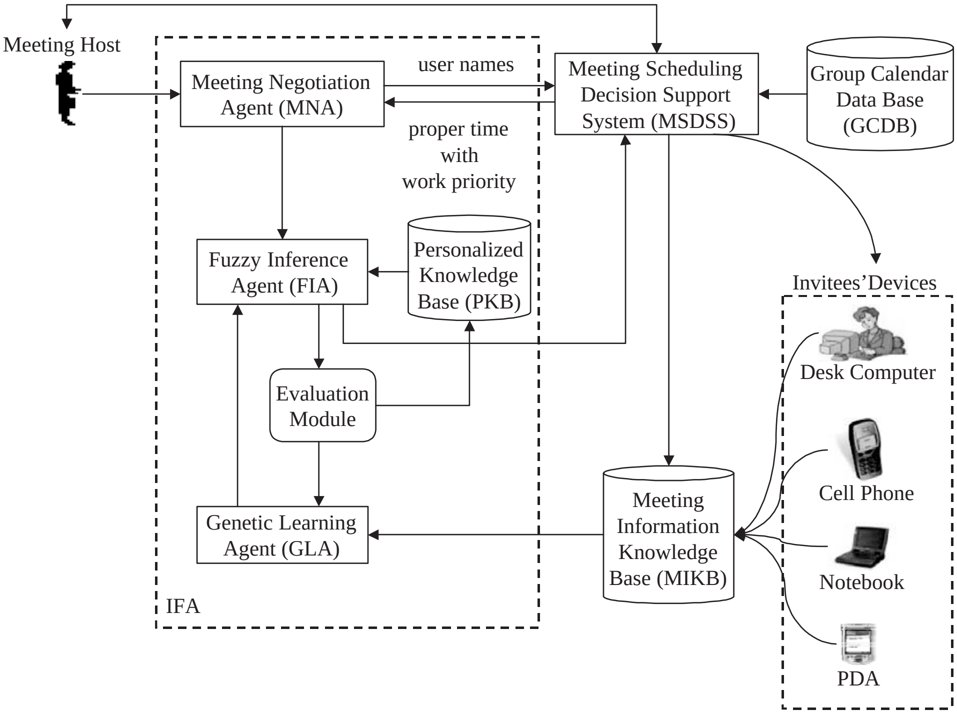


Fig. 1. The architecture of the meeting scheduling decision support system.

learning tasks. In addition, the personalized knowledge base is also used in this agent. Fig. 1 shows the architecture of the meeting scheduling decision support system.

In Fig. 1, when a meeting host requests a meeting, Meeting Negotiation Agent (MNA) receives the invitees' names, meeting time slot and meeting subject, then immediately sends the meeting information to MSDSS and Fuzzy Inference Agent (FIA). MSDSS obtains the invitees' names, retrieves their schedules from Group Calendar Data Base (GCDB), computes the common free time or removable working time for all invitees, and then responds the computing results to MNA. Moreover, FIA infers the adequate meeting time with the attending possibility of all invitees based on the information provided by MNA and Personalized Knowledge Base (PKB). The parameters of the fuzzy variables that represent the behavior of the invitee will be stored in PKB for FIA. MNA sends the meeting information including meeting time length, invitee's names and their priority, meeting event importance, and meeting time preference with working priority to FIA. In addition, FIA also retrieves the PKB to get the meeting time preference of all the invitees for computing the attending possibility and sends the computing results to MSDSS. The meeting host will receive the adequate meeting time slots from MSDSS and decide the final meeting time, then send the result back to MSDSS. Finally, MSDSS will refer the GCDB to get the personal profile and announce the meeting information by various platforms including PDA, WAP, FAX, or E-mail. Furthermore, the actual decisions of the invitees will be stored into Meeting Information Knowledge Base (MIKB) for genetic leaning invitee's personal behavior. To solve the contradictory information problem, the GLA will judge the consistency of the training data set retrieved from MIKB first. If there is a case with different output (AMP) for the same input (UP, MEP, MTL, MEP1, and MTP2), then GLA will discard the contradictory information. The fuzzy inference agent and genetic learning agent will be detailed introduced in the next section.

3. Intelligent fuzzy agent

The Intelligent Fuzzy Agent (IFA) contains three sub-agents including Meeting Negotiation Agent (MNA), Fuzzy Inference Agent (FIA) and Genetic Learning Agent (GLA) to assist the meeting host in holding the meeting. MNA gathers the meeting invitees' names from meeting host and sends them to FIA and MSDSS. In addition, MNA also receives the computed results of MSDSS and sends them to FIA. The architecture and inference mechanism of FIA will be introduced below.

3.1. Fuzzy inference agent

The architecture of Fuzzy Inference Agent (FIA) in the IFA is shown in Fig. 2.

In Fig. 2, the structure consists of five layers [10,11]. There are five kinds of nodes in this model: input linguistic node, input term node, rule node, output term node and output linguistic node. A fuzzy linguistic node represents a fuzzy variable. A term node represents the mapping degree of the

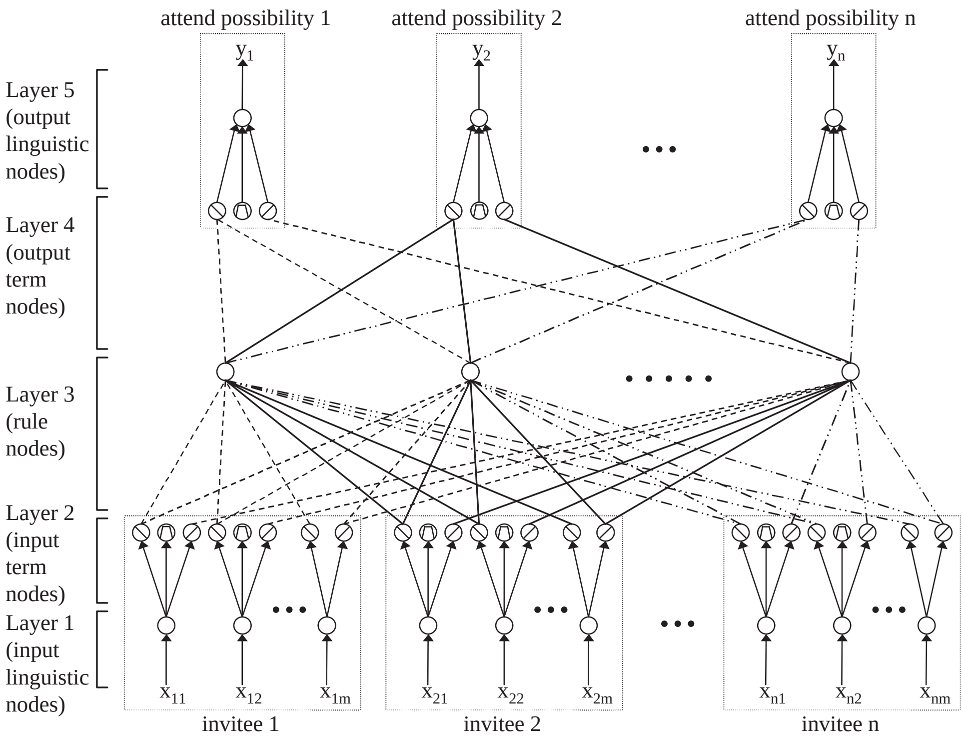


Fig. 2. The architecture of Fuzzy Inference Agent.

