



# IMPROVING AGENT-BASED MEETING SCHEDULING THROUGH PREFERENCE LEARNING

<sup>1</sup>MD. NASIR SULAIMAN, <sup>2</sup>TANG EN LAI, <sup>3</sup>MOHD HASAN SELAMAT, <sup>4</sup>ZAITON MUDA

<sup>1</sup>Assoc.Prof and Dr., Department of Computer Science and IT, UPM University, Malaysia, 43400.

<sup>3</sup>Assoc.Prof., Department of Computer Science and IT, UPM University, Malaysia, 43400.

E-mail: [nasir@fsktm.upm.edu.my](mailto:nasir@fsktm.upm.edu.my), [enlai80@gmail.com](mailto:enlai80@gmail.com), [hasan@fsktm.upm.edu.my](mailto:hasan@fsktm.upm.edu.my),  
[zaiton@fsktm.upm.edu.my](mailto:zaiton@fsktm.upm.edu.my)

## ABSTRACT

This paper presented an autonomous Secretary Agent (SA) that can perform meeting scheduling task on behalf of their respective user through negotiations. Previous study of searching strategy uses relaxation process to allow agents negotiate by relaxing their preference when conflict arises. However, this increased the cost of searching process. As a result, an improvement of relaxation searching strategy by adapting Neural Network (NN) learning mechanism is proposed. The back-propagation learning method is used in this research to intelligently predict the participants' preferences and guide the host in selecting proposals that are more likely to get accepted. Hence, higher quality solution can be found in lower communication cost. The comparison result between the proposed and two previous estimation strategies showed improvement of quality of the solution as well as the communication cost of the proposed strategy.

**Keywords:** *Meeting Scheduling, Agent-based Negotiation, Negotiation Strategy, Neural Networks,*

*Back-propagation learning method.*

## 1. INTRODUCTION

Meeting scheduling is a routine task in the office, which is tedious and time-consuming. To find a meeting time that satisfies every participant is not merely considering the availability of the participant, but also their preferences. Usually, participants hide their personal schedule, preferences and constraints from others for privacy. In addition, meeting scheduling becomes a distributed task when organizations structure is more likely to transform to distributed structure. This makes the meeting scheduling task more difficult.

The application of agent technology in meeting scheduling causes the emergence of Agent-based Meeting Scheduling System (ABMS) or Multi-Agent Meeting Scheduling [1][2][3][4]. Generally, ABMS consists of a group of agents called secretary agent (SA), which act as personal assistant that responsible for meetings scheduling

task. Each SA is operating on behalf of their respective user. SA negotiate with other SAs to determine a meeting time and at the same time concerns their user's preferences as when the preferred meeting time [13][7][8]. The negotiation over a meeting can be viewed as a distributed searching process in a search space of calendar, guided by the host. Therefore, the quality of the solution found is highly depends on searching strategy that the host applies during negotiation [2][14].

The objective of this research is to improve the relaxation searching strategy and relaxation with preference estimation strategy. The improved searching strategy is obtained by learning and predicting participants' preferences before selecting proposals for negotiation. This research proposed an NN learning approach in SA to perform the participants' preference learning in ABMS.

The following section discusses some related research in domain of meeting scheduling problem. Section 3 presents the model of meeting scheduling agent. Section 4 and section 5 describes the architecture of NN used in ABMS and how the new proposed strategy works. Section 6 explains the design of experiment as well as the measurement method. The result of the experiment and some discussion will be presented in section 7. Finally this paper ends up with conclusion in section 8.

## 2. RELATED WORKS

There are many researchers proposed different approach in meeting scheduling problem. Meeting scheduling problem is first studied by [11]. They discuss the meeting scheduling problem as timetable rearrangement problem. They present a meeting scheduler with an automatic mechanism for updating a schedule of meetings by applying heuristic algorithm. Subsequently, in Ref.[17] model the meeting scheduling problem with parallel algorithm. In their approach, two NNs interact with each other to generate meeting scheduling solutions yielding both meeting-assignment and person-assignment scheduling. In our study, meeting scheduling is viewed as a searching process over user's and participants' calendar to find a common available time.

Recent researches adopt the agent-based approach in meeting scheduling problem. Work presented in [3] introduced an autonomous and independent agent to schedule meeting in distributed environment. Their study focused on examined the meeting scheduling performance when user information (calendar and preference) is kept for privacy.

