Theory and Properties of a Selfish Protocol for Multi-Agent Meeting Scheduling Using Fuzzy Constraints

Xudong Luo, Ho-fung Leung and Jimmy Ho-man Lee [1]

Abstract. This paper develops an agent-based methodology for meeting scheduling. In such a multi-agent system, each agent acts on behalf of a user. For each user the meeting scheduling problem is modeled by a fuzzy constraint satisfaction problem, and an appointment is made by negotiations among agents. A negotiation procedure concerns with two key components: the protocol for organizing negotiations among agents, and the operator for fusing agents' individual evaluations for a feasible time slot. In particular, we propose a kind of selfish protocol, and present an axiomatic framework for fusion operators. In addition, a meeting scheduling example is used to illustrate the proposed methodology.

keywords: Scheduling, Constraint Satisfaction, Multi-Agent Systems, Distributed AI, Uncertainty in AI.

1 Introduction

A meeting scheduling task usually requires a lot of efforts in communication and negotiation among attendants since they may have different timetables, constraints and preferences. If the task is done manually, a great deal of human resources have to be poured in. Unfortunately, the result may still be unsatisfactory, especially in the case where a meeting involves a lot of attendants and constraints. Since such tasks always follow similar routines in their decision making processes, it is possible to develop computer systems for this kind of tasks with manual involvement as little as possible. That is, attendants simply need to feed their timetable, constraints and preferences into a computer system, and then the system automatically makes an appointment among attendants for a meeting.

Initial meeting scheduling systems usually used centralized approaches, in which all users' information are collected and processed in batch mode. Recent systems adopt an agent-based approach since agents [1], 1) allow users to focus on more productive tasks, and can solve the problem without users' guidance; 2) can accomplish tasks through cooperation among agents; 3) can improve the quality of information processing by preventing errors perhaps due to the tedious nature of such tasks; 4) can take into account any change of any agent's need dynamically; and 5) allow users to keep their privacies.

In our multi-agent system for meeting scheduling, each agent can act on behalf of a user and hold the user's information necessary for scheduling meetings, *e.g.* available time slots, constraints and preferences. Such pieces of information are modeled by fuzzy constraint satisfaction problems (FCSPs) [13, 5]. When a user wants to host a meeting with other users, the user just needs to run the corresponding agent, which negotiates with the agents acting on behalf of other concerned users. During the negotiation procedure, there are two key components: 1) a protocol used to organize the negotiation, and 2) a fusion operator used to aggregate all agents' individual evaluations for a feasible time slot for a meeting. Main characteristic of our protocol is that during negotiation each agent tries to maximize its interest, and so we call the protocol selfish protocol.

Although there have been some works which link meeting schedule problems to constraint satisfaction problems and multi-agent systems, they are different from ours here. First, the issue of fusing agents' individual evaluations for a feasible time slot is almost ignored. Instead, this paper addresses the issue. In fact, we suggests an axiomatic framework for fusion operations, and discuss their construction. Although a sort of fusion operator is also involved in [14, 6], they are just some specific operators rather than an axiomatic framework. In addition, unlike our setting, they do not put weights of agents into account when fusing. Second, according to [18] the constraints techniques are necessary for this sort of problems, but not many researchers handle this sort of problems by using constraint techniques, especially by fuzzy constraint techniques [16, 13, 5]. For example, [6, 7, 2, 14, 8] do not handle the problems by constraint techniques. In [18, 17], the constraint techniques rather than fuzzy constraint techniques are used. While this paper uses fuzzy constraint techniques to handle the problem. The conventional constraint techniques provides an elegant way to formulate problems with hard constraints which can never be violated. However, in real-life, this is sometimes inflexible. Thus, various efforts have been made for equipping conventional constraint technique with soft constraints which can partially be violated. One of them is introducing the concept of fuzzy constraint satisfaction [16, 13, 5]. Third, our protocol is different from the previous ones (*e.g.* in [17, 2]) mainly in that through ours an optimum appointment can be made but not through theirs. In [6], Garrido $et\ al.$ just implements a simplified version of the protocol presented by Sycara $et\ al.$ in [17].

The rest of this paper is organized as follows. Section 2 recalls concepts related to FCSPs. Section 3 defines the basic concepts and terms involved in multi-agent system for meeting scheduling. Section 4 outlines a selfish protocol for making an appointment among agents, and studies the properties of the protocol. Section 5 suggests an axiomatic framework for fusion operations on agents' individual evaluations for a meeting proposal, and studies the construction of fusion operators. Section 6 illustrates our methodology with a meeting scheduling problem. The last section summarizes our main contributions and sheds light on some future research.

