Persuasion among Agents : An Approach to Implementing a Group Decision Support System Based on Multi-Agent Negotiation

Takayuki Ito and **Toramatsu Shintani**
Department of Intelligence and Computer Science
Nagoya Institute of Technology
Gokiso, Showa-ku, Nagoya, 466 JAPAN
E-Mail:{itota,tora}@ics.nitech.ac.jp

Abstract

In this paper, we propose a persuasion mechanism for negotiation among agents for a group decision support system and implement a group choice design support system (GCDSS). GCDSS helps a group decision to make a reasonable choice from alternatives. In the system, each user manages a system for an Analytic Hierarchy Process (AHP) and an agent. Each user subjectively constructs a decision hierarchy and determines the various weights of alternatives by using AHP. Based on the hierarchy and weights, agents negotiate with each other on behalf of their users. During the negotiation, agents persuade one another. Adopting some of the features of AHP, we implement a new persuasion mechanism. We have implemented the GCDSS to see how effectively the persuasion mechanism can be used. The results of our current experiments demonstrated that the persuasion mechanism is an effective method for a group decision support system based on multi-agent negotiation.

1 Introduction

Group decision support systems (GDSSs)[Desanctis and Gallupe, 1987] are being investigated very actively in the field of operations research. The GDSS aims to improve the process of group decision-making by removing common communication barriers, providing techniques for structuring decision analysis, and systematically directing the pattern, timing or content of discussions.

We can classify GDSSs into three types according to their approach to supporting a group. Type 1 GDSSs improve the decision process by facilitating the exchange of information among members. Type 2 GDSSs provide decision-modeling and group-decision techniques aimed at reducing the uncertainty and noise that occur in the group's decision process. Type 3 GDSSs are characterized by machine-induced group communication support and often provide expert advice in the selection and arrangement of rules to be applied during a meeting. If we introduce Artificial Intelligence (AI) methods into the Type 3 GDSSs, we can expect to further enhance the intelligence of their support. For this reason, and in keeping with the current high expectations for effective GDSSs, we here implement a Type 3 GDSS using intelligent agents.

The term *agent*, used in a variety of ways, has recently commanded much attention in the field of AI. For purpose of this paper, an agent can act autonomously and cooperatively in a network environment on behalf of its users. Agents have many crucial functors, one of the important being the attainment of consensus. Reaching a consensus also is a group decision process that should be supported by the Type 3 GDSS.

Agent negotiation for group decision support has been studied widely. In particular, multi-agent meeting scheduling [Ephrati *et al.*, 1994; Garrido and Sycara, 1996; Sen and Durfee, 1994] has been a current topic of research. [Sen and Durfee, 1994] has been focused on solving meeting scheduling problem using a central host agent. However, user preferences are not taken into account. [Ephrati *et al.*, 1994] presented an alternative approach which is economic in flavor. They introduced the Clark Tax Mechanism as a method for removing manipulability from agents. [Garrido and Sycara, 1996] has been focused on decenterized meeting scheduling with user preferences taking into account. They did not, however, establish how to measure the subjective judgements of users. In this paper, in order to measure the judgements of users, we employ the Analytic Hierarchy Process (AHP) [Saaty, 1980].

In general, a group reaches consensus by use of a vote. But the result of voting are often inconsistent, largely due to the inconsistency of voting rules: Majority Rule, Single Voting Rule, etc. Arrow's impossibility theorem has shown that no voting method exists which satisfies all of the following four conditions: positive association of social and individual values, independence of irrelevant alternatives, citizen's sovereignty and no-dictatorship[Luce and Raiffa, 1985]. In this paper we propose a persuasion mechanism[Ito and Shintani, 1996a; 1996b] rather than voting methods for negotiation among agents.