Furthermore, another researcher [14][15] study the usefulness of different heuristic negotiation strategies in solving distributed meeting scheduling problem. The result of the analysis shows that no strategy combination dominates one another over all circumstances and led them to develop an adaptive scheduling. This allows agent to predict the best strategy combination for different situation. The author presents the usage of user preferences to evaluate the acceptability of meeting proposals and introduced voting theory to formally represent and reason with conflicting preferences [13]. Similar user preference model is also presented in

literature [2].

[5] present a design and implementation of ABMS, in which the agents have knowledge of their user preferences. A novel scheduling algorithm is also outlined by the author. In their literature, architecture of the meeting scheduling agent is described. This literature provides basic requirement and guide in developing an ABMS. Some researchers have study different approaches to improve the performance of meeting scheduling. Woo *et al.* in [18] present cooperation and rescheduling strategy to reduce scheduling failure when agent does not meet a consensus time slot. This can increase the reliability between agent and user.

In other approach in [12] A\* algorithm is used to select high potential node from the search tree for solution and thus pruning the rest to speedup searching process. Although this approach gives optimum or nearly optimum solution without exploring entire search tree, it is impossible to implement in distributed environment as part of the search space in meeting scheduling problem. Similar approach has also been proposed by Chun and Wong in [2], which is known as the preference estimation technique. This technique allows an A\* algorithm to be implemented in distributed environment and allows optimal solution to be found. But the increase of communication cost is the weakness of this approach.

## 3. EXPERIMENTAL AGENT-BASED MEETING SCHEDULING SYSTEM (ABMS)

ABMS consist of several software agents called secretary agents (SA) that act on behalf of their user in scheduling meetings. SA knows the preferences and constraints of their respective user. It also has access of their user's personal calendar. User can interact with SA through a user interface as shown in Figure 1. Functions such as request meeting, cancel meeting, view scheduled meeting as well as customize user preference are provided in this interface.

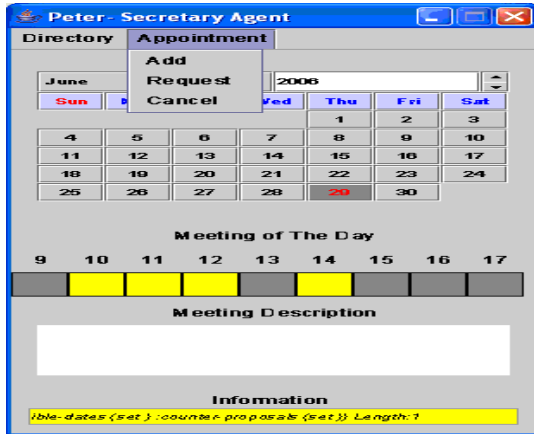


Figure 1: GUI of Secretary Agent

#### 4. USER PREFERENCE REPRESENTATION

Sen *et. al.* in [13] and Chun and Wong in [2] propose a simple model to effectively represent user preferences in ABMS. From their study, user preferences can be defined by a set of meeting attribute and a simple interface is used to allow the user to customize their preferences (Refer to Figure 2).

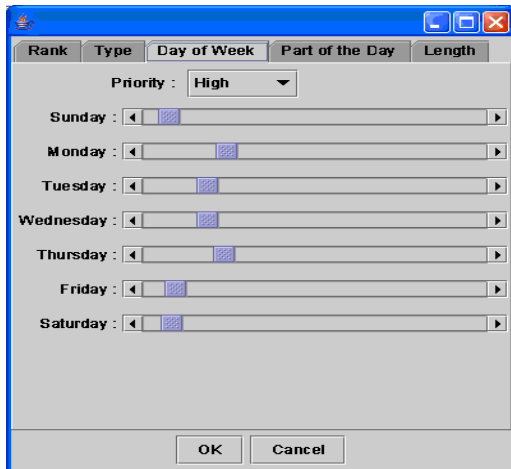


Figure 2: GUI for User Preferences Customization.