2 Preliminaries

This section recalls some basic concepts of FCSPs.

[1] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, P.R. China, Email: {xluo,lhf,jlee}@cse.cuhk.edu.hk

Definition 1 *A* **fuzzy constraint satisfaction** *problem (FCSP) is defined as a 3-tuple* (X, D, C^f)*, where*

1) $X = \{x_i | i = 1, \cdots, n\}$ *is a finite set of variables;*
2) $D = \{d_i | d_i$ *is the domain on which the variable* x_i *takes values,* $i = 1, \cdots, n\}$ *is a finite set of all domains associated with each variable in* X*; and*
3) C^f *is a set of fuzzy constraints:*

$$C^f = \left\{ R_i^f | \mu_{R_i^f} : \left(\prod_{x_j \in var(R_i^f)} d_j \right) \rightarrow [0,1], i = 1, \cdots, m \right\}. \tag{1}$$

where $var(R_i^f)$ *denotes the set of variables of* R_i^f.

Clearly, in an FCSP (X, D, C^f), each constraint $R^f \in C^f$ is a fuzzy relation among the variables in the subset $var(R^f)$ of X. If each constraint is a crisp relation among the variables, namely its membership function (or called **characteristic function**) takes values only on $\{0, 1\}$, then the FCSP degenerates to a **constraint satisfaction problem (CSP)**.

Definition 2 *The assignment of value* v *to a variable* x*, denoted as* v_x*, is said to be a* **label** *of the variable. A* **compound label** $v_{X'}$ *of all variables in set* $X' = \{x'_1, \cdots, x'_m\} \subseteq X$ *is a simultaneous assignment of values to all variables in set* X'*, that is,*

$$v_{X'} = (v_{x'_1}, \cdots, v_{x'_m}). \tag{2}$$

Given a compound label, the membership degree of a fuzzy constraint tells just a local degree to which the constraint is individually satisfied by the label. Naturally, we would like to know the *global degree* to which all constraint are satisfied with a compound label.

Definition 3 *In an FCSP* (X, D, C^f)*, given a compound label* $(v_{x_1}, \cdots, v_{x_n})$ *of all variables in* X*, the* **global satisfaction degree** *is defined as*

$$\alpha(v_{x_1}, \cdots, v_{x_n}) = \min\{\mu_{R^f}(v_{var(R^f)}) | R^f \in C^f\}. \tag{3}$$

A **solution** *of an FCSP* (X, D, C^f) *is a compound label* $(v_{x_1}, \cdots, v_{x_n})$ *of all variables in* X *such that*

$$\alpha(v_{x_1}, \cdots, v_{x_n}) \geq \alpha_0. \tag{4}$$

Here α_0 *is called the threshold for solutions.*

Generally speaking, people are interested in finding out the degree to which a compound label satisfies *all the constraints* in an FCSP. Thus, the operator min is used in (3). Generally, in an FCSP min can be replaced by a T-norm. Corresponding to a T-norm is a Triangular conorms (T-conorms), which we shall also use later. So, here we recall both of them briefly. A detailed description of T-norms and T-conorms can be found elsewhere [4].

Definition 4 *If an operator* $\circ : [0,1] \times [0,1] \longrightarrow [0,1]$ *satisfies:*

1) $a_1 \circ a_2 = a_2 \circ a_1$,
2) $(a_1 \circ a_2) \circ a_3 = a_1 \circ (a_2 \circ a_3)$,
3) *if* $a_1 \leq a_2$ *and* $a_3 \leq a_4$ *then* $a_1 \circ a_3 \leq a_2 \circ a_4$,
4) $a_1 \circ 1 = a_1$,

where $a_1, a_2, a_3, a_4 \in [0, 1]$*, then* $\circ$ *is said to be a Triangular norm (***T-norm***) on* $[0, 1]$*, denoted as* $\triangle$*. If* $\circ$ *satisfies 1), 2), 3) and*

4)$'$ *boundary:* $a_1 \circ 0 = a_1$,

then $\circ$ *is said to be a Triangular conorm (***T-conorm***) on* $[0, 1]$*, denoted as* $\triangledown$*.*

One of the important properties of $\triangle$ and $\triangledown$ is

Lemma 1 $\forall a_1, a_2 \in [0, 1]$,

$$a_1 \triangle a_2 \leq \min\{a_1, a_2\} \leq \max\{a_1, \cdots, a_n\} \leq \triangledown(a_1, a_2). \tag{5}$$

3 Basic Concepts

This section defines some basic concepts in our methodology.

