This article can be cited as K. Panagidi, K. Kolomvatsos and S. Hadjiefthymiades, An Intelligent Scheme for Concurrent Multi-Issue Negotiations, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 129-149, 2014.

An Intelligent Scheme for Concurrent Multi-Issue Negotiations

Kakia Panagidi, Kostas Kolomvatsos and **Stathes Hadjiefthymiades**

Pervasive Computing Research Group (p-comp), Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, 15784, Athens, Greece
e-mails: {k.panagidh, kostasks, shadj}@di.uoa.gr

ABSTRACT

Automated negotiations are an active research field for many years. In negotiations, participants' characteristics play a crucial role to the final result. The most important characteristics are the deadline and the strategy of the entities. The deadline defines the time for which each entity will participate in the negotiation while the strategy defines the proposed prices at every round. In this paper, we focus on the buyer side and study multi-issue concurrent negotiations between a buyer and a set of sellers. In this setting, the buyer adopts a number of threads. We propose the use of known optimization techniques for updating the buyer behavior as well as a methodology based on the known Particle Swarm Optimization (PSO) algorithm for threads coordination. The PSO algorithm is used to lead the buyer to the optimal solution (best deal) through threads team work. Hence, we are able to provide an efficient mechanism for decision making in the buyer's side. In real situations, there is absolutely no knowledge on the characteristics of the involved entities. We combine the proposed methods adopting the Kernel Density Estimator (KDE) and Fuzzy Logic (FL) in order to handle incomplete knowledge on entities characteristics. When an agreement is true in the set of threads, KDE is responsible to provide to the rest of them the opportunity to calculate the probability of having a better agreement or not. The result of the KDE is fed to a FL controller in order to adapt the behavior of each thread. Our experiments depict the efficiency of the proposed techniques through numerical results derived for known evaluation parameters.

Keywords: Multi-Issue Concurrent Negotiations, PSO, Optimization, Fuzzy Logic, KDE

Computing Classification System: I.2.1 [Artificial Intelligence]: Applications and Experts Systems

1. INTRODUCTION

In E-Commerce applications one can find virtual places, called Electronic Marketplaces (EMs), where users can exchange products for specific returns. Users, usually, are involved in interactions for concluding the discussed transactions. The interaction between entities is called negotiation. Negotiation is the process where unknown entities try to agree upon the exchange of specific items for specific returns (Raiffa, 1982). *Intelligent Agents (IAs)* can undertake the responsibility of representing users in EMs. We can identify three types of users: the buyers, the sellers, and the middle entities. Such intelligent autonomous components can take specific user preferences and return the desired product (for buyers) or the return (for sellers). In general, entities are selfish and try to maximize their utility. Buyers want to buy products at the lowest possible price while sellers want to sell products at the highest possible price.

The negotiation could involve a single issue of the product (e.g., the price) or multiple issues (e.g., price, delivery time, quality). In the first case, the utility is based only on the single issue while in the second case, the utility is depended on the entire set of issues. For instance, the utility could be defined as the weighted sum of the values received for each issue. One can identify bilateral (one-to-one) or one-to-many negotiations. In bilateral negotiations, one buyer negotiates with exactly one seller. In one-to-many negotiations, a buyer can negotiate, in parallel, with a number of sellers. In EMs, a product can be provided by a number of sellers, however, with different characteristics (e.g., price, delivery time). Every buyer should decide from which seller to buy the desired product. This can be achieved through the adoption of concurrent negotiations. The buyer utilizes a number of threads in order to negotiate with every seller. However, in such cases, the buyer needs a coordination module that could probably define the strategy for each thread. Such strategy is compatible with the results obtained by applying specific strategies in the remaining threads.

In this paper, we focus on multi-issue concurrent negotiations. We propose specific methodologies to enhance the intelligence of the discussed IAs. Our aim is to develop a decision making mechanism which could be applied in real negotiations. We assume absolutely no knowledge on the players' characteristics. Such knowledge involves the players' deadlines, reservation values, etc. Reservation value is the acceptable upper / lower limit of price for the buyer / seller. We propose specific algorithms that can dynamically change the buyer's strategy without the need of a coordinator. In our scenario, there is no need for coordination as threads adopt *Swarm Intelligence (SI)* (Engelbrecht, 2007) techniques in order to converge to the best possible agreement. When an agreement is achieved by a specific thread, the remaining threads can readjust their strategy, if needed, in order to force the respective sellers to accept lower prices. Each thread provides feedback to the seller for the final agreement after a specific time interval that enables the remaining threads to achieve lower prices. Additionally, we propose a mechanism, based on *Fuzzy Logic (FL)* and the *Kernel Density Estimator (KDE)*, responsible to update the deadline of each thread. The rationale is that based on the seller's offers, the buyer could have an insight on the probability of having better agreement than the already observed agreement in another thread. Each thread applies the proposed mechanism and decides if it remains more or less to the negotiation by increasing or decreasing its deadline. We assume that the buyer permits a small extension of the deadline for each thread (in the case where the deadline should be extended).