Each variable attributes is associated with a domain that defines a set of possible values, which an attribute may be assigned. For example, variable attribute “day of week” has a set of values of “Sunday”, “Monday”, and so on. User can define the priority on all variable attributes and the preferred values for their possible values. In this study, four variable attributes are used in user preference model, which include period of time that agent considers in scheduling a meeting,

duration of a meeting, and both part-of-day and day-of-week information. SA can evaluate how “satisfied” a user is with a proposal or counter-propose the user by calculating the “preference level”. For agent  $i$  with  $n$  variable attributes, the preference level  $pl_i$  for a proposal or counter-proposal  $P_x$  is defined as

$$pl_i(P_x) = \sum_{j=1}^n ap_j pv_{jk} \quad (1)$$

where  $pv_{jk}$  is the preference value for the assignment of domain value  $x_k$  to the variable attribute  $v_j$  and  $ap_j$  is the attribute priority. The details of user preference model and proposal evaluation are described in [2].

#### 5. USER PREFERENCE LEARNING

When a meeting request is initialized, a series of negotiations should be carried out between SAs to find a consensus time slot for a meeting. The agent who initializes the meeting request becomes the host. Agents who are invited for the meeting become participants. The negotiation between host and participants can be done through sending messages, which contain proposals and counter-proposals with their preference level.

The negotiation protocol used in our study is a multistage contract net protocol. Contract net protocol is a protocol designed for allow one-to-many negotiation [16]. Consequently, multistage contract net protocol is an enhancement of the contract net to support multi-round iterative negotiation. Similar protocol is also used in [2] [14]. The basic steps of this negotiation are illustrated as sequence diagram in Figure 3.

For the searching strategy, negotiation between host and participants is a searching process that is guided by host agent. The efficiency and accuracy of the solution (committed meeting) highly depends on the host negotiation strategy. Another strategy that can be applied in meeting scheduling problem is called relaxation-based negotiation [3]. In this strategy, each potential solution is assigned a heuristic value. The potential proposal will be sorted in an array with decreasing heuristic value. The highest preference level proposal (first proposal in the array) will be selected first for negotiation. If a solution is not found, the agent will relax to lower preference level of the proposal and expand the search tree until a mutually beneficial solution is

found. In our study, the preference levels of the potential proposals are used as the heuristic values.

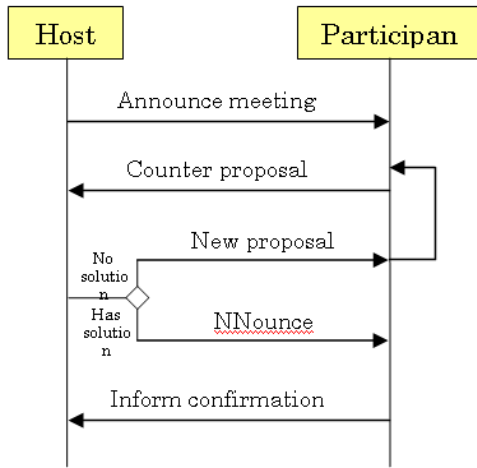


Figure 3: Sequence Diagram of SA Negotiation

Chun and Wong [2] have extended the relaxation strategy with a novel preference estimation technique to optimize the relaxation-based negotiation. Experiment results from their study shows that preference estimation technique is able to improve the relaxation process by obtaining optimal solution. Nevertheless, more communication cycle is required when preference estimation technique is used. Table 1 shows the strengths and weaknesses of relaxation strategy as compared to the extended relaxation with preference estimation strategy according to the experiment done in their study.

Table 1: Strengths and Weaknesses of Relaxation Strategy and Extended Relaxation with Preference Estimation Strategy

	Relaxation	Relaxation + preference estimation
Strengths	Find fast solution; Low communication cost;	Find an optimal solution; Reach global satisfaction;
Weaknesses	The solution might not be optimal; Solution bias towards the meeting initiator/host;	High communication cost (more cycle needed to find optimal solution); Lower proposal success rate (more proposal need to be negotiated);

## 6. NEURAL NETWORK (NN) MODEL

The main goal of this study is to extend the relaxation based negotiation with NN Model. The objective of this extension is to maintain the strengths of both strategies mentioned in Table 1 and eliminate their weaknesses. The basic idea of this study is that user preferences for each participant are learned by a set of NN. The trained networks then are used to predict the participants' preferences and guide the host in selecting proposals that are more likely to get accepted. Hence, higher quality solution can be found in lower communication cost.

