# Degree of Satisfaction in Agent Negotiation

K-M Chao, M. Younas, R. Anane
*DSAM Research Group*
*School of MIS, Coventry University, UK*
*{k.chao, m.younas, r.anane}*
*@coventry.ac.uk*

C-F Tsai
*Department of IM,*
*Aletheia University,*
*Taiwan*
*tsai@email.au.edu.tw*

V-W Soo
*Department of CS*
*National Tsing- Hwa*
*University, Taiwan*
*soo@cs.nthu.edu.tw*

## Abstract

*Agent negotiation over multiple issues is often seen as the process of searching for a solution in a complex and large space. Depending on the negotiation mechanism such search space can be dynamic wherein agents may be cooperative or non-cooperative. In a realistic negotiation, agents are unwilling to reveal their utility functions to their opponents or collaborators. These important characteristics of negotiation increase the complexity of the design of efficient and effective negotiation agents. In this paper, we propose a novel approach that combines a co-evolutionary mechanism with the notion of degree of satisfaction. The former effectively searches the space, while the latter improves negotiation efficiency. Agents under the proposed scheme can carry out co-operative, or non-cooperative, without revealing their utility functions. The proposed approach is implemented as a prototype system and evaluated through a number of experiments. The evaluation shows the effectiveness of the proposed approach in cooperation and non-cooperation.*

## 1. Introduction

The number of companies using E-business to support their business operations is increasing dramatically. However, current E-business technology lacks flexibility and offers a limited scope to participating parties, especially, for negotiation over business matters such as product prices, quantities, quality, and delivery time [14]. It is therefore desirable to employ an automated negotiation mechanism in E-business that can improve quality of service, time efficiency, and the accuracy of complex cost functions.

Automated negotiation as a field of research has attracted a number of research groups in particular to areas involving work on the simulation of human negotiation in a computing environment [1,2,3,4,5,6,7,8]. The introduction of agent technologies and the use of agents, acting as delegates to users, is an attempt to conduct negotiations over conflicting issues between participating parties. This research aims at improving existing automated negotiation techniques in order to facilitate practical negotiation by agents in E-business. Various theories and methods are proposed with the aim of facilitating automated negotiation in E-business. These include: argumentation approach [9.10], game theory [4,11,12], and Genetic Algorithms (GA) [13,14,15]. Argumentation–based approaches require further exploration in negotiation protocol in order to provide flexibility in decision-making [7]. Game theory models often assume that all possible deals are available to agents and no computation is required to find mutually acceptable solution [7]. However, these assumptions are not rational in the real world cases. The Nash bargaining solution provides great insight into bargaining behaviours, but it is limited to a one-stage bargaining game [8]. It does not consider multiple issues and incomplete information bargaining games. The Alternating-Offer model [8] allows agents to take turns to make offers until the agreement is secured. This model employs the sub-game perfect equilibrium concept and the Nash bargaining solution to support the theory. This model still assumes that the agent's utility functions are common knowledge. However, it cannot deal with multiple issues simultaneously. Further, there exist a number of issues in GA that need to be investigated. Oliver [13] first proposed a co-evolutionary approach to automated negotiation by relaxing a number of assumptions imposed by game theorists. The multiple issues and thresholds are encoded in the chromosomes with random combinations throughout the co-evolutionary process to find the optimised agreed solution for negotiating agents. The consequence is that agents may accept a worse deal than a previous one. Peyman [15] and Matos [14] enhance this concept by introducing a set of family tactics in order to search for optimised generic negotiation strategies. Since negotiation issues may vary in different cases the utility functions may change for the agents involved in negotiation. Due to the diversity in dependent utility and


0-7695-1969-5/03 $17.00 © 2003 IEEE

fitness functions, the best generic strategies do not guarantee good performance in all environments. Furthermore, these models often select outcomes (deals) that are sub-optimal, due to their applications to non-cooperative negotiation. It is generally difficult (if not impossible) to predict precisely how the system and the constituent agents will behave in a wide variety of circumstances [7]. Another barrier to applying co-evolutionary approaches such as GA to solve negotiation problems is that the solutions are built upon a known Nash equilibrium, and then refined them into better outcome [7]. This is not a practical assumption for most real world applications. However, the resulting payoff matrix may have one Nash equilibrium, multiple Nash equilibria or none at all when different negotiation strategies are employed. Even though mixed strategies can be used to identify a unique Nash equilibrium (if the payoff table is large) finding mixed strategy Nash equilibrium is difficult.

