International Journal of Information Technology, Vol. 11 No. 4

# A Framework for Multi-Agent Negotiation System Using Adaptive Fuzzy Logic in Resource Allocation

Cheng Wai Khuen, Chan Huah Yong, and Fazilah Haron

School of Computer Science
Universiti Sains Malaysia, 11800 Penang, Malaysia

{khuen, hychan, fazilah}@cs.usm.my

## Abstract

A multi-agent system is a system which applies various autonomous agents to accomplish some specified goals. Such system is suitable for resource allocation problem since the nature of resource trading requires multiple agents to request for geographically dispersed heterogeneous resources. In order to support such a dynamic situation, a multi-agent system must be extended to accommodate agent with a high negotiation skill capability, that is the ability of learning from the requirements of resources advertised by the involving agents during each interactions. One way of achieving this is to analyze negotiant tactics and strategies in every counter-offer.

This paper addresses the above issue by using an Artificial Intelligent (AI) approach – Adaptive Fuzzy Logic (AFL) - in learning the behavior of other agents in the process of negotiation. The learning continues during each reasoning process on the subsequent reaction of negotiants. We present a framework of multi-agent negotiation with multi-issue (*e.g. price, time, resource conditions*) in the domain of resource allocation to illustrate the proposed approach. From our analysis, we found that the AFL gives a more reliable results compared to other methods in analyzing opponent tactics and strategies.

**Keywords**: multi-agent system, machine learning, Fuzzy Logic, resource allocation.

## I. Introduction

One of the main objectives of a distributed system (including grid computing) is to enable virtual resources sharing. For this reason, Resource Allocation (RA) becomes an important aspect of a distributed system. An efficient resource allocation technique is necessary in a frequently accessed large scale distributed computational resources. Besides, an efficient technique will increase resource utilization effectively, thus maximizing system utility or user profit.

An agent is a program that operates autonomously and accomplishes unique tasks without direct human supervision while a Multi-Agent System (MAS) is a computational system where several autonomous or semi-autonomous agents interact and cooperate or compete to perform some set of tasks or satisfy some set of goals. For this reason, the internal behavior of agents' interaction becomes an important concern in the multi-agent technology.

The idea of adopting learning ability in distributed system especially via MAS is not new. Many theoretical papers from machine learning and psychological perspective on learning in multi-agent environment have been published [7, 8, 9, 14]. However, there exist a need of a comprehensive learning model which supports multiple outstanding abilities, such as intelligence, optimization, comprehension and others. Among all the ability of learning, analyzing tactics and strategies extracted from an agents behavior during agents interaction, is still an open problem.

To perform an intelligent resource allocation module in a multi-agent environment, a machine learning method will be one of the important elements. A negotiator (or a negotiation agent) with a learning ability is essential since a resource broker promotes federalization between different users across heterogeneous domain in the grid environment. Most of the research on learning and negotiation in MAS [7, 8, 9] emphasize on the learning through the architecture of communication and interaction instead of learning the internal behavior and the content of communication, such as tactics and strategies. By understanding the tactics and strategies taken by agents in each attempt of negotiation, an agent will be able to develop its experiences and knowledge, thus, gain more benefit compared to other agents which are not. If an agent successfully applies the tactics and strategies, then it will give a great impact to agent's negotiation progression because it makes the agent in a never-lose position.

In [10], they design autonomous computational agents and their interaction via protocols for the task allocation had emphasized *negotiation tactic* must take into consideration of the environment in which the agent operates. Normative models of choice often ignore issues such as time pressures, resources and others. More specifically, tactics are the set of functions that determine how to compute the value of an issue (such as price, volume, duration, quality, etc), by considering a single criterion (such as time, resources or behavior of other agents). A *negotiation strategy*, on the other hand, is used to determine the course of action which will result in an agreement on a contract that maximize its scoring function, or in other word, how to prepare a counter-offer through negotiation. It also denotes the way in which an agent changes the weights of the different tactics over time.

The rest of the paper is as follows; section 2 discusses the related work on agent's behavior learning in the domain of resource allocation. Section 3 gives an overview of the negotiation model. Section 4 presents the methodology applied in our machine learning and reasoning for multi-agent's behavior analyzing through negotiation. Section 5 explains the evaluation criteria used in our work and section 6 concludes with future work.

## II. Related Work

Over the past few years, various approaches to resource allocation have been developed for distributed system [1, 2]. There is a rapidly growing body of literature on the subject, but question still arise as to whether a multi-agent system is able to adapt in the domain of RA [3, 4, 5]. These inquiries originate from the wide applicability and efficient used of MAS on heterogeneous domain. Hence, we focus our work on the adaptability and the efficiency of MAS in RA.

One of the most famous approaches in multi-agent behavior learning is Reinforcement Learning (RL). Many research have been done through the application of RL in multi-agent environment for various domains [11, 18]. RL allows the agents to learn its behavior based on the feedback from the environment, which can be learnt once and for all, or keep on adapting as time goes by. Besides, there are several advantages in applying RL in MAS. Firstly, there is a little need for human expert who knows about the domain of an application. Much less time will be spent designing a solution, since there is no need for hand-crafting complex sets of rules as with the expert system. Secondly, if the problem is modeled with care, it can converge to the global optimum. This is the ideal behavior that maximizes the reward. However, RL has encountered three major problems. First is perceptual aliasing – ignorant into perceiving two or more different states as the same state. The second problem is concurrent learning, due to limited perception, it is often impossible to fully determine the current state. The third problem is memory expensive to store values of each state, since a problem can be pretty complex which make an agent behave ineffectively.