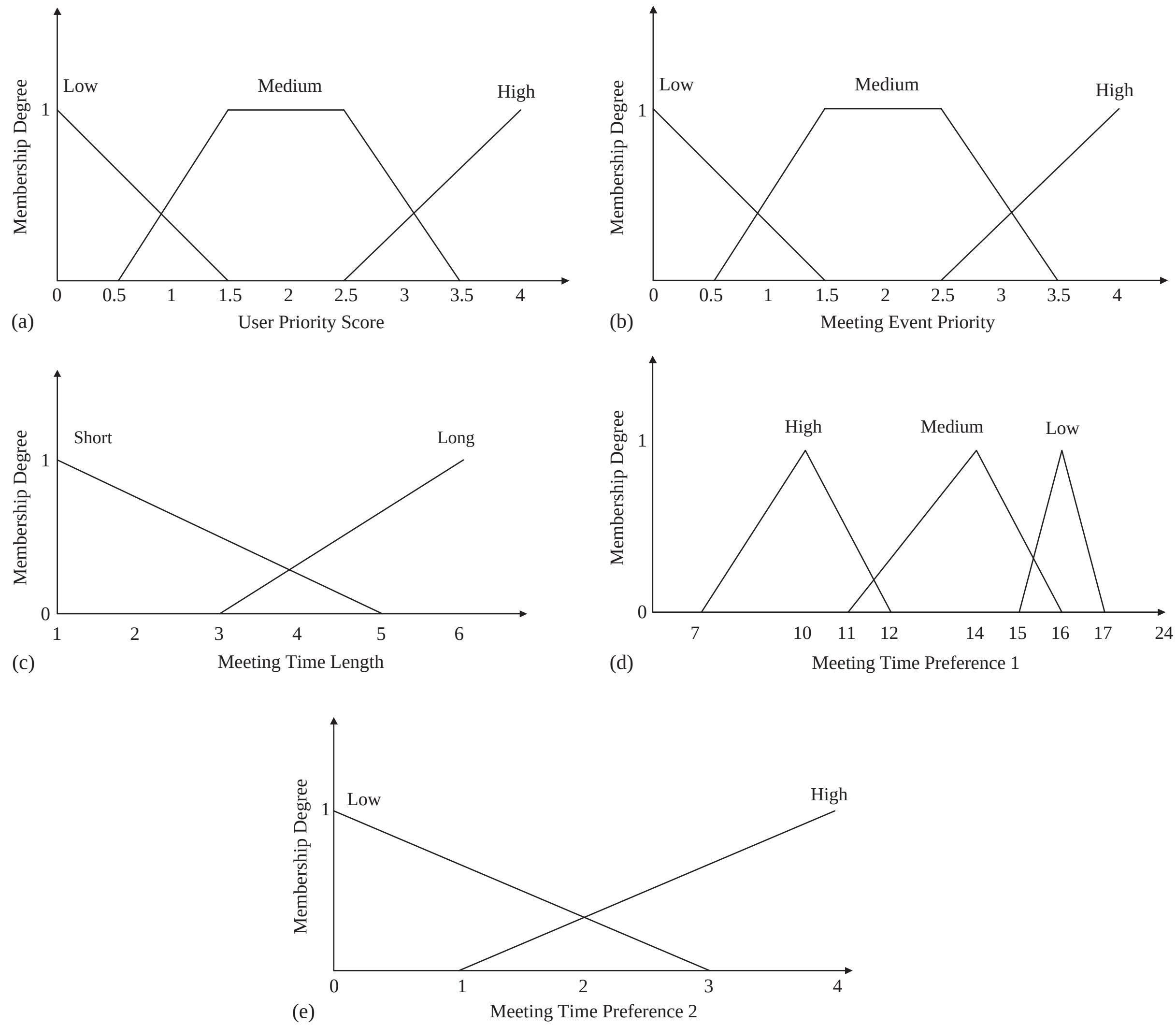


Fig. 3. (a) Fuzzy sets for UP fuzzy variable. (b) Fuzzy sets for MEP fuzzy variable. (c) Fuzzy sets for MTL fuzzy variable. (d) Fuzzy sets for MTP1 fuzzy variable. (e) Fuzzy sets for MTP2 fuzzy variable.

fuzzy variable. A rule node represents a rule and decides the final firing strength of that rule during inferring. Now, we will describe each layer in details. The first two layers are called the premise layer, which is used to represent the premise part of the fuzzy system.

Layer 1: The nodes in first layer just directly transmit input values to next layer. If the input vector of the ith invitee is $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$, where x_{im} is denoted as the input value of mth fuzzy variable for ith invitee, then the output vector of ith invitee for this layer will be

$$\mu_i^1 = ((x_{i11}, x_{i12}, \ldots, x_{i1N_1}), (x_{i21}, x_{i22}, \ldots, x_{i2N_2}), \ldots, (x_{im1}, x_{im2}, \ldots, x_{imN_m})), \tag{1}$$

where x_{ijk} is input value of the kth linguistic term in the jth fuzzy variable for ith invitee.

Table 1
Fuzzy Rules Base for Fuzzy Expert System (L:Low, M:Medium, H:High, Lg:Long, S:Short)

Rule	UP	MEP	MTL	MTP(1)	MTP(2)	AMP	Rule	UP	MEP	MTL	MTP(1)	MTP(2)	AMP
1	L	L	Lg	L	L	L	55	M	M	S	L	L	M
2	L	L	Lg	L	H	L	56	M	M	S	L	H	H
3	L	L	Lg	M	L	L	57	M	M	S	M	L	M
4	L	L	Lg	M	H	L	58	M	M	S	M	H	H
5	L	L	Lg	H	L	L	59	M	M	S	H	L	H
6	L	L	Lg	H	H	M	60	M	M	S	H	H	H
7	L	L	S	L	L	L	61	M	H	Lg	L	L	H
8	L	L	S	L	H	M	62	M	H	Lg	L	H	H
9	L	L	S	M	L	L	63	M	H	Lg	M	L	H
10	L	L	S	M	H	M	64	M	H	Lg	M	H	H
11	L	L	S	H	L	M	65	M	H	Lg	H	L	H
12	L	L	S	H	H	M	66	M	H	Lg	H	H	H
13	L	M	Lg	L	L	L	67	M	H	S	L	L	H
14	L	M	Lg	L	H	L	68	M	H	S	L	H	H
15	L	M	Lg	M	L	L	69	M	H	S	M	L	H
16	L	M	Lg	M	H	M	70	M	H	S	M	H	H
17	L	M	Lg	H	L	L	71	M	H	S	H	L	H
18	L	M	Lg	H	H	M	72	M	H	S	H	H	H
19	L	M	S	L	L	L	73	H	L	Lg	L	L	L
20	L	M	S	L	H	M	74	H	L	Lg	L	H	M
21	L	M	S	M	L	M	75	H	L	Lg	M	L	L
22	L	M	S	M	H	M	76	H	L	Lg	M	H	M
23	L	M	S	H	L	M	77	H	L	Lg	H	L	M
24	L	M	S	H	H	H	78	H	L	Lg	H	H	M
25	L	H	Lg	L	L	M	79	H	L	S	L	L	M
26	L	H	Lg	L	H	M	80	H	L	S	L	H	M
27	L	H	Lg	M	L	M	81	H	L	S	M	L	M
28	L	H	Lg	M	H	M	82	H	L	S	M	H	H
29	L	H	Lg	H	L	M	83	H	L	S	H	L	M
30	L	H	Lg	H	H	M	84	H	L	S	H	H	H
31	L	H	S	L	L	H	85	H	M	Lg	L	L	M
32	L	H	S	L	H	M	86	H	M	Lg	L	H	M
33	L	H	S	M	L	M	87	H	M	Lg	M	L	M
34	L	H	S	M	H	H	88	H	M	Lg	M	H	H
35	L	H	S	H	L	M	89	H	M	Lg	H	L	M
36	L	H	S	H	H	H	90	H	M	Lg	H	H	H
37	M	L	Lg	L	L	L	91	H	M	S	L	L	M
38	M	L	Lg	L	H	L	92	H	M	S	L	H	H
39	M	L	Lg	M	L	L	93	H	M	S	M	L	M
40	M	L	Lg	M	H	M	94	H	M	S	M	H	H
41	M	L	Lg	H	L	L	95	H	M	S	H	L	H
42	M	L	Lg	H	H	M	96	H	M	S	H	H	H
43	M	L	S	L	L	L	97	H	H	Lg	L	L	H
44	M	L	S	L	H	M	98	H	H	Lg	L	H	H
45	M	L	S	M	L	M	99	H	H	Lg	M	L	H
46	M	L	S	M	H	M	100	H	H	Lg	M	H	H
47	M	L	S	H	L	M	101	H	H	Lg	H	L	H

Table 1. (*Continued*)

Rule	UP	MEP	MTL	MTP(1)	MTP(2)	AMP	Rule	UP	MEP	MTL	MTP(1)	MTP(2)	AMP
48	M	L	S	H	H	H	102	H	H	Lg	H	H	H
49	M	M	Lg	L	L	L	103	H	H	S	L	L	H
50	M	M	Lg	L	H	M	104	H	H	S	L	H	H
51	M	M	Lg	M	L	L	105	H	H	S	M	L	H
52	M	M	Lg	M	H	M	106	H	H	S	M	H	H
53	M	M	Lg	H	L	M	107	H	H	S	H	L	H
54	M	M	Lg	H	H	M	108	H	H	S	H	H	H

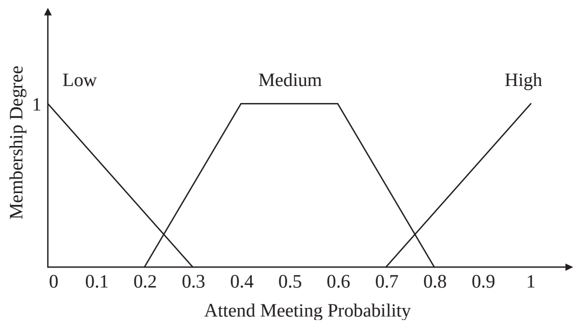


Fig. 4. Fuzzy sets for Attend Meeting Probability fuzzy variable.