In order to overcome the above limitations, we have proposed an integrated negotiation mechanism in our previous work [1,2,3] that combines a co-evolutionary method with a game theory approach [16]. After a number of iterations, a solution, which is a Pareto-efficiency and is close to a Nash equilibrium, is found. This approach has led to some contributions to agent-based automated negotiation and has been applied to different problem domains [17,18]. However, this approach is limited to non-cooperative negotiation. In this paper we propose a new approach, which is capable of supporting cooperative or non-cooperative negotiation.

The paper is structured as follow. Section 2 describes the requirements of an automated negotiation mechanism. Section 3 presents the proposed automated negotiation approach, which is based on the requirements listed in Section 2. Section 4 illustrates a case study from an industrial collaborator in order to justify the need for an automated negotiation mechanism. Experimental results and their analysis are discussed in Section 5. Section 6 concludes the paper and defines future work.

## 2. Automated negotiation: a requirements perspective

The design of an automated negotiation mechanism requires the consideration of the type of game that the mechanism is going to support. There are two types of games: cooperative and non-cooperative. Cooperative game theory focuses on joint efficiency and payoff distribution while non-cooperative theory focuses on strategy and enforcement. Below we introduce some of the criteria that need to be considered.

In a cooperative game the negotiating agents mutually search for the common highest payoffs by cooperative methods, and then select one agreement from the set of payoffs. The main issues in this type of game are joint efficiency and payoff distribution. In order to achieve joint efficiency negotiating agents seek a mutually beneficial set of feasible payoffs, and select an optimal contracts or treaty. The agents have to choose an element from the set of payoffs when the mutually most efficient outcome has been identified.

In the non-cooperative game, the negotiating agents apply the strategies to achieve their goals, and have to implement the identified join strategy. This type of games focuses on coordinating negotiation strategies and implementing the agreement. The negotiators have to identify the particular strategies to affect the outcome. Each negotiator interacts with others so as to maximise his/her own payoffs.

When the negotiators choose a particular joint strategy, they have to enforce it or to guarantee to implement the agreed join strategy. This could be resolved by introducing penalties, so the negotiator who deviates from the agreement is not be better off. The negotiators are aware of that they will obtain fewer payoffs if they deviate from the agreements. In other words, the chosen strategy is not dominated by others. In addition to the nature of games, other criteria should be taken into account. These include:

**Efficiency:** The system should ensure that participating agents would reach an agreement in shortest time, if there is any.

**Effectiveness:** If there is an agreement between agents, a stable equilibrium in the end of negotiation should be produced in order to ensure that agents do not deviate from their agreements.

**Private utility function:** The utility function is used to determine how good the offer and counter offer are. It also provides essential information for agents to decide whether the offer should be accepted or not. Thus it is important to keep it private to the agent.

**Multiple issues:** An automated negotiation should allow agents to express the issues that they want to negotiate. Multiple issues normally form a large and complex search space. Thus an effective search algorithm is required to explore the space.

## 3. The proposed approach for automated negotiation

This section presents definitions and the negotiation process of the proposed approach. Negotiation process in the proposed method is supported by a number of components and their interactions. The approach includes business strategic objectives, utility functions, and negotiation strategies to support decision-making in the negotiation process. The level of cooperation and a degree of satisfaction provide a way for negotiating agents to speed up the negotiation process. The measurement of the degree of satisfaction is the difference between the payoff resulted from the desired outcome and the payoff resulted from the current offer. Negotiating over a number of issues can be seen as a part of search space. The co-evolutionary mechanism is used to explore the search space and to support the negotiation process over multiple issues. The combination of the co-evolutionary mechanism and the aforementioned components is designed to meet the criteria described in the previous section.