Game theory is also a well-known negotiation approach in multi-agent. It has been implemented for one-to-one and one-to-many (auction) negotiation [6]. The major challenge facing by game theory is the way to design an approach to pay attention to its environment, instead of only a dictionary of optimal strategies for a stereotype set of game. A more intelligent way to decide the strategy used in each attempt of negotiation is needed. Furthermore, using game theory in learning has a constraint for implementation. Each strategy choose by a player must consider as rational move. Problem will occur if facing a hyper-rational player, which might randomly choose the counter-offer. The conventional game theory will be unable to model the dynamical element of the game play. Only games with simple setting able to solve by conventional game theory. The other criticisms on the theory's assumption include of beliefs are consider as common knowledge and individuals are optimizers and computationally unbounded. These issues raised arguments to their applicability in designing an intelligent agent.

Beside the above approaches, a Markov Decision Process (MDP) [12] – delayed reward reinforcement learning also has been proposed. This approach gives agent the ability to learn about other's strategies from past negotiation steps, via computation of strategy's transition probability for each counter-offer. In their perception, an agent will change their tactics over time to avoid the shortcomings of a deterministic and fixed negotiation strategy. However, the list of tactics applied in their system is assumed as a common knowledge, which means every agent will expose their tactics to public. It is not applicable in the real world negotiation system with privacy exposition and limited tactic applied. Moreover, the author only applied MDP for anticipating negotiation strategies but did not mentioned how to provide a precise value for a particular issue. Besides, if the MDP state space is excessive, then the belief states might not be explicitly represented. Moreover, the MDP unable to provides an approximately or precisely value for particular issue in reasoning process.

In [20], fuzzy similarity is used to compute tradeoffs among multiple attributes during bilateral negotiations. A fuzzy technique is used to find negotiation solutions that are beneficial to both parties (win-win situation). The way of applying fuzzy technique in [20] is different with our research work because we emphasize on competitive agents instead of collaborative agents. An agent first generates some potential contracts for which it receives a score $\theta$. After that, the agent finds the contract on the indifference curve for $\theta$, which has the maximum similarity degree to the last proposal from the negotiant. The proposed algorithm is designed to work well in a distributed setting for agent having limited information about the preferences of their negotiant and limited computational resources to devote to the negotiation process.

Fuzzy Logic has been a powerful problem solving methodology in the past few years. It provides a simple and easy way to draw a definite conclusion from ambiguous, imprecise or vague information [17]. Fuzzy Logic mimics the human decision making by providing a precise solution from approximate data. The Adaptive Fuzzy Logic (AFL) evolves by dynamically eliminating the useless rules in the fuzzy rules set. The proposed AFL aims to provide a more accurate, adaptive and reliable machine learning methodology, which will be discussed in the rest of the paper.

## II. Negotiation Model

### A. *Example Scenario*

Each user starts by registering a provider agent to handle resource offering or create a consumer agent to request for resources. Let $x$ ($x \in \{x_1, x_2 \ldots x_m\}$) represents the provider agent and $y$ ($y \in \{y_1, y_2 \ldots y_n\}$) the consumer agent. After registration of agent in a platform, user can fill in the details of the resource, which is ready for selling or buying. The details consist of multiple issues. Let $i$ ($i \in \{i_1, i_2 \ldots i_n\}$) be the issues under negotiation, such as price, volume, duration, level of security and so on. For certain important issues like price and volume, user can assign each issue with single tactic for monochromatic increase or decrease of the corresponding value during process of negotiation. Once the issue binds with a tactic, it cannot be changed by user unless the agent itself exchanges the information due to the condition of the negotiation. Besides, an overall strategy, $s \in \{s_1, s_2 \ldots s_n\}$ for negotiation must be determined after filling in the details. After confirmation, all the values for issues are set to unchangeable and the discovery and negotiation process will begin. After several attempts of offer and counter-offer between agents, the negotiation process will end up with two possibilities. It may be an agreement between provider and consumer as an ideal case, or a failure to meet the agreement because of the duration deadline.

Our proposed machine learning method for tactics and strategies learning is best located during the offer and counter-offer processes (*during communication*). This intermediate period is suitable for agent to learn the behavior of each others, which mimic the human interaction. We usually gain experiences through process of interaction with others. The more often we exchange contents, the higher the possibility of understanding others' behavior. The learning process might benefit from both successful and failed cases because agent will gain some experiences from the incident. The useful information will be stored into the database for later analysis to enhance both agent intelligence and negotiation skill.

### B. *Negotiation Assumptions*

Our proposed multi-agent negotiation model for tactics and strategies learning is based on the following assumptions:

1. Information privacy - Each agent can only access the information such as resource type, time, price, and condition of resource, during each attempt of counter-offer.