The rest of the paper is organized as follows: Section 2 discusses the related work to the aforementioned problem while Section 3 presents the examined scenario. We present the entities behavior and analyze our methodologies for changing utility function weights. We also describe the *Particle Swarm Optimization (PSO)* approach for reaching to the optimal deal. In Section 4, we discuss the use of the KDE and FL in our setting. We describe the proposed FL controller and present the fuzzy rule base. Section 5 elaborates on our results through an analysis of our experiments. The results reveal the efficiency of the proposed models. Finally, in Section 6, we conclude the paper by giving future work directions.

This article can be cited as K. Panagidi, K. Kolomvatsos and S. Hadjiefthymiades, An Intelligent Scheme for Concurrent Multi-Issue Negotiations, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 129-149, 2014.

2. RELATED WORK

In literature, one can find interesting research efforts involving either finite or infinite horizon negotiations. The horizon (deadline) defines the time for which the involved parties will participate in the interaction. At every round, the entities exchange specific offers in an alternative way. The aim is to have an agreement and, finally, profits for all. Many solutions have been provided so far. These solutions could be categorized in:

i) Game Theoretic (GT) approaches,

ii) Machine Learning (ML) approaches,

iii) Fuzzy Logic (FL) based models and,

iv) Models based on heuristic decision functions.

In the discussed solutions, researchers have proposed models for handling bilateral or one-to-many negotiations (Faratin et al., 1998; Fatima et al., 2005; An et al., 2006; Chen et al., 2009; Da Jun & Xian, 2002; Sun et al., 2007; Robu et al., 2005; Lau, 2005). The authors define the strategies of the entities as well as the interaction protocol.

In (Fundeberg et al., 1987), the authors describe a negotiation model with outside options in the seller side. The seller faces an infinite number of buyers in the negotiation process. The buyers' valuation (i.e., the upper price limit – reservation value) is common knowledge. An important characteristic of the presented model is that buyers do not have the opportunity to make any offers to the seller. A negotiation between a seller and a buyer is also analyzed in (Chaterjee & Samuelson, 1988). The authors describe two players' strategies: hard and soft. Prior probabilities for reservation values are common knowledge. The context of an infinite horizon negotiation under two sided uncertainty is examined in (Crampton, 1984). The authors describe how time and information affect the rational behavior of IAs when commitment is not possible. The players should exchange some private information before an agreement is reached.

A Pareto optimal algorithm is described in (Jazayeriy et al., 2011). The authors describe the Maximum Greedy Trade-offs algorithm that generates offers. IAs have limited knowledge about the opponents' deadline and utility. The generation of Pareto-optimal offer needs information about the opponent's importance weights. In (Faratin et al., 1998), the authors describe functions for the definition of alternating offers and present a set of tactics based on which IAs try to conclude an agreement. They propose a formal model for negotiations and provide relevant results. They examine a large set of tactics and define the following metrics: i) the intrinsic benefit of the agent, ii) the cost, and, iii) the performance of the intrinsic utility relative to a complete knowledge interaction. The authors in (Fatima et al., 2005) present a negotiation between buyers and sellers, and describe a set of strategies for both sides. The examined parameters are the player's deadline, the acceptable prices and the discount factor. The optimal strategy of the player is studied w.r.t. a product pricing scheme. Three types of functions are defined: linear (over time), boulware (the player reaches its final proposed price slowly) and conceder (the player reaches its final price quickly). The authors in (Robu et al., 2005) propose a method for complex bilateral negotiations over multiple issues. The negotiation can be further improved by incorporating a heuristic proposed in (Jonker et al., 2004). Based on the proposed model, an IA utilizes the history of the opponent's bids to predict her preferences. Lau (2005) proposes a multi-agent multi-issue mechanism. A decision making model is described based on a genetic algorithm. In (Wu et al., 2009), the authors present an automated multi-agent multi-issue negotiation. In their model three agents bid sequentially in consecutive rounds. In (Türkay & Koray, 2012), the authors present a multi-issue negotiation mechanism which adapts a modified Even-Swaps method. A FL inference system is responsible for negotiating on several issues simultaneously.

ML models are used for predicting the opponent's characteristics (Rahwan et al., 2002). Genetic algorithms have been proposed to provide an efficient methodology for the prediction and learning of the opponent strategy (Gerding & van Bragt, 2003; Oprea, 2002). Additionally, in (Bui et al., 1995; Rahwan et al., 2002), the authors focus on the use of Bayesian models for learning the opponent's behavior. Finally, in (Sun et al., 2007), the authors propose the use of reinforcement learning techniques in order to be able to decide if their offers will be accepted by the opponent. The above mentioned solutions have specific drawbacks when used in negotiations. Learning mechanisms aim to discover the optimal strategy as the response to the opponent move and not to provide an efficient generic decision making mechanism. ML models require increased computational effort and the use of training samples while Bayesian learning techniques require the knowledge of the a priori probability on the opponent type.