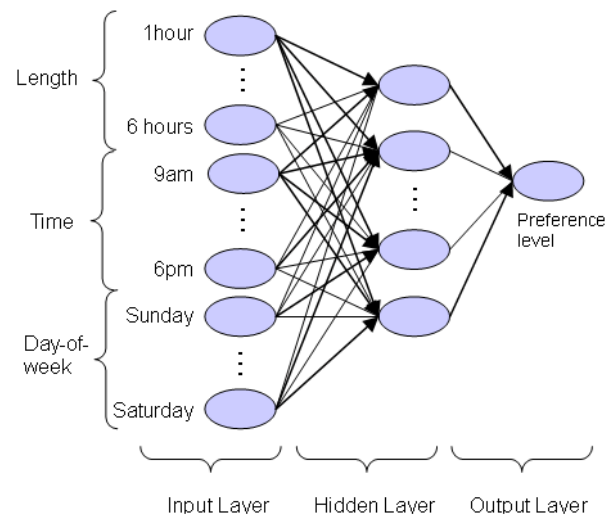


Figure 4: Back-propagation Architecture for Preference Learning

A Back-Propagation learning is used in this study to learn the participants' preferences. Back Propagation NN has proven its effectiveness in many problem domains [19][10]. It is a feed-forward NN with supervised learning. It uses back-propagation of error to adjust the connection weights to perform learning [9]. The architecture of this network is organized into three main layers: an input layer, an output layer, and a hidden layer as shown in Figure 4. The input layer consists of number of input nodes that receive meeting attributes value as input signal of the network. All nodes in input layer are fully connected to a hidden layer of four hidden nodes. Next, all node in hidden layer are fully connected to a single node in output layer. The output layer's node represents the preference level for given meeting attributes.

During network training, a set of meeting attribute values with preference level are prepared to form input-output pair data. Input data are normalized into suitable value before presented to the input layer. Input data flows through the network until the node in output layer. The actual output of the network will compare to desired output and the error of network output is calculated and passed backward from output layer to input layer to adjust the connection weights between nodes. The theory and computational details of back-propagation network are described in [9].

## 7. RELAXATION WITH NN MODEL

SA with Relaxation and NN Model as proposed in this study is an extension of relaxation searching strategy. The architecture of proposed strategy is illustrated in Figure 5. Based on the figure, the architecture of SA is divided into three components, which are Negotiation Module, Preference Learning Module, and Proposal Evaluation Module.

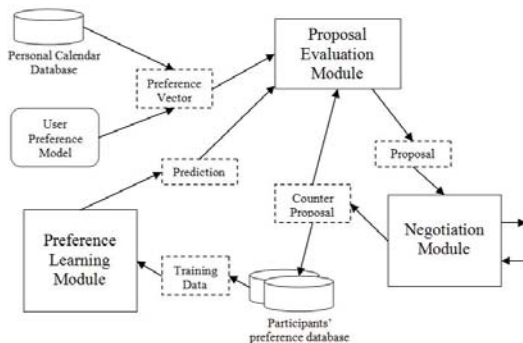


Figure 5: The Architecture of SA with Proposed Strategy

Below are the main process done by SA when it negotiates meeting schedules with other SA.

1. Participant preference data collection
2. Preference data pre-processing
3. NN Training
4. Participant Preference Prediction
5. Proposal Evaluation and Selection

When a SA negotiate meeting with other SAs, the counter-proposals from other SA are stored in database as training data respective to the SA's name. Preference Learning Module creates a set of NNs to learn the preferences of each SA known by the user's SA with the training data collected.

Then, some negotiation steps are carried out in negotiation module to perform the relaxation with NN searching strategy as described below:

**Step 1:** Announce meeting – A SA, which initialize a meeting plays the role as host to control the negotiation process.  $n$  highest  $apl'$  proposals are selected from  $PPV$  and sent to all participants together with the meeting detail.

**Step 2:** Participants' responses – Each participant agent that receives the meeting announcement generates a list of possible proposals. The participant agent first determines whether any of  $n$  proposal is feasible or not, then it selects  $m$  counter-proposals, has higher preference level from the list of possible proposal and send back to host as counter proposal. Refuse message will be sent if there are no feasible proposal and counter-proposal available.

**Step 3:** Consolidate participant replies – Host collects all participants' replies. It keeps waiting for messages from participants until messages from all participants are received.