2. Time restrictions - Each agent must terminate if the deadline is met. No other agent can discover other agent's life-span. Each period of calling-for-proposal is fixed by the user when the agent is created to perform negotiation.

3. Privacy and uncertainty of tactics and strategies: Each agent will not realize which tactics and strategies being apply by others. Besides, agents can change the tactic and strategy applied during the negotiation process, according to the situation. This assumption is carried out to solve the problem of tactics and strategies become the common knowledge to every agent in negotiation platform.

### C. Negotiation Tactics

As mentioned earlier, the application of tactic on issues is best placed during the offer and counter-offer period. Each new value of an issue is generated using a mathematical function depending on the tactic. The work in [10] has categorized major tactics into three types.

1. Time dependent - A function that is able to model an agent which is likely to concede more rapidly as the deadline approaches.
2. Resource dependent – This is similar to the time dependent function except the issue time itself is replaced with the quantity of resources available.
3. Behavior dependent - A this tactic chooses to imitate opponent tactics to avoid itself from being exploited.

Apart from the above types, users can also generate some unique functions to be adapted on the particular issue for the negotiation process.

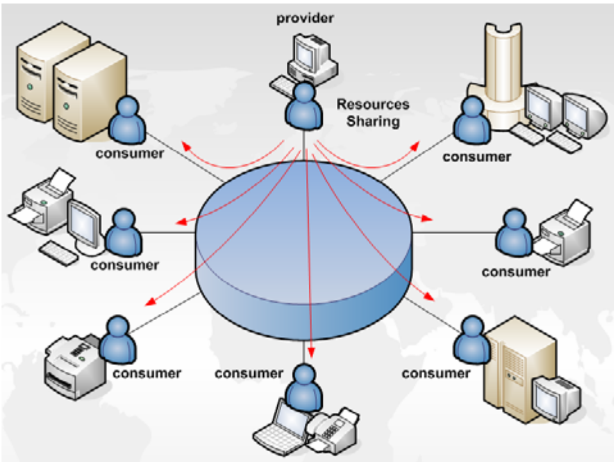


Figure 1: Each user can register himself as a provider of a resource or as a consumer requesting for resources.

### D. Negotiation Strategies

Strategy is a combination of weights of all the issues involved in a proposed negotiation offer or counter-offer. Before an attempt of a proposal being made to others, a strategy must be predefined and set. This helps agent to calculate and compare the incoming proposals. A proposal with maximum utility function, $U_{max}$ will be filtered out. The strategy also indicates the way an agent changes the weight and tactic for each issue over time.

### E. Utility Function

A consumer agent $x$, calculates the proposal's utility function, $U$ of provider agent $y$, using the following formula.

$$U^y = \sum_{1 \le i \le n} w_i^y f(v_i^y)$$

$U^y$ refers as utility function for provider agent, *y*. $w_i^y$ is the weight for corresponding issue, $i \in \{i_1, i_2 \ldots i_n\}$ in negotiation. It also denotes the importance of issues, *i* for agent, $y \sum_{1 \le i \le n} w_i^y = 1$. The scoring function, $f(v_i^y): [\min_i^y, \max_i^y] \to [0,1]$, indicates the value for issue, $v_i^y$ acceptable by agent, *x* from agent, *y*.

Every agent has its own minimum utility function, $U_{min}$ as a guideline to evaluate counter-offers provide by others. Only proposal which exceed the $U_{min}$ will be processed and considered for further action. Normally, a consumer agent will choose the proposal with maximum utility for itself, $U_{max}$ from a list of provider's proposal to successfully deal with a contract and vice versa.