Layer 2: Each fuzzy variable of the second layer appearing in the premise part is represented with a condition node. Each of the outputs of the condition node is connected to rule nodes in the third layer to constitute a condition specified in some rules. Note that the output links must be emitted from proper linguistic terms as specified in fuzzy rules. This layer performs the first inference step to compute matching degrees. If the input vector of this layer is $\mu_i^1 = ((x_{i11}, x_{i12}, \ldots, x_{i1N_1}), (x_{i21}, x_{i22}, \ldots, x_{i2N_2}), \ldots, (x_{im1}, x_{im2}, \ldots, x_{imN_m}))$, then the output vector will be

$$\mu_i^2 = ((u_{i11}^2, u_{i12}^2, \ldots, u_{i1N_1}^2), (u_{i21}^2, u_{i22}^2, \ldots, u_{i2N_2}^2), \ldots, (u_{im1}^2, u_{im2}^2, \ldots, u_{imN_m}^2)), \tag{2}$$

where u_{ijk}^2 is the membership degree of the kth linguistic term in the jth fuzzy variable for ith invitee. In this paper, the triangular function and trapezoidal function are adopted as the membership functions of linguistic terms. The trapezoidal membership function is specified by four parameters $[l, h_1, h_2, r]$ as follows:

$$u(x) = \begin{cases} 0, & x < 1, \\ (x-l)/(h_1-l), & l \leqslant x < h_1, \\ 1, & h_1 \leqslant x < h_2, \\ (r-x)/(r-h_2), & h_2 \leqslant x < r, \\ 0, & x \geqslant r. \end{cases} \tag{3}$$

Obviously, the triangular membership function is a special case of the trapezoidal function when $h_1 = h_2$.

X_{i1}	X_{i2}	...	X_{im}	Y_i

Fig. 5. Chromosome of the CS_a part for invitee i.

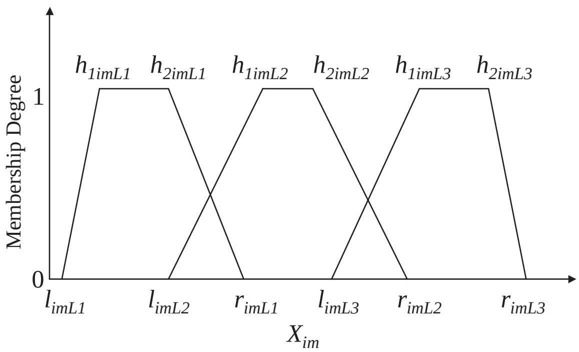


Fig. 6. The X_{im} fuzzy variable with three linguistic terms.

There are five input fuzzy variables used in FIA for MSDSS. They are User_Priority (UP) fuzzy variable denoted the importance of invitee in the meeting, Meeting_Event_Priority (MEP) fuzzy variable denoted the important of the meeting, Meeting_Time_Length (MTL) fuzzy variable denoted the length of meeting time, Meeting_Time_Preference_1 (MTP1) fuzzy variable, and Meeting_Time_Preference_2 (MTP2) fuzzy variable. MTP1 fuzzy variable denotes the preference of meeting time for invitee. For example, if most people prefer to attending the meeting at 10:00 AM in the morning, but the meeting time is 9:30 AM, then FIA may infer the possibility of attending this meeting for all participants is high. MTP2 fuzzy variable is used to consider the work priority of each invitee's schedule. For instance, if a meeting event priority for user A is low, and A has another high priority work to do at the meeting time, then A may not attend the meeting this time. Figs. 3(a)–(e) show the fuzzy sets of fuzzy variables UP, MEP, MTL, MTP1 and MTP2, respectively. Notice that the membership functions of *Low*, *Median* and *High* for user priority score are predefined by the domain expert and will be learned by GLA. There are similar reasons for the other fuzzy variables shown in Figs. 3(b)–(e).

Now we show an example for layer 1 and layer 2 as follows. If the input parameter set x_i for invitee i is $(2,4,2,10,4)$, that is, the user priority is 2, the meeting event priority is 4, the meeting time length is 2 hours, the meeting time is at 10:00 AM, and meeting time preference is 4, then the output vector for layer 1 is $\mu_i^1 = ((2,2,2),(4,4,4),(2,2),(10,10,10),(4,4))$. In addition, the input vector for layer 2 is $\mu_i^1 = ((2,2,2),(4,4,4),(2,2),(10,10,10),(4,4))$, and the output vector of layer 2 is the membership degree set $\mu_i^2 = ((0,1,0),(0,0,1),(0.75,0),(1,0,0),(0,41)$ based on the membership functions defined in Fig. 3.

Layer 3: The third layer is called the rule layer where each node is a rule node to represent a fuzzy rule. The links in this layer are used to perform precondition matching of fuzzy logic rules. Hence, the rule nodes should perform the fuzzy AND operation [11], the outputs will be linked with associated linguistic node in the fourth layer, and each output link has its weight w. In our model, the rules are defined by domain expert's knowledge previously, and we show them in Table 1. Eq. (4) shows the precondition matching degree μ_{i1}^3 and the output linking weight w_{i1}^3 of *Rule node 1*

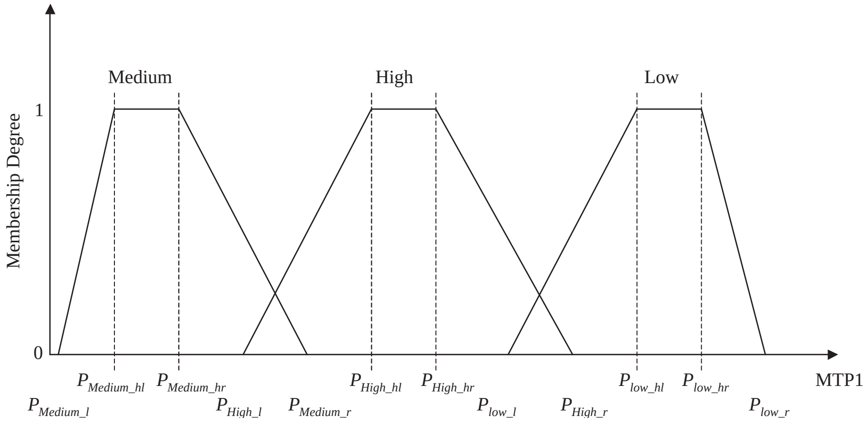


Fig. 7. The position of each linguistic term of MTP1.

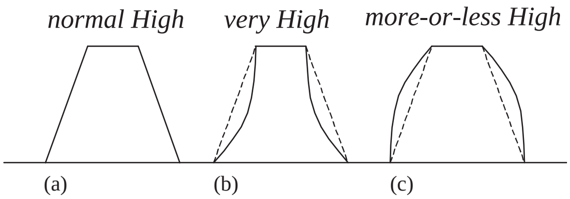


Fig. 8. The graphic representation for linguistic terms *normal High*, *very High* and *more-or-less High*.

for ith invitee as follows:

$$\mu_{i1}^3 = \min(u_{i11}^2, u_{i21}^2, \ldots, u_{im1}^2),$$
$$w_{i1}^3 = \prod_{t=1}^{m} u_{it1}^2. \tag{4}$$

Layer 4: The fourth layer and the fifth layer are called the output term node and output linguistic node respectively, and they are also called the conclusion layer. The output fuzzy variable for FIA is Attend_Meeting_Possibility (AMP) denoted the possibility of attending the meeting for each invitee. There are three linguistic terms *Low*, *Median*, and *High* in the output fuzzy variable. Fig. 4 shows the fuzzy sets of AMP.

The output term node performs the fuzzy OR operation to integrate the fired rules which have the same consequence. If P_L, P_M, and P_H denote the rule nodes output linking to the linguistic terms *Low*, *Medium*, and *High* respectively. Then the output of layer four is as follows:

$$\mu_i^4 = \left(\mathit{Centroid}\left(\max_{P_\mathrm{L} \in Low} \{\mu_{iP_\mathrm{L}}^3\} \times Low \right),\ \mathit{Centroid}\left(\max_{P_\mathrm{M} \in Medium} \{\mu_{iP_\mathrm{M}}^3\} \times Medium \right),\right.$$
$$\left. \mathit{Centroid}\left(\max_{P_\mathrm{H} \in High} \{\mu_{iP_\mathrm{H}}^3\} \times High \right) \right), \tag{5}$$

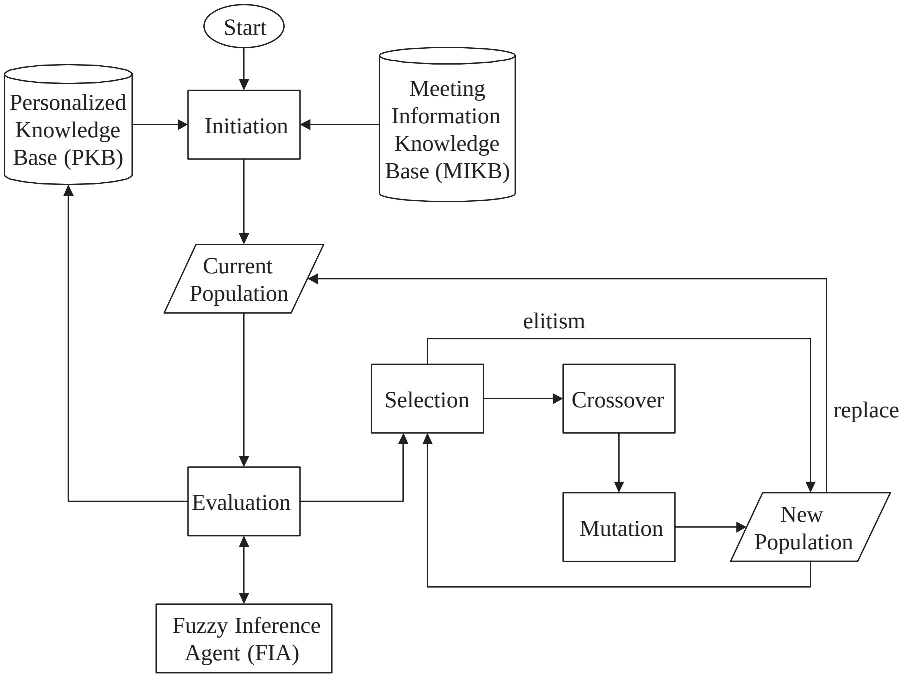


Fig. 9. The flow chart of GLA.

where the function $Centroid(\cdot)$ denotes the centroid defuzzification process.