The co-evolutionary system is formed by two GA systems. The detailed description of the co-evolutionary mechanisms is described in [2, 3]. GA is a powerful search algorithm that can effectively produce an optimized solution over large and complex search space through the evolutionary process. The possible solutions and/or problems are encoded in the chromosomes. The system can randomly generate a set of possible solutions to the problems that form a population. A utility function is used to evaluate these solutions in order to determine how good they are. Normally two most preferred chromosomes are selected as parents and are used to generate offspring through mutation and crossover operations. After a number of generations, an optimized solution can be found. In this research, we exploit this characteristic to explore a large and complex negotiation search space. We also introduce the notion of a degree of satisfaction for the negotiating agent in order to speed up the negotiation process. The method may include a number of business Strategic Objectives (*SO*) such as increasing market share, capital turnover, stock clearing, and marginal profits for the selection of the utility functions. These are formulated as a set *SO,* where $SO=\{so_1, so_2, so_3, so_i, \ldots, so_j\}$, and $1 \leq i \leq j$. The decision maker prioritises *SO* according to his/her preferences such that $so_1 \succ so_2, \succ so_i, \succ \ldots, \succ so_j$. The selection of $so_i$ is through a function D which requires two parameters, i.e., negotiation feedback *f* and strategic objectives *SO*, thus $so_i \leftarrow D(f, SO)$ and $1 \leq i \leq j$. The function for selecting a cost function, denoted as $C$: $u_i \leftarrow C(so_i, U)$, depends on the available cost functions represented as $U=\{u_1, u_2, u_i, \ldots u_j\}$ and $1 \leq i \leq j$ and the chosen strategic objective *SO*. The negotiation strategy (denoted as S) generates a series of offers, $\delta$, by considering the opponent's degree of satisfaction, its target utility, and the time constraint. $\delta$ includes a number of issues, $I=\{i_1, i_2, \ldots, i_k\}$. Two negotiating agents denoted as $Ag_1$ and $Ag_2$ have the same set of issues, *I*, over which they are negotiating. An agent evaluates an offer, $\delta$, with its utility function $u_i(\delta)$ to gain the utility $\mu$: $Ag_1 u_i(\delta) \rightarrow Ag_1 \mu$.

In the proposed approach, each agent has a lowest expected utility gain (as a threshold) denoted as $Ag_i \mu_{min}$ and a highest expected utility gain $Ag_i \mu_{max}$ as an ideal utility to obtain for Agent$_i$, where $i=\{1,2\}$. If the utility $Ag_i u_j(\delta)$ of the offer is greater than $Ag_i \mu_{max}$, then agent accepts the offer, i.e., $Ag_i u_j(\delta) - Ag_i \mu_{max} > 0$. The degree of its satisfaction with respect to the deal, $\delta$, expressed by the agent is computed as follow:

$$Ag_i\, Score\,\delta = (Ag_i \mu_{max} - Ag_i u_j(\delta))/Ag_i \mu_{max}$$

Where $Ag_i Score\,\delta$ represents the degree of satisfaction. When the agent makes a counter offer $\delta'$, the $\delta'$ associated with $Ag_i Score\,\delta$ will be passed on to its opponent. The agent will take the information into consideration with $Ag_i Score\,\delta$, and the chosen negotiation strategy that includes resource and time factors for generating the next proposal. The negotiation strategy consists of a number of expected gain utilities $\mu_1, \mu_2, \mu_i, \ldots \mu_j$ before the expected utility $Ag_i\ \mu_{max}$ is reached. In addition, the agent can express the different levels of cooperation by assigning various weightings to a degree of satisfaction received from its opponent and its expected utility. This is to give the negotiating agents guidance for making offers and counter offers. The evaluation of the best strategies is based on the actual gained utility from the deal $\delta$, so a function $E(Ag_i\, S, Ag_i \mu_{max})$ for searching for the best strategy $Ag_i\, s_i$ is introduced.