## IV. Agent Negotiation Using Adaptive Fuzzy Logic

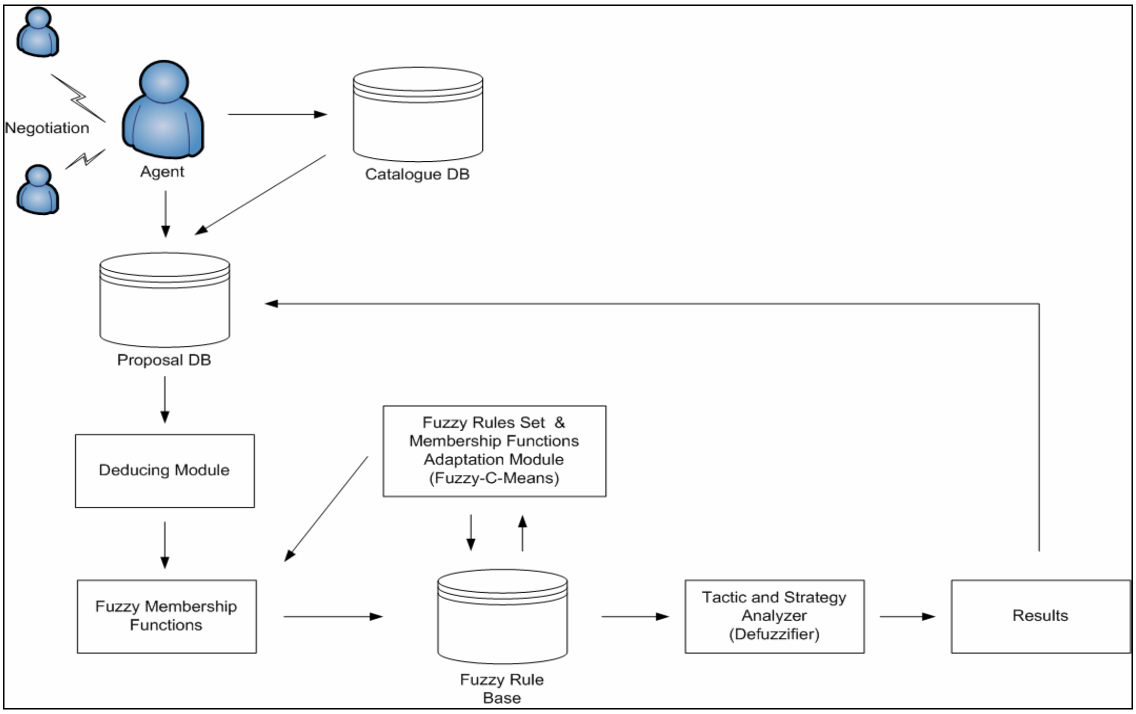


Figure 2: An Overall Architecture of Adaptive Fuzzy Learning and Reasoning Process.

A good machine learning module in MAS provides facility for agents to communicate and cooperate in order to learn effectively, and also enhances their overall performance in decision making process. By adapting the machine learning and reasoning module in multi-agent system, it may help in a better understanding of other agents' behavior. Thus, improve the agent itself in terms of skills on negotiation. This approach will give facility for an agent to easily filter out the non-potential negotiants because it can identify the behavior of other agents from past experiences. If agent can learn about others' tactics and strategies, then it will be able to propose a more beneficial counter-offer in a contract.

### A. *An Architecture of the Multi-Agent System*

The overall architecture of our adaptive fuzzy learning and reasoning is shown in figure 2. The architecture is divided into three parts:

- Learning module which learns opponents' behavior in tactics and strategies from counter-offers.
- Reasoning module which deduces and provide a possible proposal to be counter-offered in decision making process.
- Adaptation module which evolve the fuzzy membership functions and fuzzy rule base to maintain the applicability of our approach according to environment condition.

In our opinion, an agent must have knowledge of its historical experiences in negotiation to increase its negotiability skill. For this reason, we propose to equip each agent (*provider/consumer*) with three databases – catalogue, proposal and fuzzy rule base. Catalogue's database contains a list of resource offerings or requirements by an agent over time. Proposal's database will be responsible to store other agents proposal or calling on resources. Fuzzy rule base is the intelligent module of the agent to support agent in decision making for tactics and strategies.

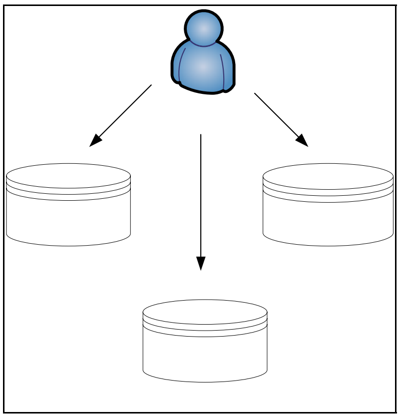

Figure 3: Each agent interacts with three databases – catalogue, proposal and fuzzy rule base.

We use Java Agent DEvelopment Framework (JADE) [13] as our platform in developing the multi-agent system. Java language is used to implement both types of our agent (provider/consumer). JADE agent platforms have containers to hold agents. Each platform may consist of many containers located on different computers. The main container resides on the host. It runs the platform's Remote Method Invocation (RMI) server. Agents on various containers on a platform use the RMI protocol to communicate.

### B. *Learning Module*

In our negotiation model, we will apply the consumer perspective to illustrate the application of AFL in the resource trading. To start a negotiation, the consumer will prepare the first request's proposal and pass to the catalogue's database. So, the catalogue's database is responsible to store value of each issue in every offer/request proposed by an agent. This database is useful for an agent to trace back the previous offering or requesting resources. In other word, it will act as the historical bank of agent's offer/request for future references. After the consumer request for particular resource, multiple counter-offers will be proposed by interested providers. The proposal's database is in charged of storing the entire transaction or counter-offer from other agents. The data stored into this database may probably include sender name, resource type, calling receiving time, propose/counter-offer price by sender, offer's price by receiver, status (success, counter-offer and failed) and more. After some analysis, we discover that the collection of continuous counter-offer from a corresponding provider always tend to present an unseen form of behavior, as shown in figure 4. This useful information enables the intelligent agent to deduce some variables to construct fuzzy rule set. The third database, fuzzy rule base, is used as our core knowledge base in the methodology. We use the fuzzy rule base as a knowledge base because it stores all others agents' behavior and provide certain level of knowledge to our intelligent agent in fuzzy inference process. Each single record in the fuzzy rule base is the combination of linguistic variables from various corresponding fuzzy membership functions, which can construct fuzzy rules for reasoning process. Furthermore, each single fuzzy rule set in the rule base is represented as a scenario of negotiation process between our intelligent agent and its opponent. In other words, with further fuzzy rule set gather by our intelligent agent, more expertise will be handled by our intelligent agent to perform in the domain of resource allocation.