Layer 5: Support that the $V_{i1} = Centroid(\max_{P_L \in Low}\{\mu^3_{iP_L}\} \times Low)$, $V_{i2} = Centroid(\max_{P_M \in Medium}\{\mu^3_{iP_M}\} \times Medium)$, and $V_{i3} = Centroid(\max_{P_H \in High}\{\mu^3_{iP_H}\} \times High)$, then the AMP value for invitee i is as follows:

$$y_i = \frac{\sum_{q=1}^{3} w_{iq} \times V_{iq}}{\sum_{q=1}^{3} w_{iq}}, \tag{6}$$

where w_q denotes the integrated weight of rule nodes for each output linguistic term. For example, Eq. (7) shows the weight w_{i1} of linguistic term *Low* for invitee i.

$$w_{i1} = \sum_{P_L \in Low} w^3_{iP_L} \times \sigma, \quad \text{where } \sigma = \begin{cases} 1, & \mu^3_{iP_L} \text{ is maximum,} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Next, we will introduce the genetic learning agent as follows.

3.2. Genetic learning agent

This subsection will introduce the GLA for personalized meeting behavior learning. The factors of meeting behavior considered here are the fuzzy variables UP, MEP, MLT, MTP1, MTP2, and AMP. After holding a meeting, the invitees' meeting records will be stored into MIKB. GLA immediately retrieves the personal meeting records to be the training data from MIKB and sends them to FIA for getting the invitees' meeting behavior. The important questions when using genetic learning are

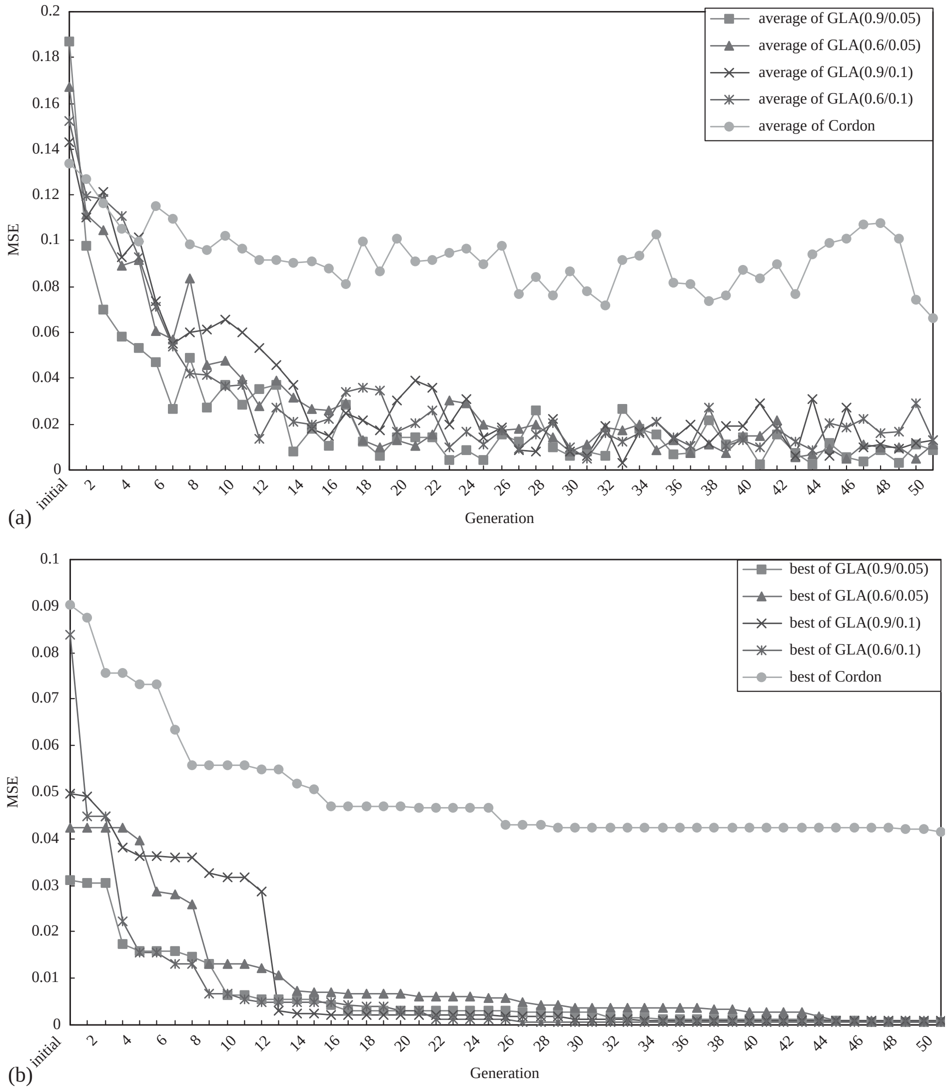


Fig. 10. (a) The convergence curves of the population for the 15 times meeting behavior learning. (b). The convergence curves of the elitist chromosome for the 15 times meeting behavior learning.

how to encode each solution, how to evaluate these solutions and how to create new solutions from existing ones. Moreover, it is relatively important choice of the initial population, because we can quickly obtain the better solutions if an adequate initial gene pool is chosen [5,6]. In this subsection, we apply the learning approach proposed by Cordon et al. [5,6] to learn the PKB containing Personal DB (PDB) and Personal RB (PRB) for the next meeting behavior learning. Now we describe the main components of GLA as follows. The two components of the PKB to be encoded are the membership functions of the fuzzy variables (PDB) and the linguistic modifiers of the linguistic terms (PRB). Therefore, each chromosome will be composed of the above two parts, called CS_a part and CS_b part, respectively. Fig. 5 shows the chromosome of CS_a part for invitee i.