The user's information about a meeting can be divided into three classes: a timetable, constraints and preferences. The latter two kinds of information can definitely be modeled by fuzzy constraints. The first kind can also be modeled by fuzzy constraints. In fact, a timetable can be represented by fuzzy constraint with one variable: 1) the more the user favors a time interval, the bigger the membership of the constraint when the variable takes the time interval as its value; 2) that the membership is 1 means the user feels fully satisfactory if the meeting can hold within the time interval; 3) that the membership is 0 means the user is not available within the time interval.

In order to solve a meeting scheduling problem, we introduce the concept of an FCSP multi-agent system as follows:

Definition 5 *An* **FCSP multi-agent system** *is a 4-tuple* $(\mathcal{A}, \oplus, \omega, \mathcal{P})$ *where*

1) $\mathcal{A} = \{(A_j, \eta_j) | A_j = (X, D, C_j^f)$ *is an FCSP,* η_j *is the threshold for solutions of* A_j*,* $j = 1, \cdots, n_{\mathcal{A}}\}$ *is the set of all agents, each of which is associated with an FCSP;*
2) $\oplus : [0,1] \times [0,1] \rightarrow [0,1]$ *is a fusion operator;*
3) $\omega : \{A_1, \cdots, A_{n_{\mathcal{A}}}\} \rightarrow [0, 1]$ *is a weight function, which assigns a weight to each agent;*
4) $\mathcal{P}$ *is a protocol used to organize a negotiation for a solution approval by all agents.*

Let us explain some terms which we will use in the following sections. In an FCSP multi-agent system, a **coordinator agent** is responsible for proposing a time slot, called a **proposal**, for a round of negotiation. At the end of a round of negotiation, if a proposal is feasible for each agent, it becomes an **appointment candidate** among agents. During a round of negotiation, the FCSP in each agent has an overall satisfaction degree to a proposal. This evaluation is called the agent's **individual evaluation** for the proposal. When a proposal becomes an appointment candidate, the corresponding coordinator agent uses a fusion operator to fuse all other agents' individual evaluations for the appointment candidate, as well as its own individual evaluation for the appointment candidate. The fused result is called the **overall evaluation** for the appointment candidate. When the negotiation procedure is finished, one of the appointment candidates is promoted to become the **appointment** among all agents.

4 A Protocol for Meeting Scheduling

This section gives a protocol for organizing an appointment for a meeting, and discusses basic properties of the protocol.

4.1 Selfish Protocol

The basic idea of the protocol is as follows: 1) In one round of negotiation, the coordinator agent first proposes a proposal, and then other agents check the proposal with their own timetables, constraints and preferences. If the proposal cannot be accepted by all other agents, the coordinator agent proposes another proposal. The procedure continues until a proposal is accepted by all agents or the coordinator agent cannot propose any more proposal. In the latter case the procedure terminates and no appointment can be made among agents. In the former case the proposal becomes an appointment candidate. 2) Each agent, in parallel, plays a role of the coordinator to organize a round of negotiation to find an appointment candidate. 3) The appointment candidate with the highest evaluation among all appointment candidates is promoted to become the appointment among all agents. During negotiation each agent tries to maximize its own interest, and so the protocol is called a **selfish protocol**.

The protocol consists of the following steps:

0) In parallel, each agent plays a role of the coordinator, denoted as $A_{coordinator}$, to organize a round of negotiation. Let $\lambda_{coordinator} = 1$.

1) **Initiating a round of negotiation**.

1.1) Based on the current value of $\lambda_{coordinator}$, we construct a CSP, in which everything is the same as the original FCSP of the coordinator agent but for each constraint R its characteristic function is given by

$$\mu'_R(v_{var(R)}) = \begin{cases} 1 & \text{if } \mu_R(v_{var(R)}) \geq \lambda_{coordinator}, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

1.2) For the above CSP, if the coordinator agent can find a solution, it sends the solution as a proposal to the relevant agents.

1.3) If the coordinator agent cannot get a solution to the above CSP, then set

$$\lambda_{coordinator} = \lambda_{coordinator} - \Delta\lambda_{coordinator}, \quad (7)$$

where

$$\begin{aligned} \Delta\lambda_{coordinator} = \min\{&\mu_{R'}(v_{var(R')}) - \mu_R(v_{var(R)}) \mid \\ &R', R \in C^f, v'_{var(R')} \subseteq v'_X, v_{var(R)} \subseteq v_X, \\ &\mu_{R'}(v'_{var(R')}) \neq \mu_R(v_{var(R)})\}. \end{aligned} \quad (8)$$

If $\lambda_{coordinator}$ is less than its threshold for solutions of the coordinator agent, then the protocol returns no solution and terminates; otherwise the protocol turns to step 1.1) again.