As illustrated in figure 4, the deducing module is created to infer some hidden information from proposal's database and responsible as the first learning component in our architecture. By analyzing two continuous proposals from the same agent, we will gather some indication of value's changing behavior with the different of values in two proposals. The information deduces from the list of proposals will then provides some useful hints in reasoning. For instance, calculation on continuous time issue for a corresponding agent in calling for a resource allows the intelligent agent to predicts in the following calling time, and also indicate the status of the opponent's proposal in the platform. If the proposal is acceptable, then no more proposal calling is necessitate, unless abandoned by the public. A deductive algorithm is shown in figure 5 below, as an example. Beside time issue, the changes of important issue such as price can let the receiver agent derive some conclusion. A constant or sudden change of the value depends on the applied tactic (predefined function). Thus, the movement and tendency of the value provide agent a prospect to predict for the next negotiation. In our deducing module, there will be 6 issues to be analyzed in the domain of resource allocation, which includes duration's changing rate, price's changing rate, volume of resource's changing rate, level of performance, level of authority and level of security. These issues are the most preferred criteria for the users in the domain of distributed system.

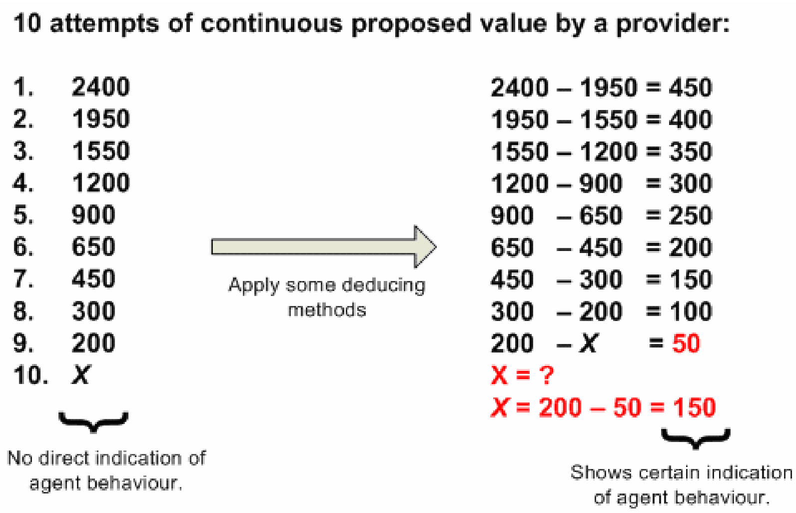


Figure 4: Example use of deducing module in our methodology.

After the deducing process, all the gaining information will be sent to the corresponding fuzzy membership function before storing in the fuzzy rule base. Every single useful information from the deducing module will have a particular corresponding fuzzy membership function, *f*, such as callingDuration, priceChangeRate, amountUsage, level of performance, level of authority and level of security (figure 6). Besides, each single membership function will consist of several linguistic input variables, *v*, such as low, medium and high, which is predefined (every agent has multiple membership functions, $f_i$, where $1 \leqslant i \leqslant n$ and n is number of issues). All the linguistic variables are set according to the human expert's perspective on the description to a particular negotiation issue. Although the triangular shape of fuzzy membership function is under standard setting at the initial stage, but our negotiation model applied a FCM to adapt our fuzzy membership function from time to time. Further description on adaptation module can be found in Section 4.4.

*IF senderName1 == senderName2 AND*
*mercType1 == mercType2 AND*
*status1 == status2 == failed*
*THEN*
*priceChangeRate = |(price2 – price1) / price1|.*

Figure 5: Sample algorithm for deducing module to extract "price's changing rate" from proposal's database.

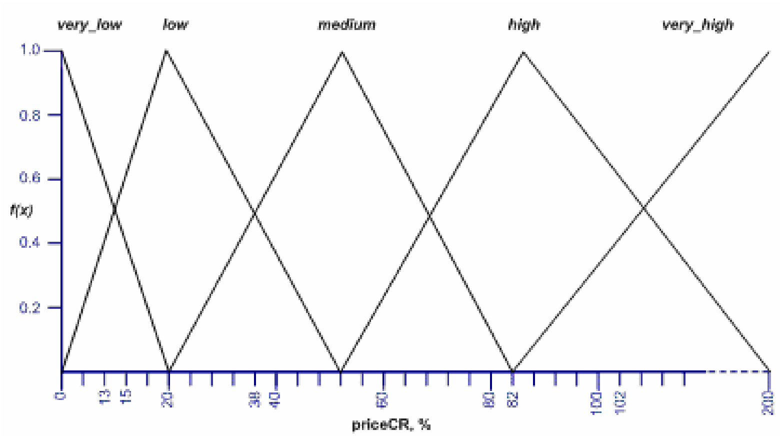


Figure 6: Sample of priceCR's fuzzy membership function.