$G(Ag_i s_i, Ag_i Score\,\delta)$ is an evolutionary function for generating a set of possible counter offers represented as $\xi=\{\delta_1, \delta_2, \delta_3, \ldots \delta_n\}$. $Ag_i s_i$ is a selected negotiation strategy for Agent$_i$. The fitness of $\delta$ in $\xi$ is determined by the distance between $Ag_i u_i(\delta)$ and the expected utility $Ag_i \mu_i$ that is given by $Ag_i s_i$. The selected next counter offer from the set $\xi$ will be the closest to the offer proposed by the other agent. The process can be expressed as follows: $Ag_1 \delta'_i \leftarrow Max(similarity\,(Ag_1 \delta, Ag_2 \delta'))$, $\forall\ \delta_i$ and $1 \leq i \leq n$. The actual utility of the deal $\delta$ indicates the fitness of the strategy $s_i$.

The negotiation process would stop when timeout occurs or when one of the agents makes a final offer. If the other agent accepts the offer, then a deal is agreed, otherwise there is no agreement.

As a result, the system could support agents to negotiate over multiple issues and to search large and complex spaces. The degree of satisfaction could offer useful information to the agents and help them determine how good offers are. This also provides the agent with an expression of the degree of cooperation in the negotiation by assigning different weightings. Consequently the negotiation process could be improved and speeded up because agents can generate their counter offer by considering the above factor. Agents can carry out negotiation without revealing their utility functions. An agreed solution can be an equilibrium since the search space is explored through the GA. The agent can be sure that no other solution could be better than the current one. The agents use negotiation strategies to specify how the negotiation progresses.

This paper does not consider the demonstration of the effectiveness and efficiency of the proposed system. It only focuses on the relationships between the co-evolutionary mechanism and the degree of satisfaction. This could help examine the functions and roles of these two components in the proposed system. Thus a single business strategic objective is introduced in order to eliminate unintended processes. The negotiation strategy was simplified in order to eliminate its effect on the negotiation results.

The proposed approach is fully implemented as a prototype system for examining its effectiveness in the simulated real world case study.

## 4. Case study

In this section, a case study is used to explain a degree of satisfaction, and its relation to utility function and payoffs in the automated negotiation. The case study will also show the complexity of a utility function and how the utility function is derived in order to provide a rationale for the need of using GAs.

The case study is related to the selling and buying of electronic components and provided by an industrial collaborator in order to test the implemented system in a realistic environment. The company wishes to carry out automated negotiation with its suppliers. The issues over which they are negotiating are Delivery Time (DT), quantities (NO), Products (A), and Prices (P). In this case study, we only consider one type of product denoted as product A. During the negotiation, the company takes the offers made by its opponent to calculate the profits they could make by comparing the offer with the cost involved. If the actual gained profit were larger than the designated one, then the companies would accept the offer. If not, the offers are rejected. The cost function, in this example, consists of labour, production, quality, and material costs, except from the constant overhead cost. We denote the total cost as TC which is computed as, TC = Labour cost (Lc) + Production Cost (Pc) + Quality Cost (Qc)+ Material Cost (Mc) + Overhead (Oc).