2) **Checking the proposal**. Each agent receiving the proposal, according to its constraints, evaluates the proposal.

2.1) The proposal is accepted by an agent if its evaluation for the proposal is greater than or equal to its threshold for solutions. In this case, the agent replies the coordinator agent with a message including its evaluation for the proposal.

2.2) The proposal is rejected by an agent if its evaluation for the proposal is less than its threshold for solutions. In this case, the agent notifies the coordinator agent. After receiving the notice, if the coordinator agent can find a new solution to the above CSP, it sends the new solution as a new proposal to the relevant agents, and the protocol turns to step 3); otherwise, by (7), the coordinator agent reduces the current value of $\lambda_{coordinator}$. Sequentially, if $\lambda_{coordinator}$ is less than the agent's threshold for solutions, then the protocol returns no solution and terminates, otherwise the protocol turns to step 1.1) again.

3) **Processing replies**. When a proposal is accepted by all agents, it becomes an appointment candidate. Then the coordinator agent calculates the overall evaluation for the appointment candidate by fusing all agents' individual evaluations for the candidate. The coordinator agent keeps the appointment candidate as well as the overall evaluation for the candidate.

4) **Making an appointment.** An appointment candidate with the highest overall evaluation is promoted to become the appointment among all agents.

4.2 Basic Property

Theorem 1 *If there is an appointment among agents, then the protocol must terminate and an appointment must be made. Moreover, the overall evaluation for the appointment is greater than or equal to the overall evaluation for any common time slot.*

Proof. In our multi-agent system for meeting scheduling, the number of agents is finite, and each agent organizes only one round of negotiation. So, in the procedure to make an appointment there are just finite rounds of negotiations. Since the FCSP in each agent just has a finite domain, the number of the solutions to the FCSP is finite. Hence, every round of negotiation will terminate. So, the protocol must terminate.

Since there is at least one solution accepted by all agents, there should be a solution v_X among solutions, which has the highest overall evaluation. Denote the agent, which gives the highest individual evaluation for the solution, as A_j. We can prove that the solution v_X can be found in the round of negotiation organized by A_j.

Let the individual evaluation of the coordinator agent A_j for the solution v_X be α_H, that is

$$\alpha_H = \min\{\mu_R(v_{var(R)}) | R \in C_j^f\}.$$

Thus,

$$\forall R \in C_j^f, \mu_R(v_{var(R)}) \geq \alpha_H. \quad (9)$$

1) Clearly, there is an integer n_H such that

$$1 - n_H \times \Delta\lambda_{coordinator} \leq \alpha_H \leq 1 - (n_H - 1) \times \Delta\lambda_{coordinator}, \quad (10)$$

where $\Delta\lambda_{coordinator}$ is given by (8). Let

$$\lambda_{coordinator} = 1 - n_H \times \Delta\lambda_{coordinator}. \quad (11)$$

Thus, by (10)

$$\alpha_H \geq \lambda_{coordinator}. \quad (12)$$

2) Based on the value of $\lambda_{coordinator}$ given by (11), we construct a CSP, in which everything is the same as the original FCSP of agent A_j but for each constraint R its characteristic function μ'_R is given by (6). Denote the constraint set of the CSP as C. Clearly, by (9) and (12),

$$\forall R \in C, \mu_R(v'_{var(R)}) \geq \lambda_{coordinator}. \quad (13)$$

Thus,

$$\begin{aligned} &\min\{\mu'_R(v_{var(R)}) | R \in C\} \\ = &\min\{1 | R \in C\} \qquad \text{(by (13) and (6))} \\ = &1. \end{aligned}$$

So, v_X is a solution to the CSP. Since in the case there is a solution, the value of $\lambda_{coordinator}$ is always greater than or equal to the threshold for solutions to the FCSP, clearly the solution is also a solution to the FCSP.

3) In other words, in one round of negotiation organized by agent A_j, in step n_H the solution v_X can be found. Moreover, according to the protocol, before step n_H, any solution to the FCSP of A_j cannot be accepted by all other agents. Accordingly, there does not exist any solution of A_j with an individual evaluation greater than the evaluation for the appointment. Alternatively, the solution v_X indeed is the optimum appointment among agents.