In agent negotiation, the persuasion mechanism can be defined as follows. When agent A persuades agent B, agent A sends a persuasion message to agent B. Then, according to the message, agent B tries to change its belief. If agent B is able to change its belief, the persuasion is a success. However, if agent B cannot change its belief, the persuasion is a failure. A concrete method for implementing the persuasion mechanism is given in section 3.3.

Our agents are known as *software agents* [Kautz *et al.*, 1994]. Their activities include carrying out tasks on behalf of their users, making suggestions to their users, and so on. Reliability and the ability to depute are important qualities of the software agents. It can be said that our agents have a high ability to depute users, since our agents negotiate with each other by use of the persuasion mechanism on behalf of users. In addition, our agents have an explanation mechanism. They can explain to their users why they have been persuaded, when they are persuaded, who persuaded them and how they are persuaded by use of graphical user interface. Because of their explanation mechanism, our agents are more reliable.

The aim of this paper is to present the persuasion mechanism among agents and to implement a Group Choice Design Support System (GCDSS) based on the persuasion mechanism. The paper consists of five sections. In section 2, we show the architecture of our system and the process of supporting a group decision. In section 3, we present an agent mechanism for our system. Our agent has a management mechanism for the user's hierarchy of AHP, a persuasion mechanism for negotiating with other agents, and an explanation mechanism for their reliability. In section 4, we show an implementation of the GCDSS using the persuasion mechanism and discuss the results of our current experiments. Some concluding remarks are presented in section 5.

2 Group Choice Design Support System

2.1 System Architecture

Figure 1 shows the architecture of our system. In our sys-

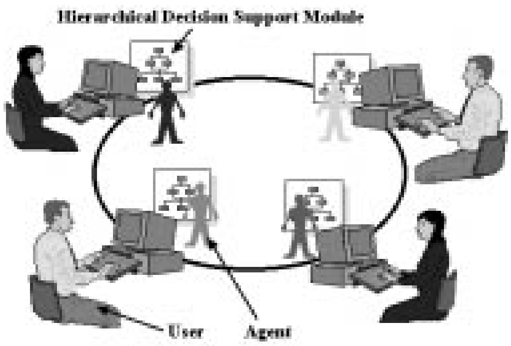

Figure 1: System architecture

tem, users' computers are connected by a network. Each

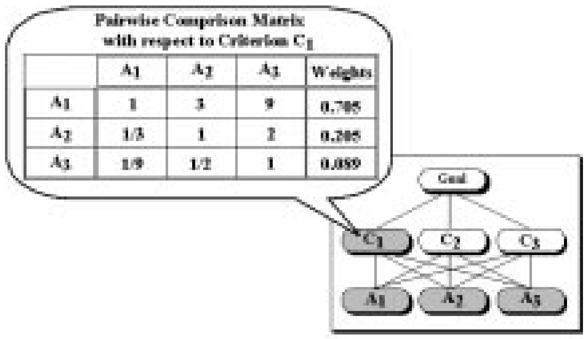

Figure 2: Analytic Hierarchy Process

user's decision-making is supported by their own hierarchical decision support module in our system. Agents manage their particular user's hierarchical decision support module, and negotiate based on the information it supplies. The agent's tasks are described in section 3. The hierarchical decision support module has functions to help generate alternatives, to make judgements for pairwise comparisons, and to construct a hierarchy. In order to lighten users' work load, we realize these functions using graphical user interfaces.

The process for supporting group decision-making in our system is described as follows: First, a host user proposes a topic to be decided. Second, users make and choose alternatives from alternative database they share. In order to get alternatives, users can employ a variety of methods, e.g., brainstorming. These methods are provided by the hierarchical decision support module. Third, each user constructs a decision hierarchy for AHP using the module. The hierarchy clarifies elements which should be considered in the decision making process. The module is used to quantify subjective judgements of decision makers by using AHP based pairwise comparisons. Fourth, agents negotiate with each other based on their users' subjective weights and decision hierarchy. Negotiation among agents is based on the persuasion mechanism described in section 3.2. Finally, the result of the negotiation is reported to all users.