The product A has a standard production procedure and Required Production Time (RPT) per unit. So, the Total Production Required Time (TPRT) for manufacturing the quantity (NO) of product A, is derived from NO×RPT. This also gives the basis for calculating required labours. If the TPRT is less than DT, there is no working overtime required. So, the Lc is TRT×cost of per normal hour. However, if the TRT is greater than DT, working overtime is required. So, the DT×cost of per normal working hour+ (TPRT-DT) × cost of per overtime working hour. Mc is a step function that depends on the time, due to the nature of the market. Current price of a component, for example, could be 10 % cheaper than it was a month ago. So, the longer the delivery time is, the cheaper the price it can be. We denoted this material cost function as *F*. Qc is the cost associated with producing quality control charts. One of the most popular quality charts used is $\overline{X}$ charts. The $\overline{X}$ charts are control charts on which single measurements of a quality characteristic are plotted (e.g. current density or temperature). Single measurements are typically used when it is difficult, or uneconomical, to obtain multiple measurements or when multiple measurements are essentially the same as a single measurement (e.g. when a mixture is homogeneous). Usually, sample means (sample averages) are plotted on $\overline{X}$-R charts. The control limits (control range) for the $\overline{X}$ chart require a meaningful estimate of the process standard deviation $\sigma$. This requires that the process variation is exhibited on an $\overline{X}$-$\sigma$ chart. The upper control limit at +3$\sigma$ and the lower control limit at -3$\sigma$ are calculated from the sample averages. We compute it as follows:

$$\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n} \quad \text{and} \quad \sigma = \sqrt{\frac{\sum_{i=1}^{n}\left(X_i - \mu\right)^2}{n}}$$

The upper control limit = $\overline{X}$ + 3$\sigma$

The lower control limit = $\overline{X}$ - 3$\sigma$

*where* $X_i$ *= observed value;* $\overline{X}$ *= sample mean;* $\mu$ *= population value;* $\sigma$ *= process standard deviation and n = the total number of observations in the population.*

The cost of producing $\overline{X}$ charts needs to consider a number of costs incurred at different stages of the whole time cycle and the control policy. The following quality cost function is built upon the down time starting at $t_3$ that can be denoted as follows:

$$Qc = \frac{(C_1 + C_2 * N)}{H} + \frac{(C_5 * \lambda * (t_2 + t_3 + t_4) + \frac{(\alpha * C_3)}{H} + \lambda * C_4)}{(1 + \lambda * (t_2 + t_3 + t_4))}$$

*Qc: total cost of whole time cycle* $C_1$*: fixed cost of operating control chart*
$C_2$*: variable cost of operating control chart* $C_3$*: cost of searching false alarm*
$C_4$*: cost of searching assignable cause* $C_5$*: penalty cost of each defective product*
*T: total cycle time* $T_1$*: average time of process in control*
$T_2$*: average time of process out of control* $T_3$*: average time of down time*
$t_1$*: searching time for false alarm* $t_2$*: search time for assignable cause*

Proceedings of the IEEE International Conference on E-Commerce (CEC'03)
0-7695-1969-5/03 $17.00 © 2003 IEEE

*$t_3$: average total time of searching action* *$t_4$: average total time of revised action*
*H: sampling interval* *N: sample size*
*$\lambda$: process parameter of Exponential distribution* *$\alpha$: the probability of the error type I of control charts*

**Figure 1. Illustration of the quality control policy with a diagram.**

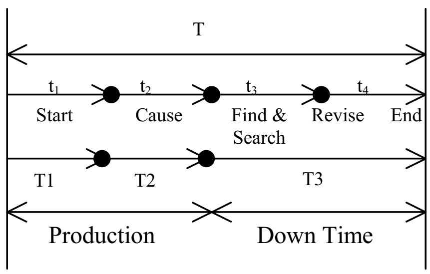


**Figure 2. A policy for generating quality control chart**

The overall cost function can be broken down into the aforementioned variables and functions. In this case, we assume that the production cost and overhead cost are constant variables. So, the values of these variables are kept as constants regardless with other variables. The complexity of utility function that shows the problem domain is a NP hard problem that requires an effective search algorithm to explore the space. The notation of payoff is derived from the calculation of the utility function that is the marginal profit resulted from the current offer against the overall cost. In other words, the payoffs in the offer made from the opponent will be calculated against the expected marginal profits in order to determine whether the offer should be accepted or not. If not, the degree of satisfaction of the agent with this offer will be calculated and passed back to its opponent. Clearly the aim of a degree of satisfaction is to narrow the search space and guide the negotiating agents toward a common solution. The generation of a counter offer is to take its possible maximum gained utility and a degree of satisfaction into account by assigning different weightings whose values are between (0-1) and their total of these two weightings is equal to one. If the agent is inclined to cooperate in the negotiation, the weighting of the degree of satisfaction will be more than the other one. In contrast, the weighting to the degree of satisfaction will be less than the other one, if the agent is unwilling to cooperate. The full cooperative and non-cooperative games are special cases of the model. Although some parameters in the cost function were simplified in order to examine the effectiveness and efficiency of the proposed system, but the complexity of the problem domain still remains.