FL is the right tool for handling uncertainty that is inherent in dynamic environments. In (Cheng et al., 2005), the negotiation tactics are presented as fuzzy rules and a simple heuristic is employed to learn the preferences of the opponent. In (Kolomvatsos & Handiefthymiades, 2008), an interaction model involving alternating offers between buyers and sellers is studied. The authors describe a mathematical model for the seller's deadline calculation and present simulation results. In (Kolomvatsos et al., 2008), a FL model for the deadline calculation is proposed. A set of fuzzy rules are defined according to experts' knowledge. The authors in (Luo et al., 2003) present a fuzzy constraint based solution. IAs seek to determine a fair solution for all parties and, thus, several options that satisfy them are determined. In (Raeesy et al., 2007), a fuzzy-model for IAs negotiations is presented. The authors investigate negotiations dealing with offers based on fuzzy values on both sides. FL approaches require the definition of a rule base that covers all the aspects of a negotiation. However, this could be a very difficult task for real negotiations.

Optimization problems are the subject of many research papers found in literature. The aim of the proposed models is to provide efficient techniques for solving computational problems under a number of constraints. Some recent efforts are as follows. A support vector regression (SVR) model in combination with chaotic genetic algorithm (CGA) is discussed in (Hong et al., 2011). SVR aims to minimize an upper bound of the generalization error rather than minimizing the training error. The CGA based on the chaos optimization algorithm and genetic algorithms (GAs) is used to overcome the premature local optimum in determining parameters of SVR. In (Precup et al., 2012), the authors propose an adaptive gravitational search algorithm (GSA) for the optimal tuning of fuzzy controlled servo systems. The system results a Takagi – Sugeno proportional integral fuzzy controller. An extension of the known cultural algorithms is presented in (Ali et al., 2013). The authors discuss the combination of cultural algorithms with information retrieved from social networks to resolve complex mechanical design optimization problems. The proposed algorithm is compared with other known algorithms in the field. Results reveal that the information retrieved from social networks can help in the adaptation process during the course of searching the problem landscape. Yazdani et al. (2013) try to cover weak points of the known artificial fish swarm algorithm (AFSA) including the lack of previous experiences in the optimization process as well as the lack of existing balance between exploration and exploitation. Moreover, a hybrid clustering algorithm is proposed based on NAFSA (New AFSA) and K-means. The combination of the two discussed methodologies leads to maximum utilization for data clustering.

Concurrent negotiations provide many advantages as a buyer can negotiate, in parallel, with a number of sellers. This gives her the opportunity to select the best possible agreement (if there is any). Researchers, identifying the advantages of this setting, have proposed a number of models. In (Zeng & Sycara, 1998), the authors discuss one-to-many negotiation between a buyer and multiple sellers. The buyer is forced to wait until the reception of offers from all the threads before generating the next offer. Furthermore, every seller can easily join and leave the negotiation. In (Rahwan et al., 2002), the authors propose three coordination strategies for one-to-many negotiation. The proposed strategies are: Desperate, Patient, and Optimized Patient. Adaptive mechanisms have been proposed to increase the efficiency of the models (Oprea, 2002; Narayanan & Jennings, 2003). In (Oprea, 2002), an adaptive model adopting an artificial neural network is discussed. The network is used for learning purposes. The authors in (Nguyen & Jennings, 2003) extend the work presented in (Rahwan et al., 2002) and propose strategies for coordinating concurrent negotiation threads. The coordinator is the most important part of the proposed architecture. It is responsible to choose the appropriate strategy applied by each thread. It receives the status of each thread and decides the strategy based on the parameters of each interaction. In (An et al., 2006) two strategies are described: Fixed-Waiting-Time-Based and Fixed-Waiting-Ratio-Based. Negotiation is conducted in continuous time and IAs rely on the discussed strategies to issue a proposal. Experimental results suggest that the proposed mechanism achieves more favorable outcomes than the general, one-to-many methodologies. The proposed approach is based on the combination of a number of ad hoc heuristics containing a large number of parameters. Finally, in (Williams et al., 2012), the authors propose another coordination strategy for multiple threads. In this work, the decisions are based on the information about opponents. This information changes after the reception of an offer. The coordinator decides the best time to stop the negotiation and calculates the utility at that time. Useful conclusions on the future extensions in the field are presented in (Baarslag et al., 2013). The paper discusses an in-depth analysis and the key insights gained from the 2nd International Automated Negotiating Agents Competition (ANAC, 2011). The authors analyze the strategies used in negotiations and techniques adopted by the teams. In particular, they show that the most adaptive strategies are not necessarily the ones that win the competition.

In our model, we do not need any coordinator to specify the strategy for each thread. Threads by following the PSO algorithm try, through a team effort, to find the optimal solution (best deal). The

This article can be cited as K. Panagidi, K. Kolomvatsos and S. Hadjiefthymiades, An Intelligent Scheme for Concurrent Multi-Issue Negotiations, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 129-149, 2014.

impact of our work is that the proposed scheme is based on a self-organization technique adopted by threads. Hence, the buyer saves resources. Moreover, there is not any need for exchanging a large number of messages as in the coordinator case. The decision making of each thread is made independently and is automatically adapted to each seller strategy. Finally, we enhance the described mechanism with applying an estimation technique on the seller strategy and a FL controller responsible to update the deadline of each thread (if needed). The FL controller, based on the negotiation information, decides changes in the deadline in order to make the buyer aligned with the needs of the negotiation and possible agreements held in other threads.