**Step 4:** Proposal evaluation – All the messages received are evaluated to determine whether a solution is found. This step is ended up with one of the alternatives below:

- No solution – all possible proposals in  $PPV$  have been proposed but are refused by participants and no possible proposal can be proposed. Go to Step 6.
- Solution found – A proposal that is feasible for all participants is found. If the solution is evaluated as optimal by proposal evaluation table, and an accepted proposal is sent to all participants to request for confirmation and go to Step 5. In other hand, if it is not on optimal solution, new  $n$  proposal are selected and the negotiation step will be reset to Step 1.
- No solution, but have possible proposals that are not yet been proposed – new  $n$  proposal are selected and the negotiation step is reset to step 1.

**Step 5:** Participants' confirmation – Participants check the feasibility of the proposal for second times. If feasible, the meeting is committed in participant's calendar and a message is sent to inform the host agent.

The solution found in step 4 is done by finding a common acceptable proposal that has highest  $apl'(Px)$ . For every new counter proposal received, the predicted  $apl'(Px)$  for each proposal is calculated in  $PVV$  and the  $PVV$  is resorted according to new calculated  $apl'(Px)$ . Predicted  $apl'$  of a proposal  $Px$  for  $n$  participants is defined as below. The calculation is based on [2] except some modification in condition of value of predicted preference level  $pl'$ .

$$apl'(Px) = \frac{\sum_{i=1}^n pli'(Px)}{n}$$

where

$$pli'(Px) = \begin{cases} pli(Px) & \text{if } Px \text{ has already been counter proposed} \\ \text{Predicted } pli(Px) & \text{if } \text{Min}(pli(Px)) > \text{Predicted } pli(Px) \\ \text{Min}(pli(Px)) & \text{if } \text{Min}(pli(Px)) < \text{Predicted } pli(Px) \end{cases}$$

The predicted preference level of agent  $i$ ,  $pli'(Px)$  is the actual  $pli(Px)$  if the proposal has already been proposed by agent  $i$ . Otherwise,  $pli'(Px)$  will be either predicted  $pli'(Px)$  or minimum preference level received so far from the participant. Since participants always counter propose their highest preference level first, we can estimate that next counter proposal from the participant will have lower preference level and current counter proposal is the minimum preference level received so far. Therefore, it is impossible to has predicted  $pli(Px)$  greater than  $\text{min}(pli(Px))$ . As the result,  $\text{min}(pli(Px))$  will be selected as  $pli'(Px)$ . The current counter-proposal or the minimum preference level is stored in a Lowest Preference List (LPL) as the upper bound of the next counter proposal.

## 8. EXPERIMENTAL DESIGN

The experiment is performed through a

computer simulation to compare the performance of proposed strategy (Stra-3) with previous strategies: namely relaxation strategy (Stra-1) and relaxation with preference estimation strategy (Stra-2). During the experiment, six SAs are created to perform the scheduling task. The preferences of these agents are randomly generated as we assume each person have different preferences. The initial state of each agent's calendar is preoccupied with a set of randomly generated timeslot according to their calendar density (CD). For example, if the CD is 5, that means there are five hours of preoccupied timeslot in each agent's calendar. Next, a set of meeting events are randomly generated. The attributes of the meeting such as length, duration, host, participants are also randomly generated. Below are the experiment parameters that were selected for this computer simulation.

Number of hours to be schedule = 35 hours;  
 Maximum person in a meeting = 6 persons;  
 Calendar size = 7 days;  
 Length of day = 9 hours;  
 Calendar density = from 0 to 13 hours;  
 Number of proposal in a message sent = 1;  
 Number of counter proposal in a message sent=1;

## 9. MEASUREMENT METHOD

At the end of this experiment, the computer simulation results from three different searching strategies are recorded and evaluated. Four types of measurements are used for this evaluation purpose. These include the average preference level of scheduled meeting, the optimality of solution, the communication cost, as well as rate of successful proposal. Below are the measurements used to obtain the simulation result based on calculation formula proposed by Chun and Wong in Ref.[2].

- i. Average preference level of committed meetings:
  - $|M_{total}|$  number of successful meeting in a set of meetings  $M_{total}$
  - $PL_{ij}$  preference level of agent  $i$  for the committed proposal for meeting  $M_j$
  - $AP_j$  average preference level of committed proposal for all participants involved in meeting  $M_j$

$$AP_j = \left( \frac{\sum_i^{n_j} PL_{ij}}{i} \right) / n_j$$

where  $n_j$  is number of participants in meeting  $M_j$ .