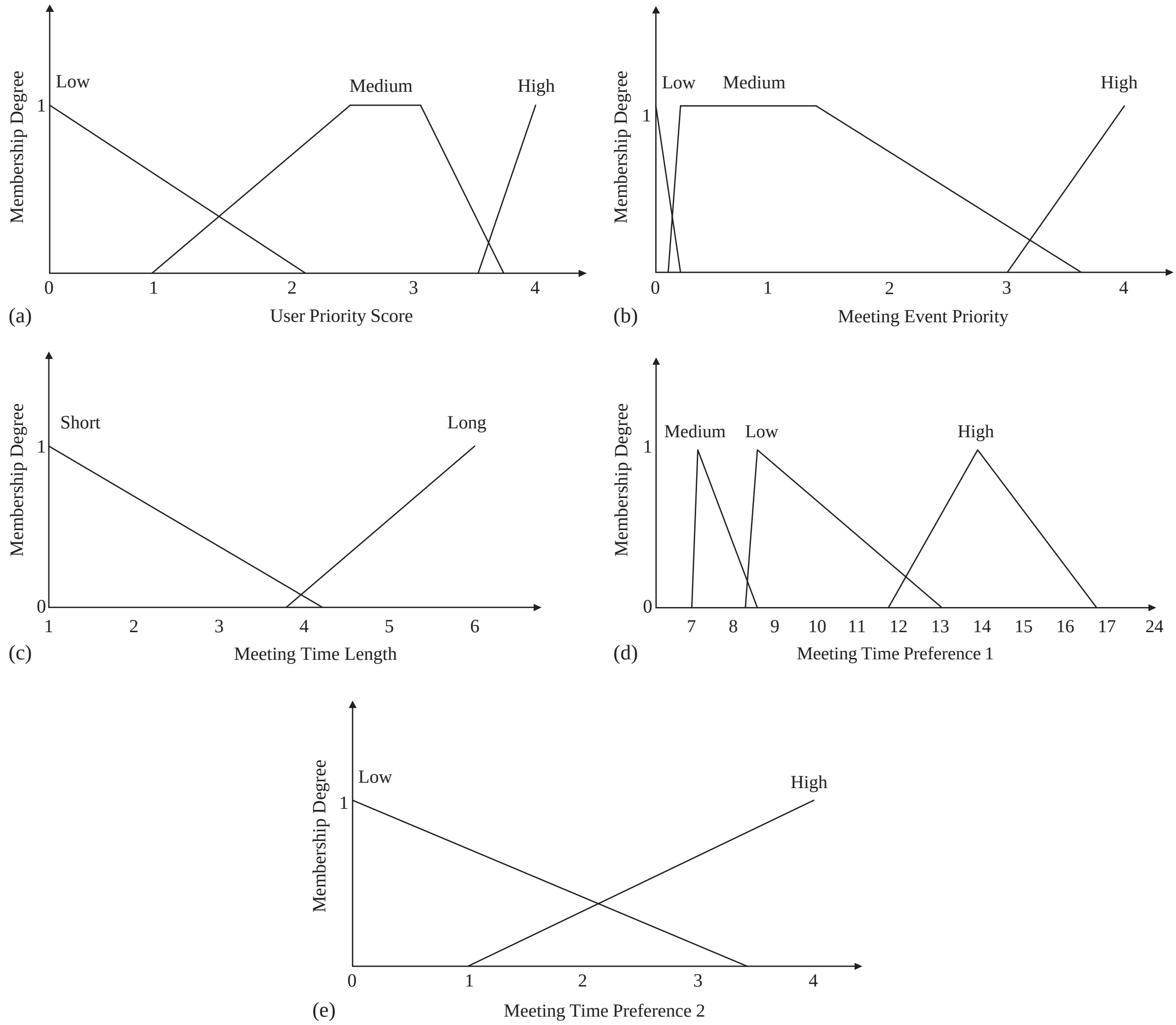


Fig. 11. (a) Fuzzy sets for UP fuzzy variable after learning. (b) Fuzzy sets for MEP fuzzy variable after learning. (c) Fuzzy sets for MTL fuzzy variable after learning. (d) Fuzzy sets for MTP1 fuzzy variable after learning. (e) Fuzzy sets for MTP2 fuzzy variable after learning.

In Fig. 5, $X_{i1}, \ldots,$ and X_{im} denote the input fuzzy variables and Y_i denotes the output fuzzy variable for invitee i. The encoding approach for the fuzzy variables is as the following equations:

$$X_{i1} = \{[l_{i1L1}, h_{1i1L1}, h_{2i1L1}, r_{i1L1}], [l_{i1L2}, h_{1i1L2}, h_{2i1L2}, r_{i1L2}], \ldots, [l_{i1LN_1}, h_{1i1LN_1}, h_{2i1LN_1}, r_{i1LN_1}]\}, \tag{8}$$

$$\vdots$$

$$X_{im} = \{[l_{imL1}, h_{1imL1}, h_{2imL1}, r_{imL1}], [l_{imL2}, h_{1imL2}, h_{2imL2}, r_{imL2}], \ldots, [l_{imLN_m}, h_{1imLN_m}, h_{2imLN_m}, r_{imLN_m}]\}, \tag{9}$$

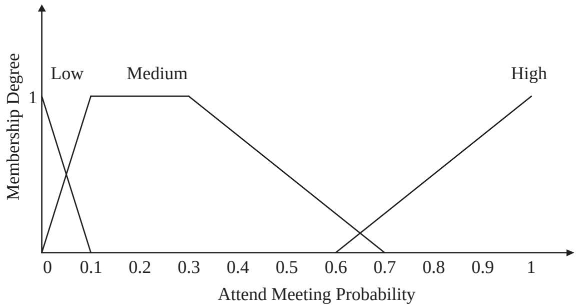


Fig. 12. Fuzzy sets for AMP fuzzy variable after learning.

$$Y_i = \{[l_{iY1}, h_{1iY1}, h_{2iY1}, r_{iY1}], [l_{iY2}, h_{1iY2}, h_{2iY2}, r_{iY2}], \ldots, [l_{iYN_y}, h_{1iYN_y}, h_{2iYN_y}, r_{iYN_y}]\}. \tag{10}$$

Fig. 6 shows a graphic representation of a specific fuzzy variable X_{im} with three linguistic terms $[l_{imL1}, h_{1imL1}, h_{2imL1}, r_{imL1}]$, $[l_{imL2}, h_{1imL2}, h_{2imL2}, r_{imL2}]$, and $[l_{imL3}, h_{1imL3}, h_{2imL3}, r_{imL3}]$.

In this case, we define the encoding restrictions for the fuzzy variable as follows:

$$\begin{aligned} &l_{imL1} \leqslant h_{1imL1} \leqslant h_{2imL1} \leqslant r_{imL1},\ l_{imL2} \leqslant h_{1imL2} \leqslant h_{2imL2} \leqslant r_{imL2}, l_{imL3} \leqslant h_{1imL3} \\ &\leqslant h_{2imL3} \leqslant r_{imL3}\ h_{2imL1} < l_{imL2} \leqslant r_{imL1} < h_{1imL2},\ h_{2imL2} < l_{imL3} \leqslant r_{imL2} < h_{1imL3}. \end{aligned} \tag{11}$$

The restrictions are used to preserve meaningful fuzzy sets. In contrast to the variation intervals proposed by Cordon et al. [3,5,6], the restrictions make the tuning range larger to improve the learning efficiency for GLA. In fact, the above restrictions can work effectively for tuning the membership functions of UP, MEP, MTL, MTP2 and AMP, but they are not suitable to tune the membership functions of MTP1. The meeting time preference is various for the invitees, it may be in the morning, noon, or afternoon. Hence the linguistic term *High* of MTP1 may be located at different time slot for different invitee. In this paper, we propose a dynamic restriction algorithm that can solve this problem. The Dynamic Restriction Algorithm for the fuzzy variable MTP1 is as follows (Fig. 7):

Dynamic Restriction Algorithm
Set the dynamic restrictions of fuzzy variable MTP1 for genetic tuning.
Input:
The chromosome from the gene population.
Output:
The restriction set of the linguistic terms for MTP1.
Method:
Step 1: Retrieve the parameters of MTP1 from the chromosome.
  Step 1.1: $[P_{High\_l}, P_{High\_hl}, P_{High\_hr}, P_{High\_r}] \leftarrow [l_{High}, h_{1High}, h_{2High}, r_{High}]$.
  Step 1.2: $[P_{Medium\_l}, P_{Medium\_hl}, P_{Medium\_hr}, P_{Medium\_r}] \leftarrow [l_{Medium}, h_{1Medium}, h_{2Medium}, r_{Medium}]$.
  Step 1.3: $[P_{Low\_l}, P_{Low\_hl}, P_{Low\_hr}, P_{Low\_r}] \leftarrow [l_{Low}, h_{1Low}, h_{2Low}, r_{Low}]$.
Step 2: Sort the positions of the linguistic term's peak values $(P_{High\_hl} + P_{High\_hr})/2$, $(P_{Medium\_hl} + P_{Medium\_hr})/2$, and $(P_{Low\_hl} + P_{Low\_hr})/2$.
  Step 2.1: *min* $\leftarrow$ the linguistic term with the minimum peak value.
  Step 2.2: *med* $\leftarrow$ the linguistic term with the medium peak value.

Table 2
Fuzzy Rules After Learning for invitee i (L:Low, M:Medium, H:High, Lg:Long, S:Short)

Rule	UP	MEP	MTL	MTP(1)	MTP(2)	AMP
5	very L	m-or-l L	m-or-l Lg	m-or-l H	L	very L
14	very L	m-or-l M	m-or-l Lg	very L	m-or-l H	m-or-l L
19	very L	M	S	m-or-l L	m-or-l L	L
26	m-or-l L	H	very Lg	m-or-l L	very H	very M
35	very L	very H	very S	m-or-l H	very L	M
47	m-or-l M	m-or-l L	m-or-l S	H	very L	m-or-l M
48	very M	L	m-or-l S	m-or-l H	very H	very H
59	m-or-l M	m-or-l M	S	H	m-or-l L	m-or-l H
70	very M	m-or-l H	very S	m-or-l M	m-or-l H	H
90	m-or-l H	m-or-l M	Lg	very H	very H	H

Step 2.3: $Max \leftarrow$ the linguistic term with the maximum peak value.
Step 3: Generate the dynamic restrictions as follows:
Step 3.1: Generate the restriction $P_{min\_l} \leqslant P_{min\_hl} \leqslant P_{min\_hr} \leqslant P_{min\_r}$
Step 3.2: Generate the restriction $P_{med\_l} \leqslant P_{med\_hl} \leqslant P_{med\_hr} \leqslant P_{med\_r}$
Step 3.3: Generate the restriction $P_{max\_l} \leqslant P_{max\_hl} \leqslant P_{max\_hr} \leqslant P_{max\_r}$
Step 3.4: Generate the restriction $P_{min\_hr} \leqslant P_{med\_l} < P_{min\_r} \leqslant P_{med\_hl}$
Step 3.5: Generate the restriction $P_{med\_hr} \leqslant P_{max\_l} < P_{med\_r} \leqslant P_{max\_hl}$
Step 4: End.