3. MULTI - ISSUE CONCURRENT NEGOTIATIONS

3.1. Scenario description

In Figure 1, we can see the architecture of our model. At every round, each thread sends / receives an offer (a bundle of values for the examined issues). If an agreement is true, in a specific thread, then the agreement message is sent to the rest of them (the message is received by every thread that currently participates in an active negotiation). Based on the PSO algorithm, the remaining threads changes their strategy in order to pursue a better agreement than the previous one. Moreover, the remaining threads change the weights for the utility calculation in order to pay more attention on specific issues and, thus, to achieve a better utility in a possible future agreement.

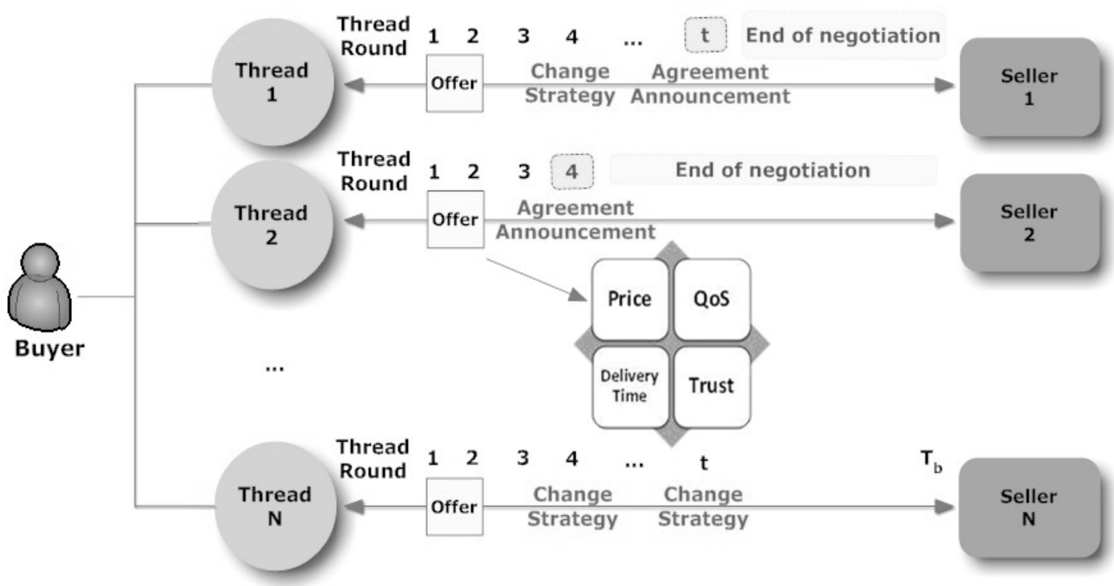

Figure 1. Concurrent negotiations scenario.

The IAs, representing each party, have no knowledge about the preferences of their opponents (limited knowledge). There is no need for a central decision maker which collects the information of

IAs and transmits directions to them. Every seller has the same product in her property retrieved by a specific cost and tries to sell it in the highest possible profit. Similarly, the buyer is interested in purchasing the product that is close to her preferences. The product has a number of characteristics (issues) that affect the final utility. These issues are categorized as *proportional (P)* or *inversely proportional (IP)* to the utility. For proportional issues, the greater the value is, the greater the utility becomes. The opposite stands for the inversely proportional issues.

The buyer has a specific deadline defined by her owner. The same stands for the seller. Let us denote the buyer deadline with T_b while the seller deadline is depicted by T_s. In each negotiation, the seller starts first and the buyer follows, if the proposed offer is rejected. If a player is not satisfied by the proposed offer, she has the right to reject it and issue a counter-proposal. Every offer involves specific values for the examined issues. This approach is defined as the package deal (Torroni & Toni, 2001; Rahwan et al., 2002). If a deadline expires and no agreement is present, then the negotiation ends with zero profit for both. Both entities utilize a specific utility function (U) defined as follows:

$$U = \sum_{i=1}^{m} w_i \cdot v_i \,,$$

where m is the number of issues, w_i and v_i are weights and values for each issue. Additionally, both players have their own strategy for offers calculation. We adopt the approach described in (Fatima et al., 2002; Oprea, 2002). Each entity has her own reservation values for every issue. We consider the interval [$\min_i$, $\max_i$] where each issue i takes values. These values differ on the buyer as well as on the seller side. Both entities generate their offers based on the following equations:

$$O_i = \min_i + \varphi(t) \cdot (\max_i - \min_i) \,,$$

$$O_i = \min_i + (1 - \varphi(t)) \cdot (\max_i - \min_i) \,,$$

for the buyer and the seller side, respectively. In the above equations, O_i depicts the next offer for issue i. As we can see, our model involves a time dependent strategy that is represented by the function *φ(t)*. More details for the φ function can be found in (Fatima et al., 2005). Finally, for every issue, we calculate the corresponding utility based on the following equations:

$$U(v_i) = \frac{v_i - \min_i}{\max_i - \min_i}, \text{if issue i is P} \quad \text{or} \quad U(v_i) = \frac{\max_i - v_i}{\max_i - \min_i}, \text{if issue i is IP} \,.$$