Therefore, the theorem holds. □

5 Fusion Operators

In this section, we give an axiomatic framework for fusion operators, and discuss their construction. The issue of evaluation fusion is also involved in a multi-agent system, developed by Scott $et\ al.$ in [14], for meeting scheduling, but it is different from ours here. We presents an axiomatic framework for this sort of operators and invents a method for constructing this kind of operators, whereas they just give a particular operator.

5.1 Axiomatic Framework

Definition 6 *A binary operator* $\oplus : [0,1] \times [0,1] \to [0,1]$ *is a* **fusion operator** *if it satisfies the following properties*[2]*:*

1) $\forall a_1, a_2 \in [0,1], a_1 \oplus a_2 = a_2 \oplus a_1$;

2) $\forall a_1, a_2, a_3 \in [0,1], (a_1 \oplus a_2) \oplus a_3 = a_1 \oplus (a_2 \oplus a_3)$;

[2] In this definition, properties 1, 2, 6 and 8 are inspired by our previous work[10]; the idea behind properties 3, 4 and 5 is from Zhang $et\ al.$ [22] (the operands of which take values on $[-1,1]$ instead of $[0,1]$); property 8 owes to Cai [3]. These papers are all related to stand-alone/distributed expert systems. So, our fusion operators may also be applicable to expert systems, especially distributed expert systems.

3) $\forall a_1, a_2 \in (0.5, 1], a_1 \oplus a_2 \geq \max\{a_1, a_2\}$;
4) $\forall a_1, a_2 \in [0, 0.5), a_1 \oplus a_2 \leq \min\{a_1, a_2\}$;
5) $\forall a_1 \in [0, 0.5), \forall a_2 \in (0.5, 1], \min\{a_1, a_2\} \leq a_1 \oplus a_2 \leq \max\{a_1, a_2\}$;
6) $\forall a \in [0, 1], a \oplus 0.5 = a$;
7) $\forall a \in [0, 1], a \oplus (1 - a) = 0.5$;
8) $\forall a_1, a_2, a_3, a_4 \in [0, 1], a_1 \leq a_2 \wedge a_3 \leq a_4 \Rightarrow a_1 \oplus a_3 \leq a_2 \oplus a_4$.

Let us explain the intuitions behind the above definition. Properties 1 and 2 are used to guarantee the result of a fusion operation is independent of the order of the operation. Property 3 captures the intuition that when two evaluations are both positive they should enhance the effect of each other, while property 4 captures the intuition that in the case where two evaluations are both negative, they should weaken each other. Property 5 means that in the case where two evaluations are in conflict we should get compromise. Property 6 exposes that if an agent has no idea about the proposal the agent should have no effect on the fused result. Property 7 means that in the case where two agents give exact opposite evaluations the coordinator agent cannot get any idea from these two agents. Property 8 captures the intuition that a fusion should be monotonic and do not decrease on $[0, 1]$. The bigger a value in $[0, 1]$ the higher an evaluation, estimated by the value, for the same proposal. Therefore, when the assessment for the evaluation of one agent is fixed and the another agent's increases, the assessment for the fused evaluation should not decrease.

The theorem below states that a fusion operator is a group.

Theorem 2 $([0, 1], \oplus)$ *is a commutative group.*

Proof. Clearly, the operator $\oplus$ on $[0, 1]$ is closed, and satisfies the associative and commutative laws. The unit element is 0.5 and the inverse element of a is $1 - a$. So, the theorem holds. □

This theorem is very interesting. Although this paper has not used the result, it bridges group theory and decision-making problems in multi-agent systems, and so may lead to some interesting and important properties.

By Definition 6, we can easily prove the following theorem:

Theorem 3

$$\forall a \in [0, 0.5), a \oplus 0 = 0; \tag{14}$$

$$\forall a \in (0.5, 1], a \oplus 1 = 1. \tag{15}$$

In the above theorem, (14) states that in the case two evaluations are both negative, if one evaluation represents the complete violation, the fused result means the absolute violation, *i.e.* the proposal is not acceptable absolutely. This is in accordance with the intuition.