$AP_{avg}$  average of the  $APs$  for all successful meeting in a set of meetings  $M_{total}$

$$AP_{avg} = \left( \sum_{j=1}^n AP_j \right) / |M_{total}|$$

- ii. Optimality of solution  
 $O_j$  optimal  $AP$  for all proposals in meeting  $M_j$   
 $ADO_j$  difference between optimal  $AP$  and  $AP$  of committed proposal for meeting  $M_j$

$$DO_j = O_j - AP_j$$

$ADO_{avg}$  average of the  $DO_j$  for all successful meetings in a set of meetings  $M_{total}$

$$ADO_{avg} = \left( \sum_{i=1}^{n_j} DO_j \right) / |M_{total}|$$

- iii. Communication cost  
 $CC_j$  number of negotiation cycles required to schedule a meeting  $M_j$   
 $CC_{avg}$  average of number of negotiation cycles required to schedule a set of meetings  $M_{total}$

$$CC_{avg} = \left( \sum_{i=1}^{n_j} CC_j \right) / |M_{total}|$$

- iv. Successful proposal rate  
 $PR_j$  rate for successful proposal in a meeting  $M_j$

$$PR_j = \frac{\text{number of proposal that are accepted}}{\text{total number of proposal sent}}$$

$PR_{avg}$  average of  $PR_j$  in scheduling a set of meetings  $M_{total}$

$$PR_{avg} = \left( \sum_{i=1}^{n_j} PR_j \right) / |M_{total}|$$

## 10. RESULTS AND DISCUSSION

In the process of constructing the NN, some experiments are performed to determine the dimension of network structure (number of input, hidden, and output nodes), the learning rate as well as the momentum value. After many combinations of values are tested, the best combination that support the optimal learning (has lowest mean square error - MSE) is chosen. A total number of 273 data instances with 3 meeting attributes are generated based on the random generated user preference model. 70 percents (191 instances) of overall data are used as training dataset and remaining 30 presents (82 instances) data are used as testing dataset. Table 2 shows the correctness of the network output after training in four networks with different network parameter. An acceptable minimum MSE as well as maximum percentage of correctness achieved by network N2 shows the efficiency of the network in learning meeting preference. The performance of the network N2 with learning rates of 0.1 and momentum values of 0.5 run in 1000 epochs is illustrated in Figure 6.

Table 2: Accuracy of Network Prediction Before and After Training

Network (learning rate, momentum)	RMS Error Achieved	% of Correctness (Training Date)	% of Correctness (Testing Date)
N1 (0.1, 0.1)	0.030792	95.29	93.90
N2 (0.1, 0.5)	0.010982	96.34	97.56
N3 (0.01, 0.1)	0.030764	93.19	91.46
N4 (0.01, 0.5)	0.041529	91.62	92.68

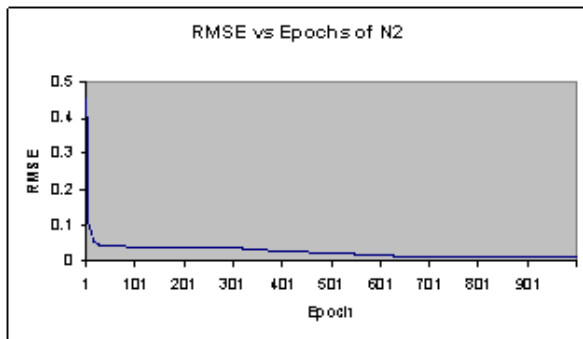


Figure 6: Performance of the Back Propagation model

## 11. SEARCHING STRATEGY COMPARISON RESULT

The average results that are obtained from five meeting scheduling simulation are presented in this section. Stra-1 and Stra-2 represents relaxation strategy and relaxation with preference estimation strategy respectively. Meanwhile, Stra-3 is the improved searching strategy proposed in this study.

Figure 7 shows the average preference level achieved for committed meeting that made by secretary agents. It reflects the quality of the solution that found by SA using difference strategy. Stra-2 and Stra-3 achieves higher average preferences compare to Stra-1. Characteristic of Stra-1 that always finds a quick solution and does not attempt to find optimal solution explain why it always obtains lower average preference level. Furthermore, Stra-3 achieves average preference level that is close to Stra-2, which is designed for optimization. In conclusion, Stra-3 improves the existing relaxation process (Stra-1) by obtaining solution that has higher average preference level.