## 5. Experiment Results

The cost function mentioned in the previous section was implemented as a function in the utility function. So the utility is calculated by adding the degree of satisfaction and the cost function with different weightings. We assume that two agents have the similar cost functions with different concerns. Both agents have different utility functions. During the experiments, the agent did not take into account its opponent's utility function when generating counter offers. They use the degree of satisfaction. Even though the agents only negotiate over 4 issues, the number of issues one agent needs to calculate is 15. So, in the GA a chromosome is represented by an array that contains 15 variables. The population size of the possible offers for agents is 50 at one time. The mutation rate is 0.1 with 1 mutation and the crossover rate is 0.1 with 1 crossover. The tournament approach is adopted for the selection strategy. The two fittest chromosomes are selected as parents in order to generate two offspring. The experiments start with 400 generations that correspond to offers and counter offers, in order to see whether the search space was explored or not. After a number of experiments having been carried out, the number of generations can be reduced from 400 to 200, as it is sufficient to explore the space. Since the search space is large and complex, the global optimised solution cannot be known, so the system did not stop at the agreed point in order to examine the quality of the solution.

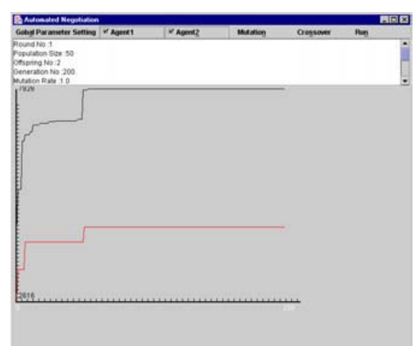

**Figure 3. A full non-cooperative game**


0-7695-1969-5/03 $17.00 © 2003 IEEE

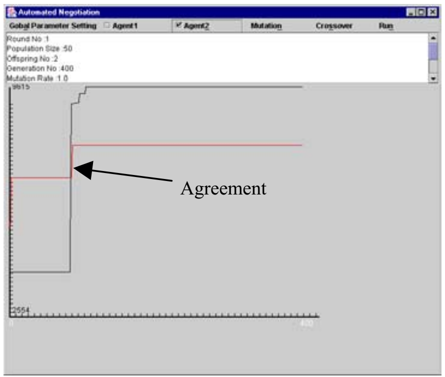


**Figure 4. Both agents have 60% cooperation level**

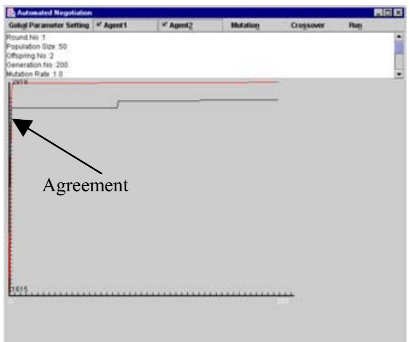


**Figure 5. Both agents have 80% cooperation level**

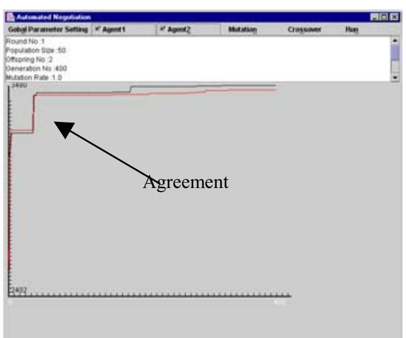


**Figure 6 Both agents have 55% cooperation level**

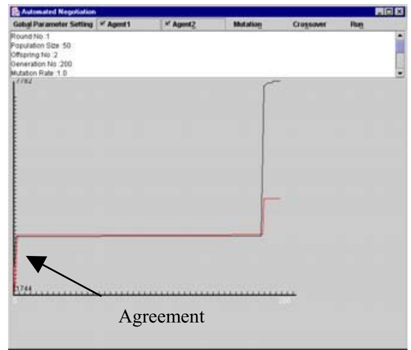


**Figure 7 Both agents have 50% cooperation level**

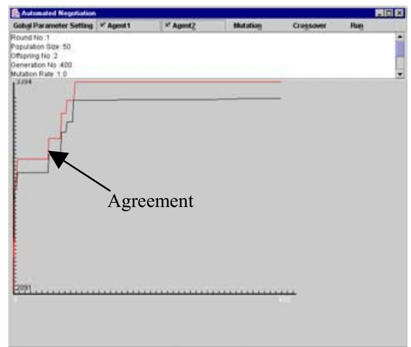


**Figure 8. Both agents have 40% cooperation level**

After a number of experiments have been carried out, some representative results are only exhibited in the above figures. Although the environment settings in two separated experiments were the same, the results may not be the same. This results from that various random seeds in the GA are generated. In Figure 3 the system shows both agents are unwilling to cooperate. Since there is no time constraint on them, there is no agreement, even though there is a possible concession in the search space. Both agents did not consider the degree of satisfaction. In some non-cooperative game experiments, the time constraints are included in the negotiation process. The agents could reach agreements, but the results show low payoffs for both agents. Figure 4, 6, and 7 show the gained utilities in agreed solution is close to the maximum gain for the both sides when both agents have between 50%-60% cooperation levels. In addition, in most cases, they came to concessions very quickly. When both agents are prepared to fully cooperate in the negotiation (Figure 5), the result is not better off than the settings in the Figure 4, 6, and 7. Figure 8 shows that the agents are more selfish than the previous cases apart from figure 3, but there is a concession. However, they could have found a better solution that both agents can have better utilities, if they did not made a concession so early.