2.2 Quantifying Subjective Judgements

In order to measure subjective judgements of users, we employ AHP in our system. The AHP is a method for making decisions that are hard to analyze quantitatively. It combines both systems approach and subjective judgements, and its primary purpose is to maximize the user's intuition and experience.

In the AHP, users decompose the problem into a hierarchy that consists of a goal, criteria (and possibly sub-criteria), and alternatives. The judgement of the pairwise comparison between factors (in Figure 2, alternatives A_1,A_2 and A_3) on a certain level is made with respect to the criterion that is a factor (in Figure 2, criterion C_1) on the upper level. By interpreting a set of values of judgements as a matrix (top left of Figure 2),

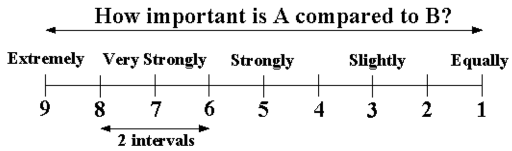

Figure 3: Scale of measurement for AHP

the weights (i.e., measurement of criteria) of factors are calculated analytically. To put it more concretely, we can achieve the weights of each factor as the eigen-vector for the max eigen-value of the pairwise comparison matrix. As a whole hierarchy, the weights of the alternatives can be calculated by composing the weights of the criteria.

A pairwise comparison matrix is characterized by the following: (1) diagonal values are 1; and (2) the values of elements in a matrix are $a_{ij} = 1/a_{ji}$. For judgements of pairwise comparisons in the AHP, we can use a 9 point scale consisting of five words (equally, slightly, strongly, very-strongly and extremely (Figure 3)) and four intermediate levels (e.g., between slightly and strongly)

AHP provides a measure the inconsistency in each set of judgements. This measure is called the inconsistency ratio (I.R.) and is defined as follows: $I.R. = (\lambda_{max} - n)/(n-1)$. Here, $\lambda_{max}$is the max eigen-value described above and n is the size of the pairwise comparison matrix. Ideally, a set of judgements in a pairwise comparison matrix will be consistent, and the inconsistency ratio will be 0. If the inconsistency ratio is no more than 0.1, this means that the pairwise comparison matrix is consistent.

In general, relative judgements are easier for users to make than absolute judgements. In the AHP, in order to lighten their work load, users can make comparisons between criteria using verbal and fuzzy expressions. Thus, the value of a pairwise comparison in the AHP is not a strict expression of a user's subjective judgement, but rather a rough approximation of that judgement. In our system, it may safely be assumed that a user's subjective judgement expresses 2 intervals with a certain value as the center point of the intervals on the 9 point scale (Figure 3). An interval is an unit of the 9 point scale. For example, while in the AHP the verbal expression "Very Strongly Important" means the value of 7 internally, in our system it means the values 6, 7, or 8.

In addition, we propose the expressions *fixed* and *assumed* to describe a user's belief in a judgement of pairwise comparison. For example, if the judgement is labeled fixed by the user, this means that the judgement value itself is fixed (i.e., the value is reliable). On the other hand, if the judgement is labeled assumed by the user, this means that the judgement value is also assumed (i.e., the value is unreliable). If an agent is persuaded in negotiation, the agent tries to adjust assumed judgements within 2 intervals in order to change the weights of the alternatives. These adjustments are made under the constraint of I.R. less than 0.1.