Meanwhile, Figure 8 shows the closeness of solution found by agent to the optimal solution. The values zero for Stra-2 proved that Stra-2 always promise an optimal solution. While value achieved by Stra-3 shows the meeting scheduling outcome are very close to optimal value.

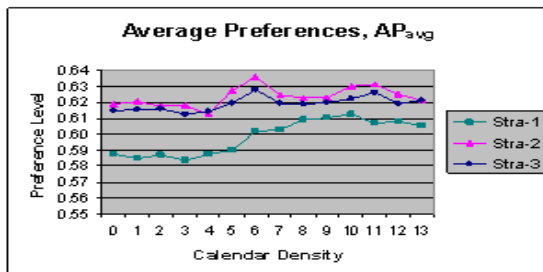


Figure 7: The Average Preference Level in Different Calendar Density

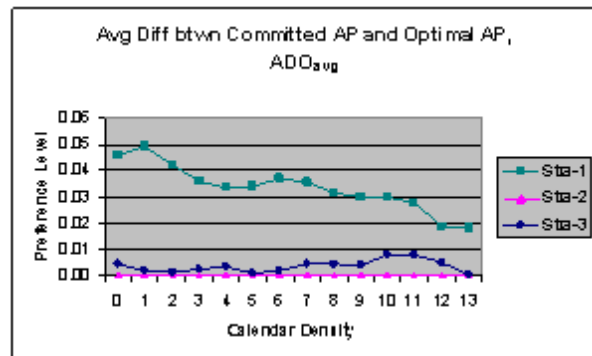


Figure 8: The Average Different between Committed AP and Optimal AP in Different Calendar Density

The number of negotiation cycle required to schedule a meeting is also compared to evaluate the communication cost of these searching strategies (Figure 9). The interesting result shows that communication cost for Stra-1 and Stra-3 are much lower compared to Stra-2. Stra-1 required lowest communication cost as it stops searching as soon as solution is found while Stra-2 required highest communication cost as it continues searching until it confirms a solution is the optimal one.

The lower communication cost for Stra-3 shows that network prediction can effectively propose a better proposal and provide shorter path to optimal or nearly optimal solution. Besides, the proposal successful rate for these three strategies are also evaluated (Figure 10). We found that, by average, Stra-2 achieves lower proposal success rate because the more the proposals need to be negotiated, the higher the possibility to get unsuccessful proposal. Meanwhile, Stra-3, which shows higher rate compare to both Stra-1 and Stra-2 suggested that Stra-3 always selects more acceptable proposal to negotiate with participants.

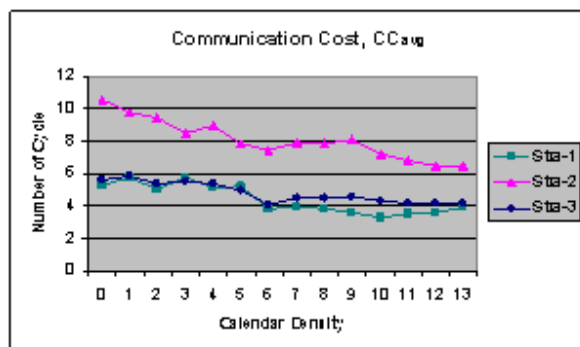


Figure 9: The Communication Cost in Different Calendar Density



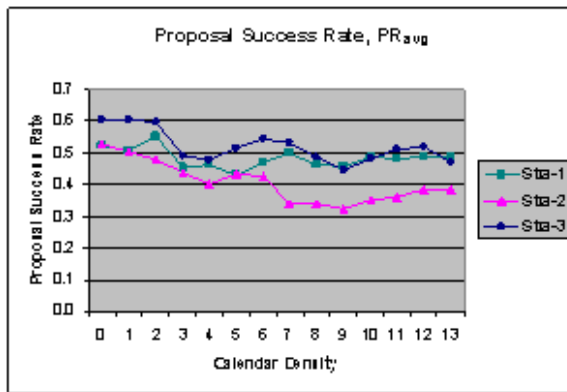


Figure 10: The Proposal Success Rate in Different Calendar Density

## 12. CONCLUSIONS

This study focused on implementation an agent-based meeting scheduling system. This system consist numbers of autonomous secretary agent that able to schedule meeting according to user preferences. This study interested in improving the strategy of meeting time searching process by extends the relaxation strategy with NN Model. Finally, the result of the simulations showed the ability of proposed searching strategy to find the timeslot that close to optimal solution. Besides, it requires fewer communications cost to achieve the solution. In conclusion, the proposed searching strategy improves both relaxation and relaxation with preference estimation searching strategy.