Next, the encoding approach for PRB will be described. A linguistic modifier used in PRB is a function with the parameter δ that lets us alter the membership functions for the fuzzy sets associated

Table 3
The training data set with the AMP before and after learning

The training data					Desired output	Original results	Cordon results	GLA (0.9/0.05) results	GLA (0.6/0.05) results	GLA (0.9/0.1) results	GLA (0.6/0.1) results
UP	MEP	MTL	MTP1	MTP2	AMP	AMP	AMP	AMP	AMP	AMP	AMP
3	3	2	15	4	1	0.9983	0.9809	1	0.9858	0.998	0.9999
2	0	2	13	3	1	0.4996	0.3988	0.9015	0.9717	0.9999	0.9906
3	1	3	12	3	1	0.958	0.9387	1	0.9791	0.9992	0.9988
0	3	3	12	4	1	0.8032	0.9897	0.9185	0.9693	0.9987	0.9836
1	2	4	10	4	0	0.5	0.2889	0	0.0125	0	0.0004
3	3	5	9	3	0	0.9186	0.4164	0	0.0386	0	0
3	2	4	13	4	1	0.8032	0.9402	1	0.9445	0.9974	0.9977
1	2	2	15	4	1	0.9097	0.9735	1	0.9846	0.9989	0.9993
1	2	4	11	4	0	0.4991	0.2955	0	0.0071	0	0
3	2	5	8	4	0	0.8032	0.2615	0	0.0912	0	0.1234
0	1	1	16	4	1	0.5002	0.416	1	0.9753	0.8273	0.9991
4	4	3	14	3	1	0.9978	0.992	1	0.9697	0.9978	0.9996
1	0	4	11	3	0	0.5	0.3319	0	0	0	0
1	1	2	14	3	1	0.7856	0.886	1	0.9426	0.9993	0.9917
3	3	4	12	3	1	0.9193	0.9641	1	0.9237	0.9978	0.9983

Table 4
The test data set with the AMP before and after learning

The test data					Desired output	Original results	Cordon results	GLA (0.9/0.05) results	GLA (0.6/0.05) results	GLA (0.9/0.1) results	GLA (0.6/0.1) results
UP	MEP	MTL	MTP1	MTP2	AMP	AMP	AMP	AMP	AMP	AMP	AMP
1	0	2	8	3	0	0.9031	0.9754	0	0.1022	0	0.0426
1	2	3	13	4	1	0.9033	0.9932	0.9949	0.9788	0.9978	0.9989
1	0	1	12	4	1	0.5	0.3795	0.8833	0.1229	0.9962	0.2895
4	3	2	15	3	1	0.9986	0.9708	1	0.8702	0.9986	0.8407
3	3	5	9	4	0	0.9191	0.3976	0	0.0967	0	0.1862

Table 5
The mean square error values of the training and test data

	MSE_{tra}	MSE_{tst}
Original results	0.09593744	0.19196871
Cordon [9] results	0.04160218	0.14954099
GLA (0.9/0.05) results	0.000544817	0.00136449
GLA (0.6/0.05) results	0.000878598	0.080639765
GLA (0.9/0.1) results	0.000994992	0.00000212398
GLA (0.6/0.1) results	0.000526634	0.056667317

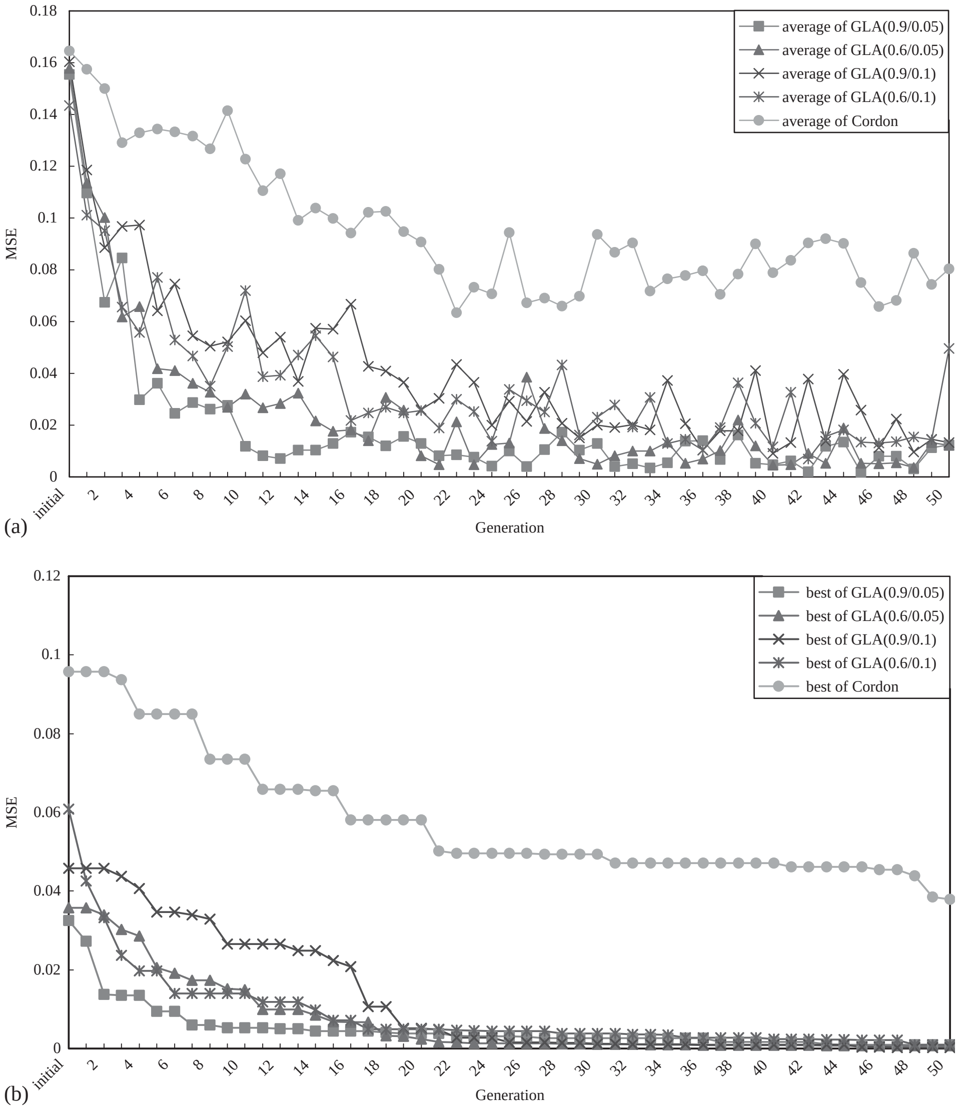


Fig. 13. (a) The convergence curves of the population for the 15 times meeting behavior learning. (b) The convergence curves of the elitist chromosome for the 15 times meeting behavior learning.

to the linguistic term. Two of the most well known modifiers are the *concentration linguistic modifier* "very" ($\delta = 2$) and *dilation linguistic modifier* "more-or-less" ($\delta = 0.5$) [3,5]. Eqs. (12) and (13) denote the functions of the two modifiers used in this paper:

$$u^{\text{very}}(x) = (u(x))^2, \tag{12}$$

$$u^{\text{more-or-less}}(x) = (u(x))^{0.5}. \tag{13}$$

Fig. 8 (a)–(c) show the graphic representation of the specific linguistic terms *normal High* ($\delta = 1$), *very High* ($\delta = 2$) and *more-or-less High* ($\delta = 0.5$), respectively.

Table 6
The training data set with the AMP before and after learning

The training data					Desired output	Original results	Cordon results	GLA (0.9/0.05) results	GLA (0.6/0.05) results	GLA (0.9/0.1) results	GLA (0.6/0.1) results
UP	MEP	MTL	MTP1	MTP2	AMP	AMP	AMP	AMP	AMP	AMP	AMP
3	3	2	15	4	1	0.9997	0.9633	1	0.9667	1	0.9985
2	0	2	13	3	1	0.4996	0.4843	1	0.9656	0.9117	0.8734
3	1	3	12	3	1	0.9989	0.9099	0.9997	0.9518	0.9999	0.9965
0	3	3	12	4	1	0.9997	0.971	0.8295	0.9666	1	0.9825
1	2	4	10	4	0	0.4993	0.5083	0	0.058	0.004	0.0604
3	3	5	9	3	0	0.9995	0.5308	0	0	0	0
3	2	4	13	4	1	0.9998	0.7465	1	0.9635	0.9823	0.9998
1	2	2	15	4	1	0.4993	0.9589	1	0.9612	1	1
1	2	4	11	4	0	0.5015	0	0	0.0635	0.0041	0.0907
3	2	5	8	4	0	0.9998	0	0	0	0	0
0	1	1	16	4	1	0.5	0.4987	0.9903	0.9646	1	1
4	4	3	14	3	1	1	0.994	1	0.9663	1	1
1	0	4	11	3	0	0.5	0	0	0.0142	0.0099	0.0181
1	1	2	14	3	1	0.5011	0.9338	1	0.953	1	0.9942
3	3	4	12	3	1	0.9989	0.9811	0.9958	0.9592	1	0.9823