3.2. Weights Adaptation

Our approach to optimize buyer's utility is to modify the weights for each issue. When a thread closes a deal, then a message is broadcasted to the rest threads and they reassess, if needed, the weights of the utility function. A change in the weights will result to a change in the thread behavior and, thus, in the utility function. The aim is to have a tradeoff between issues in order to reach to a better agreement. We emphasize on those issues having value worse than the agreement value. The reason is that the utility could increase depending on such issues (the utility will be increased, if we increase the weight of the specific issue and achieve a better value in the negotiation). This way, the thread tries to force the seller to give better offers. The utility will be increased, if and only if the seller will improve those values. We apply two methodologies for calculating the weights of the utility function: a methodology based on the *Simplex optimization* (Kennedy & Eberhart, 1995) and a methodology based on the *Analytic Hierarchy Process (AHP)* (Dagdeviren & Yuksel, 2008).

The Simplex method is a process for solving linear programming problems and aims to the optimal solution based on a set of conditions. It is a very efficient method used for solving demanding problems. The first step, in Simplex method, is to find a basic feasible solution. After that, the solution is tested for optimization in terms of the objective function and the effect of the input. The input is a non-basic variable used to replace at least one of the key variables that we already have defined as a solution. If there is an improvement by using the specific input, this gives a new feasible solution. It should be noted that another important feature of this method is that for every new solution the value of the objective function is at least as optimal as in the previous solution. This results, in each step of the iterative process, a movement closer to the optimal solution. Finally, the algorithm defines specific conditions that determine when the optimal solution is met.

In our scenario, the optimal equation is the following:

$$\text{Maximize } U$$

$$\text{subject to: } \sum_{j=1}^{n} w_j = 1,$$

$$w_j \geq 0,$$

$$U \geq U_A,$$

where U_A is the utility retrieved by the agreement (in another thread). The discussed calculation is held every time an agreement is announced and the result is the appropriate values of weights that solve the optimization problem. The definition of weights will lead us to a better position for the buyer (in utility terms). If the problem does not have a feasible solution, then the weights remain unchanged.

The AHP has been developed by Saaty and it is one of the best known methodologies in multi-criteria problems. It allows users to assess the relative weight of multiple criteria or multiple options against given criteria in an intuitive manner. In case that quantitative ratings are not available, policy makers or assessors can still recognize whether one criterion is more important than another. The basic process to carry out the AHP consists of the following steps:

a) structuring a decision problem and selection of criteria,

b) priority setting of the criteria by pair wise comparison,

c) pair wise comparison of options on each criterion,

d) obtaining an overall relative score for each option.

The process of using the AHP in our setting is as follows. Every weight is initialized to a very small number close to zero. After that, the distance between each current thread issue and values in the agreement is calculated and is normalized in [0, 1]. The greater the distance from the agreement's offer is, the greater the weight will be. A value close to 1 expresses the importance of the specific issue, in the reassessing process. If an issue has better value than the corresponding in the agreement, then the weight remains unchanged (close to zero). As a next step, we utilize the approach of redefining the weight, which is proposed in (Kong & Liu, 2005). A new fuzzy comparison matrix differs from Saaty's model in that we use membership scales, instead of the 1-9 scales, as the values of the elements. This method compares weights in pairs and is more straightforward and easier to be used. Table 1 presents our membership scales expressed in the same way as Saaty's scale.

Table 1: Scale for fuzzy pair-wise comparison.

Saaty's Scale	Scale Values	Relative Importance of Elements
1	[0.5, 0.6]	Equally important
3	[0.6, 0.7]	Moderately important with one over another
5	[0.7, 0.8]	Strongly important
7	[0.8, 0.9]	Very strongly important
9	[0.9, 1.0]	Extremely important
2, 4, 6 ,8	[0,0.5)	Intermediate Values

3.3. Autonomic behavior adopting PSO

We also propose a model where buyer threads are automatically organized to change their strategy in order to reach the best deal. Our proposed technique is based on the known PSO algorithm (Kennedy & Eberhart, 1995). PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with respect to a given measure of quality. In such systems, each individual has a very simple behavior: to follow the success of its own and its neighbors. The collective behavior that is the result of such simple behavior finally leads to discover the optimal regions of a search space. The position of each individual is adjusted according to its own experience and the experience of its neighbors. In our case, every thread is a particle that can move in the m - dimensional space. The m dimensions of space are assigned to the m issues. Threads should converge to the optimal solution which is the best deal for the specific group of sellers. Hence, every thread modifies her position according to the current velocity, current position, her distance between current position and global best position and her distance between current position and local best position. Threads position y_i^t at time t is the current offer (package) in a specific round while the personal local best position is the best offer till the specific negotiation round. Finally, the global best position is the agreement with the maximum utility so far in the entire set of threads. Combining PSO

This article can be cited as K. Panagidi, K. Kolomvatsos and S. Hadjiefthymiades, An Intelligent Scheme for Concurrent Multi-Issue Negotiations, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 129-149, 2014.

algorithm with the *Virtual Force Algorithm (VFA)* (Howard et al., 2002), we assume that every issue can create a force F_i on particle (thread) i. The combination of particle's global best position and particle's local best position will move the particle to her next position y_i^{t+1} at time t + 1.