In the above theorem, (15) implies that in the case no user is against the proposal, if there is a user who accepts the proposal completely, then the proposal should become an appointment candidate. It seems to be a little inconsistent with the intuition. If the weight of an agent is put into consideration in a decision-making process, however, it could be reasonable in the real life. For the reason, based on the idea behind our relative weight model [9], we define the concept of discounted evaluation for a proposal through a weight as follows:

Definition 7 *In an FCSP multi-agent system* $(\mathcal{A}, \oplus, \omega, \mathcal{P})$*, let the evaluation of the agent* A_i *for the proposal* v_X *be* $\alpha_i(v_X)$*, then the* **discounted evaluation**, $\alpha'_i(v_X)$*, of* A_i *for* v_X *is given by*

$$\alpha'_i(v_X) = \frac{\omega(A_i)}{\max\{\omega(A_j) | j = 1, \cdots, n_{\mathcal{A}}\}} \times \alpha_i(v_X). \tag{16}$$

Then when performing a fusion operator on discounted evaluations, even if two evaluations are positive, only in the case the evaluation of the agent with the highest weight among agents represents the full satisfaction to a proposal, the fused result means the complete satisfaction to a proposal. In fact, we have:

Theorem 4 *If and only if* $\omega(A_i) = \max\{\omega(A_j) | j = 1, \cdots, n_{\mathcal{A}}\}$*, then*

$$\alpha_i(v_X) = 1 \Rightarrow \forall a \in (0.5, 1], a \oplus \alpha'_i(v_X) = 1. \tag{17}$$

Proof. By (16), if and only if $\omega(A_i) = \max\{\omega(A_j) | j = 1, \cdots, n_{\mathcal{A}}\}$, then $\alpha'_i(v_X) = \alpha_i(v_X) = 1$. Thus, by (15), the theorem holds. □

5.2 Construction

Comparing Definition 6 with Definition 4 as well as Lemma 1, we can see that the fusion operators here are completely different from T-norms and T-conorms, but T-conorms can give us hint in constructing fusion operators. In the following, we will discuss this issue.

Firstly, we introduce T-conorm-like operators.

Definition 8 *An operator* $\triangledown' : [-1, 1] \times [-1, 1] \longrightarrow [-1, 1]$ *is a T-conorm-like operator if it satisfies:*
1) commutativity: $a_1 \triangledown' a_2 = a_2 \triangledown' a_1$;
2) associativity: $(a_1 \triangledown' a_2) \triangledown' a_3 = a_1 \triangledown' (a_2 \triangledown' a_3)$;
3) monotonicity: $a_1 \leq a_2 \wedge a_3 \leq a_4 \Rightarrow a_1 \triangledown' a_3 \leq a_2 \triangledown' a_4$;
4) unit: $a \triangledown' 0 = a$;
5) contrary: $a \triangledown -a = 0$,
where $a, a_1, a_2, a_3, a_4 \in [-1, 1]$.

By the above definition, we can easily prove the following lemma:

Lemma 2
1) $\forall a_1, a_2 \in [0, 1], a_1 \triangledown' a_2 \geq \max\{a_1, a_2\}$;
2) $\forall a_1, a_2 \in [-1, 0], a_1 \triangledown' a_2 \leq \min\{a_1, a_2\}$;
3) $\forall a_1 \in [-1, 0], a_2 \in [0, 1], \min\{a_1, a_2\} \leq a_1 \triangledown' a_2 \leq \max\{a_1, a_2\}$;
4) $\forall a \in (0, 1], 1 \triangledown' a = 1$;
5) $\forall a \in [-1, 0), -1 \triangledown' a = -1$.

Now by the above lemma, we can prove[3] the following theorem for constructing a fusion operator.

Theorem 5 *The following operator is a fusion operator:*

$$a_1 \oplus a_2 = h^{-1}(h(a_1) \triangledown' h(a_2)), \tag{18}$$

where $h : [0, 1] \to [-1, 1]$ *is an 1-1 mapping satisfying*

$$h(0) = -1, \tag{19}$$

$$h(1) = 1, \tag{20}$$

$$h(0.5) = 0, \tag{21}$$

$$\forall a \in [0, 1], h(1 - a) = -h(a), \tag{22}$$

$$\forall a_1, a_2 \in [0, 1], a_1 \geq a_2 \Rightarrow h(a_1) \geq h(a_2). \tag{23}$$

Notice if we restrict a $\triangledown'$ operator on $[0, 1]$ it turns into a T-conorm. So, by the above theorem sometimes we can construct a fusion operator from a T-conorm. For example, let $h(x) = 2x - 1$, from the following T-conorm

$$a_1 \triangledown a_2 = \frac{a_1 + a_2}{1 + a_1 a_2}, \tag{24}$$

we can obtain the following fusion operator

$$a_1 \oplus a_2 = h^{-1}\left(\frac{h(a_1) + h(a_2)}{1 + h(a_1)h(a_2)}\right) = \frac{\frac{(2a_1 - 1) + (2a_2 - 1)}{1 + (2a_1 - 1)(2a_2 - 1)} + 1}{2}. \tag{25}$$

6 An Example

We illustrate our approach by a simple meeting scheduling problem.