The deducing results for each proposal after process by fuzzy membership functions will send to fuzzy rule base to construct rules, which represent each condition of counter-offer in the negotiation. But each rule in the fuzzy rule base will set without consequence part. The consequence part for each rule is depends on the request of tactic and strategy analyzer. Since we prepared 6 variables (time, price, volume, performance, authority, and security) for each fuzzy *if-then* rule, then we have a rule with 5 antecedents plus 1 subsequent. Each antecedent will sequentially acts as subsequent for reasoning. For instance, if we hope to figure out the possible price of a resource with certain level of resource condition, the rule will be – "*If calling duration is high, amount usage is low, level of performance is good, level of authority is high and level of security is good, then what is the possible price?*". This question will be answered after the reasoning process by applying the defuzzification method in our methodology. In addition, each rule will bind with a counter to record the firing times of each fuzzy rule for further adaptation process. It helps to confirm the validity of rules from time to time, which will eliminate the rules with lesser confidence value currently. A threshold value with standard setting will be set to determine the minimum confidence value. However, this threshold value will automatically adjust according the way shown in figure 8. We set the minimum confidence value lower than 20% according to the reliability of decision after several round of experiments.

*1. IF callingDuration is constant AND priceChangeRate is constant AND …*
*THEN volume is high. Counter = 2*

*2. IF callingDuration is low AND priceChangeRate is high AND …*
*THEN volume is medium. Counter = 4*

*3. IF callingDuration is low AND priceChangeRate is high AND …*
*THEN volume is high. Counter = 6*

.
.
.

*N. IF callingDuration is high AND priceChangeRate is low AND …*
*THEN volume is low. Counter = 1*

Figure 7: Example of volume's fuzzy rule set.

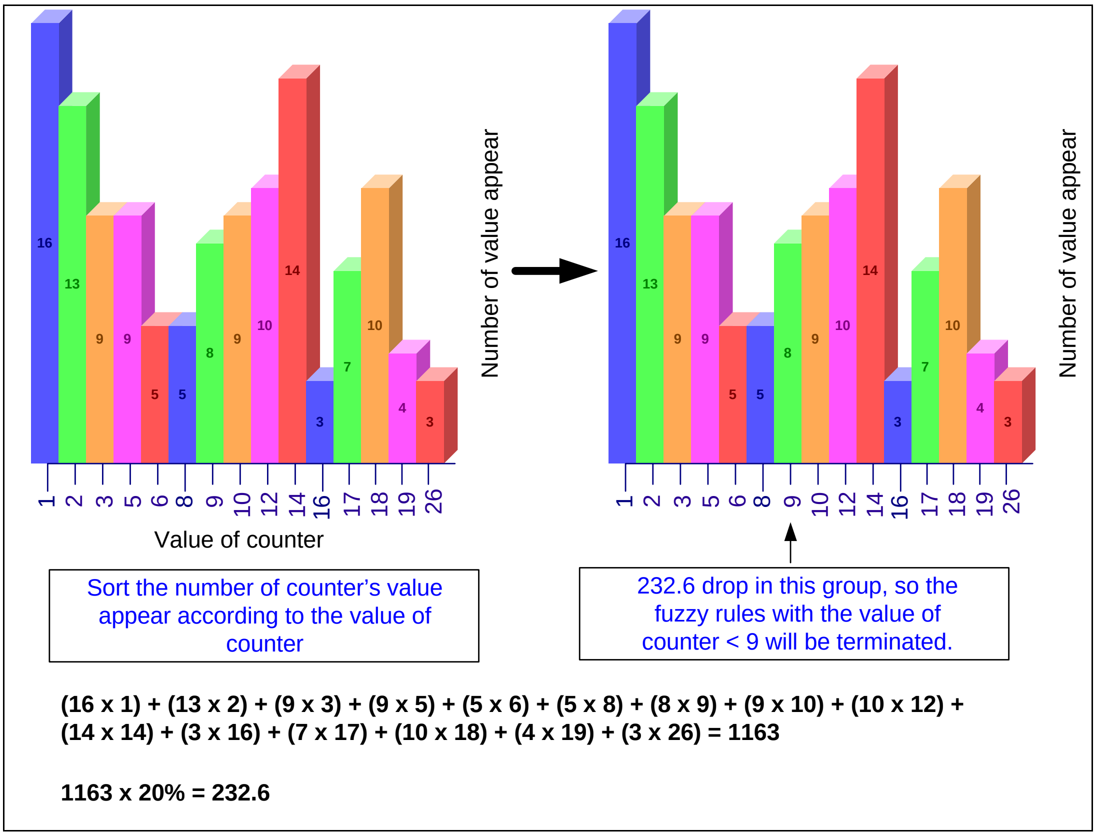


Figure 8: Method to determine the minimum confidence value.