4. APPLYING FUZZY LOGIC IN THE BUYER BEHAVIOR

In this section, we present our proposed seller's strategy estimation methodology as well as a FL controller responsible to update the behavior of the buyer. The aim is to handle the uncertainty that every thread has on the seller behavior.

4.1. Seller strategy estimation

We propose an estimation scheme on the seller strategy. Our aim is to provide an intelligent mechanism in order to update the threads deadline during negotiation. We consider that the buyer permits a small extension on the deadline of each thread. Threads aim to save time and resources when they are aware that the seller will not make an offer better than an agreement reached to another thread. If the thread sees that there is increased probability of having a better agreement, the deadline could be extended for little more. The estimation scheme is related to dynamic issues meaning that their values change at every round of the negotiation (e.g., the trust value cannot be changed during negotiation). For simplicity in our description, let us focus on price. The rationale for the rest issues (e.g., delivery time) is the same.

For the seller's pricing strategy estimation, we adopt the *Kernel Density Estimation (KDE)* (Wand & Jones, 1995). KDE is used for estimating the *probability density function (pdf)* of an unknown distribution. The discussed technique is widely used in ML, data mining, etc. Let (x_1, x_2, x_3, …, x_N) be a sample drawn from the unknown distribution. In our scenario, x_k values represent seller offers (price). Each thread maintains the list of the seller's offers in order to be able to predict the seller strategy distribution based on the KDE. The Kernel estimator of this distribution is defined by the following equation:

$$\hat{f}_h(x) = \frac{1}{N \cdot h} \sum_{k=1}^{N} K\left(\frac{x - x_k}{h}\right),$$

where *x* is the discussed variable, *N* is the sample size, *h* is the bandwidth of the kernel and *K(·)* is the Kernel function. The Kernel function is a symmetric function integrating to 1. As Kernel function, we adopt the Gaussian and, thus, the pdf of the seller pricing strategy is given by:

$$P(x) = \frac{1}{N \cdot h} \cdot \sum_{k=1}^{N} K\left(\frac{x - p_{sk}}{h}\right),$$

and, finally,

$$P(x) = \frac{1}{N \cdot h} \cdot \sum_{k=1}^{N} \frac{1}{\sqrt{2 \cdot \pi}} \cdot e^{-\frac{(x - p_{sk})^2}{2 \cdot h^2}} .$$

This article can be cited as K. Panagidi, K. Kolomvatsos and S. Hadjiefthymiades, An Intelligent Scheme for Concurrent Multi-Issue Negotiations, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 129-149, 2014.

In our scenario, without loss of generality, we take the bandwidth equal to 1. Hence, we obtain the following equation:

$$P(x) = \frac{1}{N} \cdot \sum_{k=1}^{N} \frac{1}{\sqrt{2 \cdot \pi}} \cdot e^{-\frac{(x-p_{sk})^2}{2}}.$$

In the above equations, p_{sk} depicts the seller offer proposed at the k^{th} negotiation round. Based on the above analysis, we take the *cumulative distribution function (cdf)* of the seller strategy:

$$CDF(x) = \frac{1}{N} \cdot \sum_{k=1}^{N} \frac{1}{2} \cdot \left(1 + erf\left(\frac{x - p_{sk}}{\sqrt{2}}\right) \right),$$

where *erf(·)* is the error function of the Gaussian. An approximation of the error function is given by:

$$erf(x) \approx sgn(x) \cdot \sqrt{1 - e^{-x^2 \cdot \frac{\frac{4}{\pi} + a \cdot x^2}{1 + a \cdot x^2}}},$$

where parameter *a* could be equal to 0.14 (Satyala & Pieper 2008) and sgn(x) is positive in our case.

With the use of cdf, we are able to calculate the probability of having the seller offers below a specific threshold. For instance, if we focus on price, we can have the probability of having the seller's offers below the price observed in the agreement reached to another thread. The same rationale stands for the remaining issues (e.g., delivery time). The discussed probabilities are fed to a FL controller in order to result the adaptation of the thread's deadline. The FL controller is described in the upcoming section.

4.2. The FL controller

FL is appropriate for real-time decision-making allowing a degree of uncertainty at the decision making phase. Allowing a degree of fuzziness at decision making makes players more flexible and capable of handling the opponent's offers. Based on the KDE, each thread can estimate two probabilities (one for each issue changing at every negotiation round – in this paper, we consider price and delivery time):

i) the probability of getting a price below the agreement price (p_{pr}) and,

ii) the probability of getting a delivery time better than the delivery time observed in the agreement (p_{dt}).