## 6. Conclusion and future work

In this work, we have proposed a novel automated negotiation mechanism that includes a co-evolutionary mechanism to search complex and large spaces and a degree of satisfaction to examine the negotiation process. The main contribution of this work is the introduction of a degree of satisfaction for agents to carry out co-operative, and non-cooperative negotiation without revealing their utility functions. The proposed negotiating approach allows the negotiating agents to express different levels of cooperation in the negotiation and the degree of satisfaction to provide essential feedback to the offering agent without revealing the utility function. The co-evolutionary approaches [13,14,15] were only designed for non-cooperative games. The proposed approach allows the agents to conduct different co-operative and non-cooperative games.

The proposed approach was implemented in Java. Experimental results show the effectiveness of applying the proposed system to the realistic scenario. However, in this paper, we only use a simplified realistic case study to evaluate the proposed solution, due to resource constraints. However, the nature of the problem domain is sufficiently complex to demonstrate the function of the proposed system. Parts of the proposed architecture such as business strategic objectives and their corresponding utility functions, negotiation strategies were not included in this exercise so as to demonstrate the effectiveness and efficiency of co-evolution and degree of satisfaction. As a result, we only employ one strategic objective and one utility function in the experiment in order to eliminate secondary factors. In order to ensure that there is at least one possible agreement in the negotiation and that there is no non-rational decision, the possible offers and counter offers were carefully designed. They were generated by another standalone GA before the experiments were carried out, in order to avoid a premature acceptance from occurring during the negotiation. This will prevent agents from accepting previously rejected offer. Therefore, the maximum expected gained utility for the agents was pre-defined.

In the next stage of this work, we will design a method that will allow the maximum expected gained utility to be evolved in the negotiation in order to reflect realistic requirements. Another improvement can be made in this research by assigning different weightings to the offers made by the opponent. So, the most recent offers should have more weight than the offers. This could assist agents in identifying the trend of the preferences of its opponent in order to speed up the negotiation process.