### *C. Reasoning Module*

For the reasoning module, the tactic and strategy analyzer (fuzzy inference engine) equips with inference ability to figure out the most possible value for corresponding issues. Sometimes it is useful to just examine the fuzzy subsets that are the result of the composition process, but more often; this fuzzy value needs to be converted to a single number – a crisp value. This is what the defuzzification does. From the various defuzzification methods, mean of maximums, centre of gravity and weighted average, we choose to apply the centre of gravity (centroid) as defuzzification method because its reliability in average performances, as shown in the technical report [15]. In the centroid method, the crisp value of the output variable is computed by finding the variable value of the centre of gravity of the membership function for the fuzzy value. The formula of centroid is shown below where *w* is the discrete element of the fuzzy number to be defuzzified.

$$C = \frac{\int w\mu(w)dw}{\int \mu(w)dw}$$

After the fuzzy inference engine finishes the reasoning process, a counter-offer with the result of possible value for corresponding issues will be proposed to the opponent agent and waiting for further responses. We also can combine the composition and defuzzification processes, in taking advantage of mathematical relationships that simplify the process of computing the final output variable values.

Up to this point, we can say that our architecture imitates the human expert learning and reasoning processes. Firstly, an agent will learn their opponents' behavior. From time to time, the experiences of an agent will grow incrementally. Secondly, with the available experiences, an agent is able to deduce or provides decision making in an expert mode. The overall tactic and strategy applied by negotiant can be analyzed through the trend of differentiation between continuous values for corresponding issues in counter-offer. Furthermore, if our intelligent agent communicates with new negotiant, the agent at least able to provides a reference in counter-offer by applied others closely similar condition. Historical experiences in fuzzy rule base will contribute as a part of the decision maker for further negotiation process.

### *D. Adaptation Module*

As mentioned earlier, the content of fuzzy rule base is sequentially constructed. Once the number of rules achieves the predefine threshold value, the fuzzy rule base will call for adaptation process to reconstruct the fuzzy membership functions and fuzzy rule sets. In this adaptation module, a Fuzzy-C-Means (FCM) algorithm is applied to refine the shape of fuzzy membership function [21], which will be more up to date and applicable in terms of our intelligent agent's knowledge content. The function of FCM is shown below, where $m$ is any real number greater than 1, $u_{ik}$ is the degree of membership of $x_k$ in the cluster $i$, $x_k$ is the $k$th of d-dimensional measured data, $v_i$ is the d-dimension center of the cluster, and ||*|| is any norm expressing the similarity between any measured data and the center.

$$J = \sum_{k=1}^{n} \sum_{i=1}^{c} \mu_{ik}{}^{m} \left\| x_k - v_i \right\|^2$$

From the work of [16], we understand that FCM is able to extract fuzzy membership functions from user data. The fuzzy rule base that we use for tactic and strategy analyzing will always be applicable and reliable because the knowledge contents of our intelligent agent always evolve and validated. However, there is still room for us to improve the way to apply FCM into the AFL. Specifically, on how to define values for accuracy and fuzziness in the FCM. At the moment, the values that we use for accuracy is 0.3 and fuzziness is 2.

## V. Evaluation Criteria

As mentioned earlier, there are three criteria to evaluate our proposed machine learning methodology. The first criterion is accuracy. This will evaluate the similarity of values in counter-offer by the sender and the computed result by the receiver after the tactic and strategy analyzer. The difference between counter-offer can be measured by the formula below, where $n$ is the number of corresponding issues, $\Delta m$ is the difference of issue's value and $w$ is the weight defines in negotiation strategy for corresponding issue.

$$Difference, d = \sum_{i=1}^{n} \Delta m_i w_i$$

The lower the value of $d$ the more accurate the inference proposal is in counter-offer. The second criterion is adaptability. This criterion shows the adaptation capability of our methodology in analyzing the different types of tactics locally, and the capability of our intelligent agent with AFL module to survive in a scalable grid environment. The types of tactic recognized by our methodology are not common knowledge or predefined but they are derived from the historical experiences, which is more reliable. Reliability is the third criterion of our evaluation target. Reliability can be measured by testing the capability of our intelligent agent to recover when unexpected interrupt in a negotiation occurs. Alternatively, the results obtained from the fuzzy inference system can be used to evaluate reliability as well. In general, the result based on a set of rules is more assured and traceable since every result from the analyzer is deduced according to the agent's experiences. Thus, we believe that the reliability of our proposed approach is always higher.

## VI. Conclusion and Future Work

This paper describes the framework of a multi-agent system using Adaptive Fuzzy Logic in learning other agents' tactic and strategy within the domain of resource allocation. The learning module of AFL provides a reliable result because the final outcome is derived from the historical experiences. In order to evaluate the applicability of AFL in real multi-agent negotiation environment, we intend to apply the idea of agent negotiation with AFL on a grid enabled test-bed. Our future works include experimenting our approach using three metrics, namely, accuracy, adaptability and reliability. We will also compare the applicability of AFL with other approaches such as reinforcement learning, game theory and others.

## References


[1] K. Czajkowski, I. Foster, and C. Kesselman. "Resource Co-Allocation in Computational Grids", *In Proc. 8th IEEE Symp. On High Performance Distributed Computing*. IEEE Computer Society Press, 1999.

[2] I. Foster, C. Kesselman, C. Lee, B. Lindell, K. Nahrstedt, and A. Roy. "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation", *In Proceedings of the International Workshop on Quality of Service*, pages 27-36, 1999.