We exploit FL in order to apply a fuzzy rule-based system capable of adapting the behavior of the thread i.e., the value of T_b. The rationale is that if there is possibility to achieve better values compared to the values indicated by the already observed agreement, the thread could stay a little more in the negotiation. If such case is not possible, the thread should decide to reduce the deadline and, thus, to perform more pressure on the seller. The decision is made by the proposed FL controller.

This article can be cited as K. Panagidi, K. Kolomvatsos and S. Hadjiefthymiades, An Intelligent Scheme for Concurrent Multi-Issue Negotiations, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 129-149, 2014.

The FL controller has two inputs and one output. Inputs are related to the above described probabilities (p_{pr}, p_{dt}) and the output is related to the update of the T_b (*DP*) which is depicted as a percentage. DP takes values in the interval [-1, 1] which means that T_b could be increased or decreased. Let us give a specific example. The developer is capable of defining a maximum value that will affect T_b. If this value is equal to 10, it means that T_b could be increased or decreased by 10 rounds (maximum). If the FL controller results a value equal to 0.3, it means that T_b will be increased by 30%·10 = 3 rounds while a value equal to -0.1 means that T_b will be decreased by 10%·10 = 1 round. The discussed maximum value could be changed as the developer desires. Furthermore, the proposed system could be easily extended by taking into consideration other important parameters of the negotiation.

For the reasoning process, an input to the fuzzy system might be described as: the probability value is a small number close to zero represented by the value low (*$l(u_i)$ = low*), medium number (around 0.5) (*$l(u_i)$ = medium*) or large number close to 1 (*$l(u_i)$ = high*) where *u_i = 'the probability of having the seller price or the delivery time better than in the agreement'*. The form of the proposed rules is ($p_{pr} = u_1$, $p_{dt} = u_2$ and y = DP):

$$R_j : \text{If } p_{pr} \text{ is } A_{1(j)} \text{ AND } p_{dt} \text{ is } A_{2(j)} \text{ Then DP is } B_{(j)},$$

where $A_{i(j)}$ and $B_{(j)}$ is the fuzzy set representing the jth linguistic value for the input parameter i and for the output parameter DP, respectively. We adopt triangular membership functions for the entire set of inputs and outputs. Concerning DP, we consider three fuzzy sets: *negative*, *stable* and *positive*. A 'negative' DP indicates that the thread should decrease T_b, a 'stable' DP indicates that T_b should remain as it is and 'positive' DP indicates that T_b should be increased. For the defuzzification process, we use the *Center-of-Gravity (CoG)* approach. Hence, the final value of the DP is calculated by:

$$DP = \frac{\sum_{i=1}^{N} \mu_j(u_i) \cdot u_i}{\sum_{i=1}^{N} \mu_j(u_i)}.$$

The strategy of the thread can be mapped into a set of fuzzy rules in order to result if the thread increases or decreases T_b. We present our FL rule base in Table 2.

Table 2: The proposed FL rule base.

Rule	p_{pr}	p_{dt}	DP
1	low	low or medium	negative
2	low	high	stable
3	medium	low	negative
4	medium	medium or high	stable
5	high	low	stable
6	high	medium or high	positive

This article can be cited as K. Panagidi, K. Kolomvatsos and S. Hadjiefthymiades, An Intelligent Scheme for Concurrent Multi-Issue Negotiations, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 129-149, 2014.

5. EXPERIMENTAL EVALUATION

We repost on the performance of the proposed models and present numerical results for widely used metrics. The performance metrics adopted in our experiments are:

a) *the agreement ratio (AG)*: The AG indicates the negotiations that end with an agreement out of a number of negotiations,

b) *the Average Buyer Utility (ABU)* and *the Average Seller Utility (ASU)*: The ABU is the maximum utility that the buyer gains from all the successful threads (threads with an agreement). The ASU is defined as the utility gained in the seller side, and, finally,

c) *the Average Rounds (AR)*: AR is defined as the number of rounds required to reach an agreement out of the full horizon $T = \min(T_b, T_s)$. Actually, we examine the percentage of the required rounds on T.

We run experiments for different values of the buyer valuation (*V*) about the product. This value affects the product price. It is the price upper limit (reservation value) in the buyer side. We run 300 negotiations for $N_T = 50$ (threads), $I = 4$ (issues) and $V \in \{50, 300\}$. It should be noted that at the beginning of each experiment, we randomly choose intervals $[min_i, max_i]$ for each side (buyer and seller). These values are chosen in the interval [0, 100]. Moreover, we randomly select T_b and T_s in the interval [50, 100]. The seller's parameters are also randomly selected in the beginning of every experiment (e.g., the cost is randomly selected in the interval [10, 50]). For offers calculation, we adopt the methodology presented in (Fatima et al., 2002; Oprea, 2002), as already discussed in Section 3.1.

At first, we do not apply the proposed FL system in the buyer side. Table 3 presents our comparison results. Concerning the AG, all the approaches achieve a large number of agreements (the majority is above 90%). AHP and PSO start from a small value for ABU, which increases gradually. Simplex method is more stable. PSO reaches the performance of the Simplex when V = 300 while AHP remains at low levels. When valuation is large ($V \to 300$), particles have more space to converge to an optimal deal. Concerning the seller utility, AHP and Simplex have constant values ranging near to ASU = 0.3. The PSO achieves the greatest ASU value especially for large V. In general, PSO seems to be the most efficient for the buyer as it leads to higher ABU values even for small V while it achieves the higher ASU values (more 'fair' approach).