## References


[1] Chao, K-M, Anane, R., Chen, J-H, and Gatward, R., Negotiating Agents in a Market-Oriented Grid, 2nd *IEEE/ACM International Symposium on Cluster Computing and the Grid*, IEEE Computer Society, 2002, pp 436-437.

[2] Chen, J-H, R. Anane, Chao, K-M, & N. Godwin, Architecture of an agent-based negotiation mechanism, *Proceedings of The 22nd IEEE International Conference on Distributed Computing Systems Workshops*, IEEE Computer Society, 2002, pp379-384.

[3] Chen, J.H., Chao, K-M, Godwin, N., Reeves. C., and Smith, P., 2001, An automated negotiation mechanism based on co-Evolution and game theory, *the 17th ACM Symposium on applied Computing, Special Track on Agents Interactions, Mobility, and System (AIMS)*, 2002, pp63-65.

[4] Zlotkin, G., and Rosenschein, j., Mechanisms for automated negotiation in state oriented domains. *Journal of Artifical Intelligence Research*. Vol.5, 1996, pp. 163-238.

[5] Kraus, S., Negotiation and Cooperation in Multi-Agent Environments, *Artificial Intelligence journal, Special Issue on Economic Principles of Multi-Agent Systems*, Vol.94, No.1-2, 1997, pp79-98.

[6] Jennings, N.R., Parsons. S., Noriega. P., and Sierra. C., On Argumentation-Based Negotiation, *Proceedings of the IWMAS*, MIT Endicott House, Dedham, Massachusetts, USA, October 12-15, 1998, pp. 1-7.

[7] Jennings, N. R., Faratin, P., Lomuscio, A. R Parsons, S., Sierra, C. & Wooldridge, M., Automated negotiation: prospects, methods and challenges, *International Journal of Group Decision and Negotiation*, Vol. 10, No. 2, 2001. pp 199-215.

[8] Muthoo, A., *Bargaining and Applications*. Cambridge University Press, 1999

[9] Kraus, S., K. Sycara, K. & Evenchik A., Reaching agreements through argumentation: a logical model and implementation, *Artificial Intelligence*, Vol. 104, 1998, pp. 1–69

[10] Parsons, S., Sierra, C. and Jennings, N. R., Agents that reason and negotiate by arguing, *Journal of Logic and Computation*, Vol. 8, No 3, 1998, pp 261-292.

[11] Binmore, K., *Fun and Games: A Text on Game Theory*, D.C. Heath and Co. Press, 1992.

[12] Rosenschein, J.S. & Zlotkin, G., *Rules of Encounter*, MIT Press, 1994.




0-7695-1969-5/03 $17.00 © 2003 IEEE


IEEE COMPUTER SOCIETY

[13] Oliver, J. R., *On Automated Negotiation and Electronic Commerce*, PhD thesis, University of Pennsylvania, 1997.

[14] Matos, N., Sierra, C. & Jennings, N. R., Determining successful negotiation strategies: an evolutionary approach *Proc. 3rd Int. Conf. on Multi-Agent Systems*, Paris, France, 1998, pp 182-189.

[15] Peyman, F., *Automated Service Negotiation Between Autonomous Computational Agents*, PhD thesis, University of London, 2000.

[16] Soo, V. W., Agent negotiation in trusted third party mediated uncertain games, *Conference Proceeding of the Fourth International Conference on Autonomous Agents*, ACM Press, 2000, pp 265 – 266.

[17] Laing, C., Chao, K-M, Robinson, A. & Sun, P-C, Monitoring the progression of non-traditional student expectations and progression: a game theory approach, *Conf. Proc. of IEEE 2002 Frontiers in Education*, IEEE Press, 2002.

[18] Chao K-M, Anane, R., Plumley, J., Godwin, N., & Naguib, R.N.G., A mobile agent framework for telecardiology, 23rd *IEEE Conf. Proc. of EMBC01*, IEEE Press, 2001.



IEEE COMPUTER SOCIETY