Three agents A_1, A_2 and A_3 will make an appointment for a meeting, chosen from 4 time intervals I_1, I_2, I_3 and I_4. And the type of meeting is business and the type of meeting host is boss. Suppose the constraints concerning with time intervals as shown in the following table:

	I_1	I_2	I_3	I_4
A_1	1	0.7	0.2	0.6
A_2	0.2	0.7	1	0.9
A_3	0	1	0	0.9

[3] The proof of the theorem is straightforward. For the lack of space, it is omitted here, but can be found in [11].

Suppose in agent A_1 there is another constraint which concerns two variables: time intervals and the type of meeting (e.g. emergency, business and leisure). The membership degree of this constraint is as shown in the following table:

	I_1	I_2	I_3	I_4
emergency	1	1	1	1
business	1	0.8	0.3	0.9
leisure	0	0	0.3	1

Suppose in agent A_2 there is another constraint which concerns three variables: time intervals, the type of meeting (*e.g.* emergency, business and leisure), and the type of meeting host (*i.e.* boss and colleague). The membership degree of this constraint is as shown in the following table:

time interval	I_1	I_1	I_1	I_1	I_1	I_1
meeting type	e	e	bu	bu	l	l
host type	bo	c	bo	c	b	c
membership	1	0.8	1	0.7	1	0
time interval	I_2	I_2	I_2	I_2	I_2	I_2
meeting type	e	e	bu	bu	l	l
host type	bo	c	bo	c	b	c
membership	1	0	1	0.6	1	0
time interval	I_3	I_3	I_3	I_3	I_3	I_3
meeting type	e	e	bu	bu	l	l
host type	bo	c	bo	c	b	c
membership	1	0.3	1	0	1	0
time interval	I_4	I_4	I_4	I_4	I_4	I_4
meeting type	e	e	bu	bu	l	l
host type	bo	c	bo	c	b	c
membership	1	1	1	1	1	1

In the above table, e=emergency, bu=business, l=leisure, bo=boss and c=colleague. In addition, suppose the three agents' thresholds for solutions are 0.5, 0.6 and 0.8, respectively.

In the round of negotiation organized by A_1, it first proposes I_1 as a proposal. Its own evaluation for the proposal is 1. The evaluations of the other two agents for the proposal are 0.2 and 0, respectively. They are less than their thresholds, and so the proposal is rejected by A_2 and A_3. Then A_1 proposes I_2 as another proposal. Its own evaluation for the proposal is 0.7. The evaluations of the other two agents for the proposal are 0.7 and 1, respectively, which are greater than their thresholds, and so the proposal is accepted by all agents. Thus, I_2 becomes an appointment candidate. By the fusion operator given by (25), we get 1 as the overall evaluation for I_2.

In the round of negotiation organized by A_2, it first proposes I_3 as a proposal. Its own evaluation for the proposal is 1. The other two agents' evaluations for the proposal are 0.2 and 0, respectively. Unfortunately, they are less than their thresholds, and so the proposal is rejected by A_1 and A_3. Then A_2 proposes I_4 as another proposal. Its own evaluation for the proposal is 0.9. The other two agents' evaluations for the proposal are 0.6 and 0.9, respectively. Fortunately, they are greater than their thresholds, and so the proposal is accepted by all agents. Thus, I_4 becomes another appointment candidate. By the fusion operator given by (25), the overall evaluation for I_4 is 0.99.

In the round of negotiation organized by A_3, when it first proposes I_2 as a proposal and the proposal is accepted by all agents (the reason as discussed in the round of negation organized by A_1).

I_2 has a higher overall evaluation than I_4. So, finally I_2 is promoted to become the appointment among the three agents.

Note that the negotiations of all agents start at the same time, and thus the negotiation of A_3 is necessary although all possible time intervals have been examined during the negotiations of A_1 and A_2.

7 Conclusion

Based on fuzzy techniques, the paper develops an agent-based approach for meeting scheduling problems. Compared with previous works, it is novel in three aspects. First, a meeting scheduling problem is modeled by FCSPs in multi-agent environment. Second, a kind of selfish protocol is presented. An appointment made through this protocol is a overall optimum common time slot. Third, an axiomatic framework is identified for fusing agents' individual evaluations for a proposal. The framework is also applicable to solution synthesis in distributed expert systems [21, 22, 20], parallel combination operations [10] in expert systems and aggregation operations [19] in fuzzy mathematics. In addition, a meeting scheduling example is used to illustrate the proposed methodology.

It is worth further developing: 1) other protocols for more complicated meeting scheduling problems; and 2) other models for fusing agents' individual evaluations for a proposal.