Table 7
The test data set with the AMP before and after learning

The test data					Desired output	Original results	Cordon results	GLA (0.9/0.05) results	GLA (0.6/0.05) results	GLA (0.9/0.1) results	GLA (0.6/0.1) results
UP	MEP	MTL	MTP1	MTP2	AMP	AMP	AMP	AMP	AMP	AMP	AMP
1	0	2	8	3	0	0.9031	0	0	0	0	0
1	2	3	13	4	1	0.9033	0.8878	1	0.9567	0.9083	0.996
1	0	1	12	4	1	0.5	0.494	0.8296	0.9519	0.9096	0.8547
4	3	2	15	3	1	0.9986	0.9845	1	0.9666	1	1
3	3	5	9	4	0	0.9191	0.5936	0	0	0	0

Eq. (14) shows the CS_b encoding approach for invitee i. In this case, the parameter δ^r_{iXm} denotes the linguistic modifier of mth input fuzzy variable for rth fuzzy inference rule. Similarly, the parameter δ^r_{iY} denotes the linguistic modifier of output fuzzy variable for rth fuzzy inference rule.

$$\{[\delta^1_{iX1}, \delta^1_{iX2}, \ldots, \delta^1_{iXm}, \delta^1_{iY}], \ldots, [\delta^r_{iX1}, \delta^r_{iX2}, \ldots, \delta^r_{iXm}, \delta^r_{iY}]\}. \tag{14}$$

The initial population for the gene pool is composed of four groups with the same number of CS_a part and CS_b part. The first group is composed of the original CS_a part defined in Fig. 3 and CS_b part with the unit modifier ($\delta = 1$). The second group is composed of the original CS_a part defined in Fig. 3 and randomized CS_b part. The third group is composed of randomized CS_a part and unit

Table 8
The mean square error values of the training and test data

	MSE_{tra}	MSE_{tst}
Original results	0.12498025	0.19196871
Cordon [9] results	0.037946623	0.062122606
GLA (0.9/0.05) results	0.000972735	0.002903617
GLA (0.6/0.05) results	0.000747637	0.000530406
GLA (0.9/0.1) results	0.0002747	0.001658105
GLA (0.6/0.1) results	0.000963251	0.002112808

modifier CS_b part ($\delta = 1$). The fourth group is composed of randomized CS_a part and randomized CS_a part. Next, we describe the genetic operators as follows.

Fig. 9 shows the flow chart of GLA. The GLA initiates the population by the encoding schema and restrictions, then records the initiation population as the current population. The chromosome in the current population is evaluated by the fitness function. If the evaluation does not satisfy the fitness function, then GLA processes the generation. The elitism is used in this process. In the beginning of selection, the best two chromosomes in the current population are selected to the new population without crossover and mutation. The remaining chromosomes in the new population are selected by the Roulette Wheel Selection mechanism [12]. It is one of the most common techniques being used for a proportionate selection mechanism. The new population will replace current population when the two population numbers are equal.

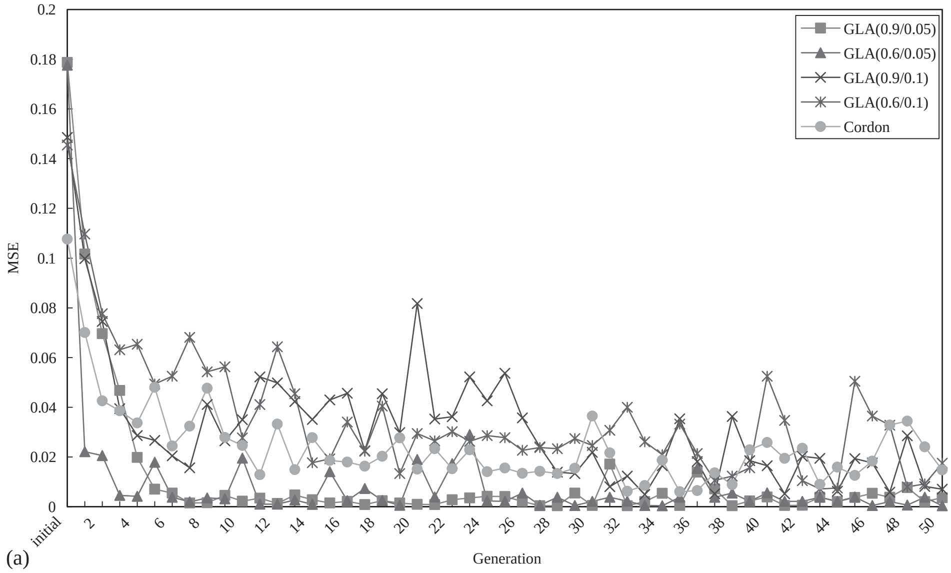
After the elitism selection, two other fundamental operators, crossover and mutation, are introduced below. The one-point crossover method is adopted and a crossover point is randomly set. The portions of the two chromosomes beyond this cut-off point to the right are to be exchanged to form the offspring. An operation rate with a typical value between 0.6 and 1.0 is normally used as the probability of crossover [12]. The mutation process is applied to each offspring individually after the crossover exercise. It alters each gene randomly with a typical probability value of less than 0.1. In this paper, the CS_a part and CS_b part are mutated in the restriction of Eq. (8) and the set $\{0.5, 1, 2\}$, respectively. The probability parameters of crossover and mutation are critically dependent upon the nature of the objective function. An objective function is a measuring mechanism that is used to evaluate the status of chromosome. The objective (fitness) function used here is to minimize the *mean square error* (MSE) as follows:

$$MSE = \frac{1}{2 \times T} \sum_{t=1}^{T} (y_{it} - y_{itd})^2, \tag{15}$$

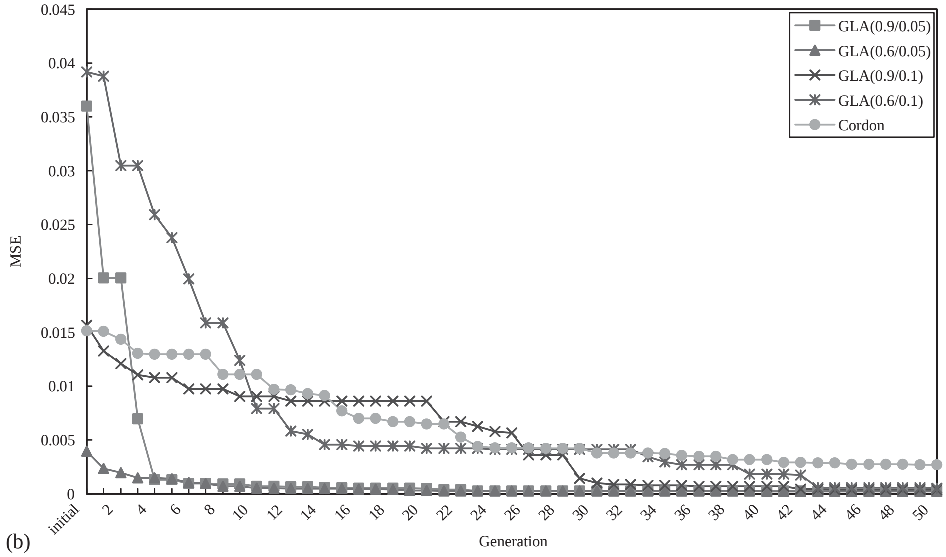
where T denotes the number of the training data for invitee i, y_{it} denotes the output of FIA for the tth training data, and y_{iid} denotes the *desired output* of the tth training data for invitee i.