[3] Yong Gong, Li Yao, Wei-ming Zhang, Ji-chang Sha, and Chang-ying Wang. "Methods for Resource Allocation via Agent Coalition Formation in Grid Computing Systems", *Proceedings of the IEEE, International Conference on Robotics, Intelligent System and Signal Processing*, China, 2003.

[4] A. Galstyan, K. Czajkowski, and K. Lerman. "Resource Allocation in the Grid Using Reinforcement Learning", International Conference on Autonomous Agents and Multiagent Systems, 2004.

[5] Junwei Cao, Daniel P. Spooner, Graham R. Nudd, et al. "Agent-based Resource Management for Grid Computing", *In Proceedings of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2002)*, Germany, 2002.

[6] Ken Binmore and Nir Vulkan. "Applying Game Theory to Automated Negotiation", *paper prepared for DIMACS Workshop on Economics, Game Theory and the Internet*, 1997

[7] Eduardo Alonso, Mark d'Inverno, Daniel Kudenko, Michael Luck, and Jason Noble. "Learning in Multi-Agent Systems", *The Knowledge Engineering Review*, Cambridge University Press, 2001.

[8] P. Brazdil, M. Gams, S. Sian, L. Torgo, W. Van de Velde. "Learning in Distributed Systems and Multi-Agent Environments", *in Machine Learning: EWSL-91 (European Working Session on Learning), Y. Kodratoff (Ed.), Lecture Notes in Artificial Intelligence*, Springer-Verlag, 1991.

[9] G. Weiss and P. Dillenbourg. “What is ‘Multi’ in Multi-Agent Learning?”, *In P. Dillenbourg (ed.), Collaborative Learning. Cognitive and Computational Approaches. Pergamon Press*, 1999.

[10] Peyman Faratin. “Negotiation Among Groups of Autonomous Computational Agents”, *Int. Journal of Robotics and Autonomous Systems*,1998

[11] S. Arai, K. Sycara, and T. Payne. “Experience-based Reinforcement Learning to Acquire Effective Behavior in a Multi-Agent Domain”, *The Sixth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2000)*, Springer-Verlag, 2000.

[12] F. Teuteberg and K. Kurbel. “Anticipating Agents' Negotiation Strategies in an E-marketplace Using Belief Models”, *Proceedings of 5th International Conference on Business Informations Systems BIS 2002*, Poland, 2002.

[13] F. Bellifemine, G. Caire, T. Trucco and G. Rimassa. “JADE Programmer’s Guide”, *Copyright (C) 2003 TILab S.p.A*, 2003.

[14] Dajun Zeng and Katia Sycara. “Benefits of Learning in Negotiation”, *Proceedings of AAAI-97*, 1997.

[15] F. S. Smith. “Selection of Fuzzification. Inference and Defuzzification Techniques for Fuzzy Logic Control”, *Technical Paper RP01-145*, Society of Manufacturing Engineers, 2001.

[16] Doctor, F., Hagras,H.A.K., Callaghan,V., Lopez, A. “An Adaptive Fuzzy Learning Mechanism for Intelligent Agents in Ubiquitous Computing Environments”, *Proceedings of the 2004 World Automation Conference*, Seville, Spain, 2004.

[17] Luger, G.F. “Artificial Intelligence: Structures and Strategies for Complex Problem Solving”, *Addison-Wesley*, 850 pages, Fourth Edition, London, 2002.

[18] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In Proc. of 10th ICML, pages 330-337, 1993.

[19] Arun D. Kulkarni. “Computer Vision and Fuzzy-Neural Systems”, *Prentice Hall, pg. 378-382*, New Jersey, 2001.

[20] Faratin, P., Sierra, C. and Jennings, N. R. “Using Similarity Criteria to Make Negotiation Trade-Offs”, In *Proceedings of 4th International Conference on Multi-Agent Systems (ICMAS-2000)*, pages pp. 119-126, Boston, USA, 2000.

[21] Arun D. Kulkarni. “Computer Vision and Fuzzy-Neural Systems”, *Prentice Hall, pg. 378-382*, New Jersey, 2001.

Mr. Cheng Wai Khuen is a Masters by research student in the School of Computer Sciences, Universiti Sains Malaysia (USM). His main research interest is the grid computing. His study focuses on the problem of resource allocation in grid computing using intelligent software agents.

Dr. Chan Huah Yong is a senior lecturer in the School of Computer Sciences, Universiti Sains Malaysia (USM). He is actively involved in grid computing research activities, both, at the national and international level. He is also one of the project leaders of the *e-Science Grid* pioneer project in Malaysia, funded by the Ministry of Science, Technology and Innovation. His research spans from grid computing, to a more specific issues such as resource allocation, load balancing, software agents, and middleware engineering.

Dr. Fazilah Haron is a senior lecturer in the School of Computer Sciences, Universiti Sains Malaysia (USM). She is an active member of the parallel and distributed computing research group. She initiated the grid computing research in the department and has been involved in grid-based activities at the national and international level. She is one of the key researchers in the pioneer grid project, the *e-Science Grid* which is funded by the Ministry of Science, Technology and Innovation, Malaysia. Her research interests include grid computing, parallel and distributed processing, and peer-to-peer computing.