2.3 Public and Private Decision Hierarchy

In general, the AHP is employed in the common objective context (all users have the same objectives)[Dyer and Forman, 1992]. In this context, users decide one decision hierarchy among them first, then each user judges pairwise comparisons individually based on the decision hierarchy. In this case, because all users have the same objectives , i.e., criteria, the existing systems support users to construct only one decision hierarchy and to judge pairwise comparisons. On the other hand, our system supports group decision-making in the non-common objective context (each user has non-shared and sometimes hidden objectives). In the non-common objective context, it is difficult to construct a decision hierarchy among users, because the users have non-shared and hidden objectives (i.e., criteria). In our system, each user may construct a decision hierarchy individually. However, when there are same criteria among users, if the users can notice this fact, each can construct a decision hierarchy more effectively. Therefore, in our system, a decision hierarchy has public parts and private parts in order to make an effective group decision. The public parts can be referenced by all users. The private parts are hidden from other users. In the concrete system, the whole decision hierarchy is basically public. Individual users can designate as public or private each new criteria they create. The goal and alternatives must be public.

3 Agent's Behavior

3.1 Reducing the Pairwise Comparisons

Each agent manages the decision hierarchy and the pairwise comparison matrix constructed by its user. In general, the AHP requires too many judgements of its user. If there are n factors for a criterion, users have to make $n(n-1)/2$ judgements. If the number of levels of the decision hierarchy or the number of factors of the level is increased, more and more judgements are required. Acting on the user's behalf, the user's agent effectively reduces the number of judgements, which leads the user to make consistent judgements dynamically using the following methods.

In general, users have to judge all pairwise comparisons in the AHP. In fact, it is very hard for a user to judge all pairwise comparisons. In our system, the initial value of the elements of all pairwise comparison matrixes is 1 (i.e.,"Equally Important") as an assumed value. Because there are initial values, the user changes only the value that the user wants to judge.

In the AHP, the value of an element a_{ij} in the matrix equals W_i/W_j. Here, the W_i expresses the weight of the factor i. If a pairwise comparison matrix is consistent (i.e., $I.R. = 0$), a certain element can be inferred from the other two elements. For example, in a matrix, because the value of an element a_{12} is W_1/W_2 and the value of an element a_{23} is W_2/W_3, we can infer that the value of an element a_{13} is $a_{12} \times a_{23} = W_1/W_2 \times W_2/W_3 = W_1/W_3$. In the system, in order to reduce the number of judgements, agents infer the value of an element in a matrix using this feature of the AHP. When the agent infers the value of an element, if the element was judged by the user, the agent asks the user to change the judgement. If the element has never been judged, the agent changes the value of the element to the new value inferred. In this case, the new value is labeled as assumed. In general, in order to reduce the number of judgements, the Harker Method [Harker, 1987] is now widely employed. However, to use the Harker method, the covering condition must be satisfied on a matrix. In order to satisfy the covering condition, many judgements, from which all elements in the matrix must be inferred, are required. In a real system, it is also hard for the user to make judgements with the covering condition. If the user has a chance to satisfy the covering condition, the system exploits the Harker method instead of the simple method mentioned above to omit some redundant pairwise comparisons and check the consistency of each comparison dynamically. In a pairwise comparison matrix, when the value of $I.R.$ is more than 0.1, this means that the matrix is inconsistent and the user must remake all the judgements. Naturally, this remaking creates additional work load for the user. In our system, while the user is making judgements in a matrix, the agent is watching the $I.R.$ dynamically. When the $I.R.$ is more than 0.1, the agent requests that the user remake the present judgement.

3.2 The Persuasion Process

In the system, a negotiation among agents consists of persuasions between two agents. Figure 4 shows an example of negotiation among agents a, b, c, d, and e in the system. First, agents pair-off into groups: agents a and b make a group and agents c and d make a group. Next, within each group one agent who selected randomly persuades the other. In Figure 4, agent a persuades agent b and agent c persuades agent d. If these individual persuasions succeed, the persuading agents assume the representation of their respective groups. In Figure 4, each persuasion succeeds, and agent a and agent c are representative of their groups. If the persuasions had failed, the agents would change places. For example, if the agent a failed to persuade the agent b, the agent b persuades the agent a next time. The groups are now a, b and c, d, and the representatives advance to negotiate with each other singly. During negotiation between agents a, c, and e, agent a persuades agent e. So that the groups become a, b, e and c, d. Finally, agent c persuades agent a, and the agents reach a consensus.