4. Experimental results

We have constructed an experimental website with intelligent fuzzy agent for meeting scheduling decision support system at the E-Commerce Lab. of Chang Jung University, Tainan, Taiwan,



(a)



(b)

Fig. 14. (a) The convergence curves of the population for the 15 times meeting behavior learning. (b) The convergence curves of the elitist chromosome for the 15 times meeting behavior learning.

to verify the performance of the agent. This intelligent fuzzy agent has been used to support the meeting scheduling for the research group of E-Commerce Lab. In this paper, we retrieve a member of E-Commerce Lab. with 20 times meeting information from MIKB and PKB, then adopt 15 and 5 times of meeting information to be the training and testing data, respectively. The first experiment adopt the trapezoidal functions to be the membership functions of the fuzzy variables. The parameter for UP, MEP, MTL, MTP1, MTP2 and AMP are $\{[0, 0, 0, 1.5],\ [0.5, 1.5, 2.5, 3.5],\ [2.5, 4, 4, 4]\}$,

Table 9
The training data set with the AMP before and after learning

The training data					Desired output	Original results	Cordon results	GLA (0.9/0.05) results	GLA (0.6/0.05) results	GLA (0.9/0.1) results	GLA (0.6/0.1) results
UP	MEP	MTL	MTP1	MTP2	AMP	AMP	AMP	AMP	AMP	AMP	AMP
3	3	2	15	4	1	0.9997	1	1	1	1	1
2	0	2	13	3	1	0.4996	1	0.9991	0.9921	0.9501	0.9996
3	1	3	12	3	1	0.9989	0.9872	0.9986	0.9984	1	1
0	3	3	12	4	1	0.9997	1	1	1	0.9231	0.9968
1	2	4	10	4	1	0.4993	1	1	1	1	1
3	3	5	9	3	1	0.9995	0.9821	1	0.9995	1	0.9984
3	2	4	13	4	1	0.9998	1	1	1	1	1
1	2	2	15	4	1	0.4993	1	1	1	1	1
1	2	4	11	4	1	0.5015	1	1	1	1	1
3	2	5	8	4	1	0.9998	0.9698	1	0.9981	0.9953	0.999
0	1	1	16	4	1	0.5	0.9971	0.9127	1	0.9445	0.8752
4	4	3	14	3	1	1	0.9999	1	1	1	1
1	0	4	11	3	1	0.5	1	1	0.9165	0.9791	0.9997
1	1	2	14	3	1	0.5011	1	1	0.9942	1	1
3	3	4	12	3	1	0.9989	0.719	1	0.9997	1	0.9998

Table 10
The test data set with the AMP before and after learning

The test data					Desired output	Original results	Cordon results	GLA (0.9/0.05) results	GLA (0.6/0.05) results	GLA (0.9/0.1) results	GLA (0.6/0.1) results
UP	MEP	MTL	MTP1	MTP2	AMP	AMP	AMP	AMP	AMP	AMP	AMP
1	0	2	8	3	1	0.4983	0.9999	0.8854	0.8117	1	0.8745
1	2	3	13	4	1	0.5011	0.9972	1	1	1	1
1	0	1	12	4	1	0.4993	0.6676	1	0.953	0.9439	1
4	3	2	15	3	1	0.9998	1	1	1	0.9409	1
3	3	5	9	4	1	0.9998	1	1	1	1	0.9984

$\{[0,0,0,1.5],\ [0.5,1.5,2.5,3.5],\ [2.5,4,4,4]\}$, $\{[1,1,1,5],\ [3,6,6,6]\}$, $\{[7,10,10,12],\ [11,14,14,16],\ [15,16,16,17]\}$, $\{[0,0,0,3],\ [1,4,4,4]\}$, and $\{[0,0,0,0.3],\ [0.2,0.4,0.6,0.8],\ [0.7,1,1,1]\}$, respectively. The compared learning approach is proposed by Cordon [6]. Fig. 10(a) and (b) show the compared *MSE* curves of Cordon [6] and *GLA* with various parameters. In the figures, the first parameter is crossover probability such as 0.9 or 0.6, and the second parameter is mutation probability such as 0.05 or 0.1. In addition, Fig. 10(a) shows the convergence curves of the population for the 15 times meeting behavior learning, and Fig. 10(b) shows the convergence curves of the elitist chromosome for the 15 times meeting behavior learning. By the results, we can see that *GLA* converges faster than Cordon's algorithm.

Table 11
The mean square error values of the training and test data

	MSE_{tra}	MSE_{tst}
Original results	0.058306918	0.07513048
Cordon [9] results	0.002678857	0.011049763
GLA (0.9/0.05) results	0.000254135	0.001313316
GLA (0.6/0.05) results	0.000235827	0.00376659
GLA (0.9/0.1) results	0.000398092	0.00066402
GLA (0.6/0.1) results	0.000519638	0.001575195

Fig. 11 shows the learned input fuzzy variables of PKB for the specific member of E-Commerce Lab. by the 15 times training data. Fig. 12 shows the learned output fuzzy variable. Table 2 presents some of the 108 learned fuzzy rules.

Tables 3 and 4 show the computing results of training data and testing data by the two learning approaches, respectively. Finally, Table 5 shows the *MSE* values of the two approaches. By the experimental results, GLA can achieve lower *MSE* values than Cordon's algorithm.

In the second experiment, we utilize the triangular functions to be the membership functions of the fuzzy variables and compare with Cordon's method. The parameters of the fuzzy variables for UP, MEP, MTL, MTP1, MTP2, and AMP are $\{[0,0,2], [1,2,3], [2,4,4]\}$, $\{[0,0,2], [1,2,3], [2,4,4]\}$, $\{[1,1,5], [3,6,6]\}$, $\{[7,10,12], [11,14,16], [15,16,17]\}$, $\{[0,0,3], [1,4,4]\}$, and $\{[0,0,0.3], [0.2,0.5,0.8], [0.7,1,1]\}$, respectively. There are two different training data sets used in this experiment. First, the training data set with different desired outputs contains $AMP = 0$ and 1 are made. Fig. 13(a) shows the convergence curves of the population for the 15 times meeting behavior learning, and Fig. 13(b) shows the convergence curves of the elitist chromosome for the 15 times meeting behavior learning. Tables 6 and 7 show the computing results of training data and testing data by the two learning approaches, respectively. Table 8 shows the MSE values of the two approaches. Next, we test the performance of GLA and Cordon Algorithm with the same desired output $AMP = 1$. Fig. 14(a) shows the convergence curves of the population for the 15 times meeting behavior learning, and Fig. 14(b) shows the convergence curves of the elitist chromosome for the 15 times meeting behavior learning. Tables 9 and 10 show the computing results of training data and testing data by the two learning approaches, respectively. Table 11 shows the *MSE* values of the two approaches.

5. Conclusions

An Intelligent Fuzzy Agent (IFA) for Meeting Scheduling Decision Support System is proposed in this paper. IFA contains three subagents including MNA, FIA, and GLA to perform the intelligent meeting scheduling support task. Moreover, an experimental website at Chang Jung University to evaluate the proposed approach is also constructed. The experimental results exhibit that the IFA can infer the suitable meeting time successfully. In the future, we will adopt various genetic learning approaches for more efficient and effective personal behavior learning, and try to propose an automatic fuzzy inference rule generating mechanism.

Acknowledgements

The authors would like to thank the anonymous referees for their constructive and useful comments. This work is partially supported by the National Science Council of Republic of China under grants NSC-91-2213-E-309-005, and partially supported by Walsin Lihwa CREDIT Research Center.

References

[1] A. Ashir, K. Hyoun Joo, T. Kinoshita, N. Shiratori, Multi-agent based decision mechanism for distributed meeting scheduling system, Proc. Internat. Conf. on Parallel and Distributed Systems, Korea, 1997, pp. 275–280.
[2] F. Bergenti, A. Poggi, An agent-based approach to manage negotiation protocols in flexible CSCW systems, Proc. 4th Internat. Conf. on Autonomous Agents, Barcelona Spain, 2000, pp. 267–268.
[3] J. Casillas, O. Cordon, F. Herrera, M.J. Del Jesus, Genetic tuning of fuzzy rule-based systems integrating linguistic hedges, 2001 IFSA World Congress and 20th NAFIPS Internat. Conf., Canada, 2001, pp. 1570–1574.
[4] H. Chen, J. Yen, Toward intelligent meeting agents, IEEE Comput. 29(8) (1996) 62–70.
[5] O. Cordon, M.J. Del Jesus, F. Herrera, Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods, Internat. J. Intelligent System 13 (1998) 1025–1053.
[6] O. Cordon, F. Herrera, P. Villar, Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base, IEEE Trans. Fuzzy Systems 9 (4) (2001) 667–674.
[7] J. Ferber, Multi-Agent Systems, Addison-Wesley, New York, 1999.
[8] T. Haynes, S. Sen, N. Arora, R. Nadella, An automated meeting scheduling system that utilizes user preferences, Proc. 1st Internat. Conf. on Autonomous Agents, USA, February 1997, pp. 308–315.
[9] W.S. Jeong, J.S. Yun, G.S. Jo, Cooperation in multi-agent system for meeting scheduling, IEEE Tencon 99 Proc. of the IEEE Region 10 conf., Volume 2, Korea, pp. 832–835.
[10] C.-S. Lee, C.-Y. Pan, M.-J. Chang, A fuzzy decision agent for meeting scheduling supported system, 2002 Internat. Conf. on Fuzzy Systems and Knowledge Discovery, Singapore, November 2002.
[11] C.T. Lin, C.S.G. Lee, Neural-network-based fuzzy logic control and decision system, IEEE Trans. Comput. 40 (12) (1991) 1320–1336.
[12] K.F. Man, K.S. Tang, S. Kwong, Genetic Algorithm, Springer, Hong Kong, 1999.
[13] E. Mynatt, J. Tullio, Inferring calendar event attendance, Proc. 6th Internat. Conf. on Intelligent User Interfaces, USA, January 2001, pp. 121–128.
[14] S. Sen, Developing an automated distributed meeting scheduler, IEEE Expert 12(4) (1997) 41–45.
[15] K. Sugihara, T. Kikuno, N. Yoshida, A meeting scheduler for office automation, IEEE Trans. Software Eng. 15 (10) (1989) 1141–1146.