## REFERENCES:

- [1]. Chen, H., A. Houston, J. Numeraker, and J. Yen, 1996. Toward Intelligent Meeting Agents. *IEEE Computer*. vol.29. no.8. pp. 62-69.
- [2]. Chun, H.W. and Y. M. Wong, 2003. Optimizing Agent-based Meeting Scheduling through Preference Estimation. *Engineering Applications of Artificial Intelligence*, 16:727-743.
- [3]. Garrido, L. and K. Sycara, 1996. Multi-agent Meeting Scheduling: Preliminary Experimental Results. *International Conference on Multi-Agent System (ICMAS'96)*, Kyoto, Japan. pp. 95-102.
- [4]. Hassine, A.B., X. Défago, and T. B. Ho, 2004. Agent-based approach to dynamic meeting scheduling problems. *Intern. conf. on Autonomous Agents and Multiagent System*, New York, USA. ACM Press. pp. 1130-1137.
- [5]. Jennings N.R. A. J. and Jackson, 1995. Agent-based Meeting Scheduling: A Design and Implementation. *Electronics Letters*, 31(5):350-352.
- [6]. Lee, C.S. and C. Y. Pan, 2003. An Intelligent Fuzzy Agent for Meeting Scheduling Decision Support System. *Fuzzy Set and System*, 143:467-488.
- [7]. Mitchell, T., R. Caruana, D. Freitag, J. McDermott, and D. Zabowski, 1994. Experience with a Learning Personal Assistant. *Communication of the ACM*, 37(7), pp: 80-91.
- [8]. Oh, J. and S. F. Smith, 2005. Learning User Preferences in Distributed Calendar Scheduling. In *The International Series of Conferences on the Practice and Theory of Automated Timetabling*. pp. 35-49.
- [9]. Rumelhart D., G. Hinton, and R. Williams, 1986. Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1:318-362.
- [10]. Sangle S. and P.R. Babu, 2006. Neural Network Model for identification of societal preference of environmental issues. *International Journal of Environment and Pollution* 6. vol. 28. no. 3/4.
- [11]. Sugihara, K., T. Kikuno, and N. Yoshida, 1989. A Meeting Scheduler for Office Automation. *IEEE Transactions on Software Engineering*, 15(10):1141-1146.
- [12]. Sugumaran, M., S. Easwarakumar, and P.A. Narayanasamy, 2003. A New Approach for Meeting Scheduler using A\*-Algorithm. *Proceedings of the IEEE TENCON'03*, pp: 419-423.
- [13]. Sen, S., T. Haynes, and N. Arora, 1997. Satisfying User Preference while Negotiating Meeting. *International Journal of Human-Computer Studies*. Academic Press Limited, 47:407-427.
- [14]. Sen, S. and E. H. Durfee, 1991. A Formal Study of Distributed Meeting Scheduling: Preliminary Results. *Proceedings of the ACM Conference on Organizational Computing Systems*, 1991, pp. 55-68.
- [15]. Sen, S. and E. H. Durfee, 1994. On the Design of an Adaptive Meeting Scheduler. In *Proceeding of the 10th IEEE Conference on AI application*, pp: 40-46.
- [16]. Smith, R.G., 1980. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, 29(12):1104-1113.
- [17]. Tsuchiya, K., Y. Takefuji, K. Kurotani, and K. A. Wakahara, 1996. Neural Network Parallel

- Algorithm for Meeting Schedule Problems. Proc. for the IEEE TENCON'96, Digital Signal Processing Application, 1:173-177.
- [18].Woo, S.J., S.Y. Jeong and S.J. Geun, 1999. Cooperation in Multi-agent System for Meeting Scheduling. Proceedings of the IEEE Region 10 Conference (TENCON 99), 2:832-835.
- [19].Zhuang C., J. Ren, B. Gao, Z. Fu, 2005. Elicitation of decision maker preference by artificial neural networks. Neural Networks and Brain, ICNN&B'05. International Conference. vol. 3. pp. 1699-1703.