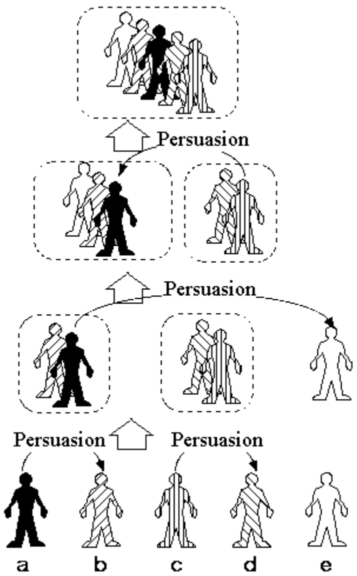

Figure 4: Negotiation among agents

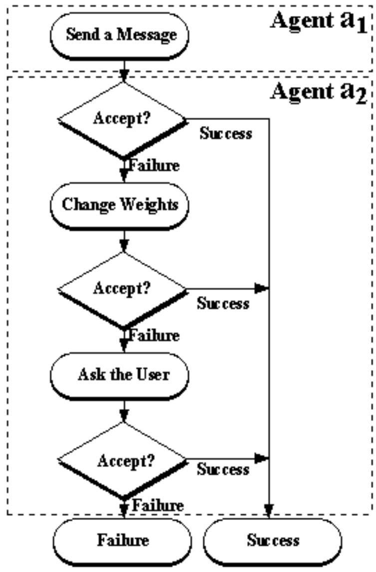

Figure 5: The process of persuasion (Agent a_1 persuades Agent a_2)

Figure 5 shows the process of persuasion between two agents. Suppose that agent a_1 and agent a_2 are in a group and agent a_1 persuades agent a_2. First, agent a_1 sends a persuasion message to agent a_2. The persuasion message is the most preferable alternative that has the highest weight and is decided by agent a_1's user with the AHP. Secondly, agent a_2 accepts or rejects this message according to the following process.

First, agent a_2 checks whether its own most preferable alternative is the same as that in the persuasion message. If this alternative is mutual, the agent a_2 accepts the persuasion message and this persuasion is a success. If not, agent a_2 does not accept the message and the persuasion process advances to the next step.

Second, agent a_2 tries to change the preference order of alternatives by adjusting the judgements of matrixes in the decision hierarchy. Figure 6 shows an example. The top and bottom halves of Figure 6 show, respectively,

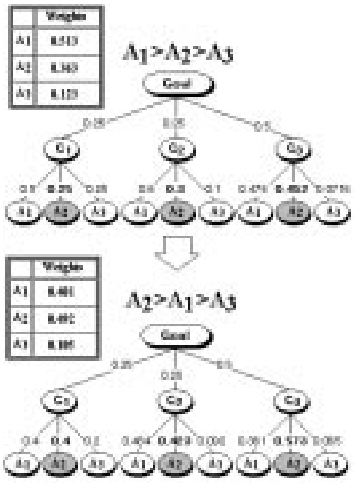

Figure 6: An example of adjusting judgements

the hierarchy before and after adjustment. In the top half of the figure, alternative A_1 is more preferable than alternative A_2 or A_3. Now, suppose that agent a_1 proposes alternative A_3 as the persuasion message. Agent a_2 tries to adjust the judgements in order to change the preference order so that alternative A_2 is more preferable than alternative A_1.