REFERENCES

[1] A. Bond and L. Gasser (Eds), *Readings in Distributed Artificial Intelligence*, Morgan Kaufman Publishers, 1988.

[2] R. Brancaleni, A. Cesta and D. D'Aloisi, 'MASMA: A Multi-Agent System for Scheduling Meetings', *Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, 31-49, (1997).

[3] J. Cai, 'Analysis and Comparison on Parallel Composite Algorithms for ES Uncertain Reasoning', *Journal of Xiamen University (Natural Science)*, **38:3**, 355-361, (1999).

[4] D. Dubois and H. Prade, 'A Class of Fuzzy Measures Based on Triangular Norms', *International Journal of General Systems*, **8**, 1, (1982).

[5] D. Dubois and H. Prade, 'Qualitative Possibility Theory and Its Applications to Constraint Satisfaction and Decision under Uncertainty', *International Journal of Intelligent Systems*, **14**, 45-61, (1999).

[6] L. Garrido and K. Sycara, 'Multi-Agent Meeting Scheduling: Preliminary Experimental Results', *Proceedings of the Second International Conference on Multi-Agent Systems*, 95-102, (1996).

[7] T. Haynes, S. Sen, N. Arora, and R. Nadella, 'An Automated Meeting Scheduling Systems that Utilizes User Preferences', *Proceedings of the First International Conference on Autonomous Agents*, 308-315, (1997).

[8] K. Kida, K. Yoshifu, T. Askura, T. Miyashita, 'Goodinator: Quick and Prudent Meeting Coordination with a Multi-Agent System', *Intelligent Agent Technology: Systems, Methodologies, and Tools*, J. Liu and N. Zhong (Eds.), World Scientific, 364-373, (1999).

[9] X. Luo, C. Zhang and J. Cai, 'The Weighting Issue in Fuzzy Logic', *Informatica: An International Journal of Computing and Informatica*, **21**, 255-262, (1997).

[10] X. Luo and C. Zhang, 'An Axiom Foundation for Uncertain Reasonings in Rule-Based Expert Systems: NT-Algebra', *Knowledge and Information Systems: An International Journal*, **1:4**, 415-433, (1999).

[11] X. Luo, H.F. Leung and H.M. Lee, 'Multi-Agent Meeting Scheduling with Fuzzy Constraints', Technique Report, Department of Computer Science and Engineering, the Chinese University of Hong Kong, 2000.

[12] K. Marriott and P.J. Stuckey, *Programming with Constraints: An Introduction*, the MIT Press, 1998.

[13] T. Schiex, H. Fargier and G. Verfaillie, 'Valued Constraint Satisfaction Problems: Hard and Easy Problems', *Proceedings of IJCAI-95*, 631-637, (1995).

[14] A. Scott, K. Jenkin and R. Senjen, 'Design of an Agent-Based, Multi-User Scheduling Implementation', *Multi-Agent Systems — Theories, Languages and Applications*, C. Zhang and D. Lukose (Eds.), Lecture Notes in Artificial Intelligence, **1544**, 152-165, (1998).

[15] S. Sen and E. Dufee, 'Using Temporal Abstractions and Cancelations for Efficiency in Automated Meeting Scheduling', *Proceedings of International Conference on Intelligent and Cooperative Information Systems*, 163-172, (1993).

[16] W. Slany, 'Scheduling as a Fuzzy Multiple Criteria Optimization Problem', *Fuzzy Sets and Systems*, **78**, 197-222, (1996).

[17] K. Sycara, and J. Liu, 'Distributed Meeting Scheduling', *Proceedings of the Sixteenth Annual Conference of Cognitive Society*, 538-588, (1994).

[18] T.K. Tan, 'Constraint-Based Distributed Meeting Scheduler', Honors Year Project Report No. 280, National University of Singapore, 1997.

[19] R.R. Yager, 'Aggregation Operators and Fuzzy Systems Modeling', *Fuzzy Sets and Systems*, **67**, 129-145, (1994).

[20] C. Zhang, 'Cooperation under Uncertainty in Distributed Expert Systems', *Artificial Intelligence*, **56**, 21-69, (1992).

[21] M. Zhang, *Synthesis of Solutions in Distributed Expert Systems, Ph.D thesis*, the University of New England, Australia, 1995.

[22] M. Zhang and C. Zhang, 'Potential Cases, Methodologies, and Strategies of Synthesis of Solutions in Distributed Expert Systems', *IEEE Transaction on Knowledge and Data Engineering*, **11:3**, 498-503, (1999).