The agents adjust the judgements in a matrix by employing the following method. From the feature of the AHP, in order to increase the weight of the alternative I_i, we increase the value of elements at ith row except a diagonal element. In the system, the agents increase the value of assumed elements of ith row of the matrix except a diagonal element within 2 intervals of the 9 point scale, in order to increase the weight of alternative I_i and change the preference order of the alternatives. Figure 7 shows an example of such an adjustment. Suppose that all elements of the matrix are labeled *assumed*, and agent a_2 wants to increase the weight of alternative A_2. In this case, agent a_2 increases 1 interval of the value of elements (i.e., judgements) on alternative A_2 against the alternative A_1 and A_3. By this adjustment, alternative A_2 becomes most preferable. Agent a_2 asks the user for permission to change the weights in practice. If the user permits, agent a_2 changes the judgements. If not, agent a_2 tries to adjust again.

Third, agent a_2 checks whether the persuasion message can be accepted using his new preference order. If the persuasion message can be accepted, this persuasion is a success. If not, this persuasion is a failure.

Fourth, the agent asks the user to change his or her judgements. If the user agrees, agent a_2 indicates which judgement the user should change.

Fifth, agent a_2 again checks whether or not the persuasion message can be accepted.

Note that the reliability of the system is best enhanced by the persuasion mechanism's second and fourth steps.

	A_1	A_2	A_3	Weights
A_1	1	2	2	0.500
A_2	1/2	1	1	0.250
A_3	1/2	1	1	0.250

	A_1	A_2	A_3	Weights
A_1	1	1	2	0.400
A_2	1	1	2	0.400
A_3	1/2	1/2	1	0.200

Figure 7: An example of adjusting judgements in a matrix

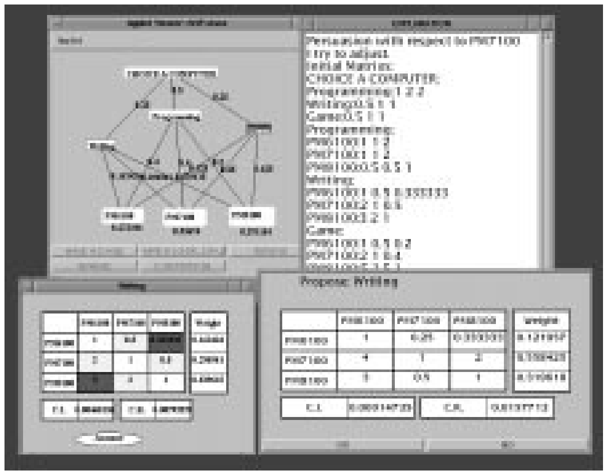

Figure 8: An example

3.3 Reliability

The reliability is an important factor to implement a software agent. The agent briefs its users on the details of negotiation via graphical user interface during negotiation. The agent explains which agents persuaded, which element in a matrix should be changed, which group the agents participated in at the time, and so on. This explanation mechanism renders the agents more reliable.

4 An Example and Discussion

As an experiment, we used the system to choose a new computer for our laboratory. The number of members in our laboratory is 12, and each research group has different objectives. Therefore, the group decision-making context was a non-common objective context. The common objectives (i.e., criteria) were writing papers, programming, and so on. The non-common objectives were playing games, music, and so on.

Figure 8 shows an example of the persuasion mechanism of the system. Here, the agent received an alternative "PM7100" as a persuasion message. Then, the agent adjusted judgements and proposed the new judgements by showing the window at the bottom right of Figure 8. This shows the agent's explanation in the process of a persuasion. The agent asks the user's permission to change particular judgements. The user agrees by pushing the *OK* button, or disagrees by pushing the *NO* button. The top right window in Figure 8 gives a detailed explanation.

These experiments have yielded some interesting results that merit discussion. A consensus is sometimes disturbed by the user who makes arbitrary judgements. To deal with such cases, we must consider the following. (1) In the AHP, we generally make decisions constructively. For this reason, arbitrary judgements should be prohibited. This idea can be applied to many cases of group decision making, but it also restricts the user's judgements. (2) An arbitrary judgement can be regarded as an opinion of the user. In this case, it will be worth reaching a consensus among a sub group that excludes this user. In addition, this idea can be applied to cases in which a group does not need consensus among all members. For example, in deciding the destination for a trip, we do not require consensus among all members. We can assume that the member who makes arbitrary judgements does not want to go on the trip, and can thus be disincluded.

Finally, the necessity of the explanation mechanism, particularly with regard to the persuasion mechanism, should be discussed. The successful persuasion of an agent requires the compliance of that agent's user. At first, we implemented a system in which the users were removed from the negotiation of their agents. In this case, we found that even if the users knew the system architecture and the process of agent negotiation in advance, the users who were persuaded were not satisfied. In other words, the reliability of agents is an important factor in realizing a multi-agent system. For this reason, in our system, as we have proposed in section 3.3, each agent explains to the user the details of the negotiation, in order to gain the user's consent and make agents more reliable.

5 Conclusion

In this paper, we proposed a persuasion mechanism among agents and implemented a group decision support system GCDSS to see how effectively the persuasion mechanism can be used. The GCDSS helps a group decision to make a reasonable choice from alternatives. We found useful characteristics of AHP for multi-agent based group decision support systems. The characteristics are as follows. (1) We can quantify the subjective judgements of users by using AHP. Agents negotiate effectively based on the subjective judgements quantified. (2) The verbal and fuzzy measurement of AHP enables us to realize the persuasion protocol. Agents can adjust the decision hierarchy of their users. The results of our current experiments demonstrated that the persuasion protocol is a suitable method for reaching a consensus among agents in group decision support systems.

References

[Desanctis and Gallupe, 1987] Gerardine Desanctis and R. Brent Gallupe. A foundation for the study of group decision support systems. *Management Science*, 33(5):589–609, May 1987.

[Dyer and Forman, 1992] Robert F. Dyer and Ernest Forman. Group decision support with the analytic hierarchy process. *Decision Support Systems*, 8:99–124, 1992.

[Ephrati *et al.*, 1994] Eithan Ephrati, Gilad Zlotkin, and Jeffrey S. Rosenschein. A non-manipulable meeting scheduling system. In *The Thirteenth International Distributed Artificial Intelligence Workshop*, pages 105–125, July 1994.

[Garrido and Sycara, 1996] Leonardo Garrido and Katia Sycara. Multi-agent meeting scheduling: Preliminary experiment results. In *Proceedings of Second International Conference on Multi-Agent Systems(ICMAS-96)*, pages 95–102. AAAI Press, December 1996.

[Harker, 1987] Patrick T. Harker. Incomplete pairwise comparisons in the analytic hierarchy process. *Mathmatical Modelling*, 9:353–360, 1987.

[Ito and Shintani, 1996a] Takayuki Ito and Toramatsu Shintani. An approach to a multi-agent based scheduling system using a coalition formation. In *Proceedings of the Ninth International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expart Systems (IEA/AIE-96)*, page 780. Gordon and Breach Science Publishers, June 1996.

[Ito and Shintani, 1996b] Takayuki Ito and Toramatsu Shintani. Implementing an agent negotiation protocol based on persuasion. In *Proceedings of Second International Conference on Multiagent Systems(ICMAS-96)*, page 443. AAAI Press, December 1996.

[Kautz *et al.*, 1994] Henry Kautz, Bart Selman, Michael Coen, Steven Ketchpel, and Chris Ramming. An experiment in the design of software agents. In *AAAI-94*, pages 438–443, 1994.

[Luce and Raiffa, 1985] Robert D. Luce and Howard Raiffa. *Games and decisions.* Dover Publications, 1985.

[Saaty, 1980] Thomas L. Saaty. *The Analytic Hierarchy Process.* McGraw Hill, 1980.

[Sen and Durfee, 1994] Sandip Sen and Edmund H. Durfee. On the design of an adaptive meeting scheduler. In *The Tenth IEEE Conference on Artificial Intelligence for Applications*, pages 40–46, March 1994.