In Figure 2, we see our results for different I values. In general, the AG value is not affected by I (is close or over 90%) while the PSO achieves larger ABU when $I \to 32$. It is natural that in these cases, PSO is not profitable for the seller (see the ASU plot). PSO requires more time than the rest models when I is small ($I < 8$) while the opposite stands when I is large ($I \to 32$). In Table 4, we see our model performance for different N_T values. AHP achieves the best performance concerning AG and it is also stable for ABU and ASU values. PSO requires the most time for all N_T. The reason is that in these cases the number of particles increases and, thus, reaching the optimal solution is a more complicated task.

Table 3: Results for different V values.

V	Simplex Optimization				AHP				PSO			
	AG	**ABU**	**ASU**	**AR**	**AG**	**ABU**	**ASU**	**AR**	**AG**	**ABU**	**ASU**	**AR**
50	0.98	0.83	0.29	0.48	0.88	0.65	0.39	0.22	0.90	0.73	0.36	0.33
300	1.00	0.91	0.31	0.10	1.00	0.85	0.32	0.21	0.99	0.91	0.49	0.28

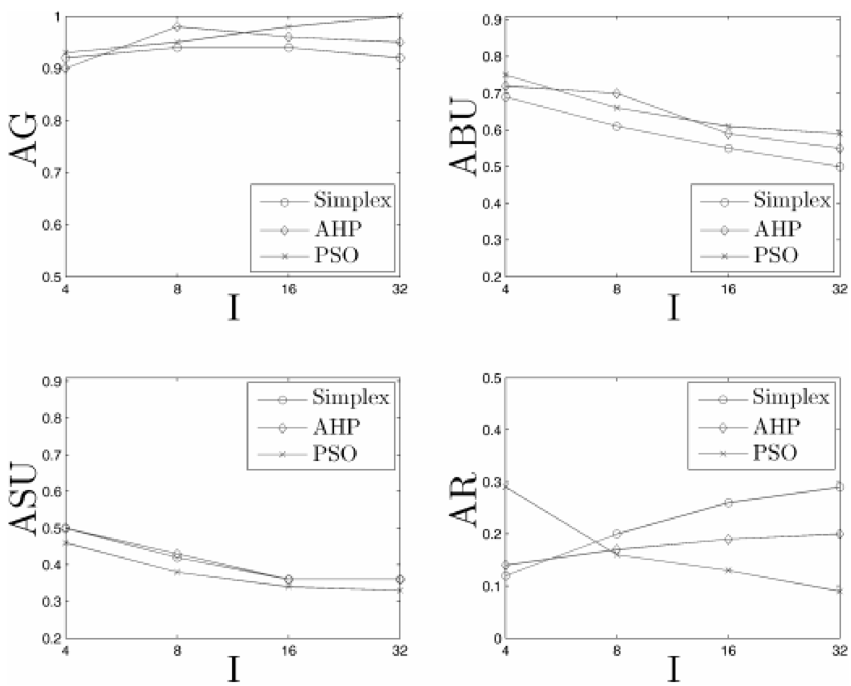

Figure 2. Results for different I values.

Table 4: Results for different N_T values.

N_T	Simplex Optimization				AHP				PSO			
	AG	ABU	ASU	AR	AG	ABU	ASU	AR	AG	ABU	ASU	AR
5	0.51	0.33	0.24	0.19	0.79	0.53	0.36	0.13	0.44	0.25	0.16	0.22
50	0.96	0.88	0.40	0.16	0.96	0.72	0.43	0.15	0.93	0.76	0.46	0.28

Figure 3 presents our results related to the optimality of our model. We run 10 experiments for different N_T values. When N_T = 32 our model has an average distance 0.0325 from the Pareto optimal with the minimum equal to 0.0017. The average distance is 0.1957, 0.1295 and 0.0736 for $N_T \in \{4, 8, 16\}$. For large N_T the buyer gains higher utility and our model approaches very close to the Pareto optimal. The greater the N_T is, the closer to the optimal the provided solution reaches as the buyer faces larger number of sellers and, thus, has more opportunities to reach the best deal.

The proposed model is very efficient for the buyer as the ABU is larger than the ASU. Additionally, the model needs very little time to result an agreement especially when AHP is used. Our ABU results outperform those presented in (Lau, 2005) where the maximum utility is equal to 0.64. It should be noted that, in (Lau, 2005), deadlines were equal to 200 rounds and agreements were concluded in the 84% of the deadlines (in our case the maximum AR value is 48%). Additionally, in (Nguyen & Jennings, 2004), the maximum ABU is equal to 0.75 while in our case is equal to 0.91. Finally, in (Robu et al., 2005), the presented model reaches the 97% of the optimal value while our model reaches the 99% of the optimal when N_T = 32 (see Figure 3).

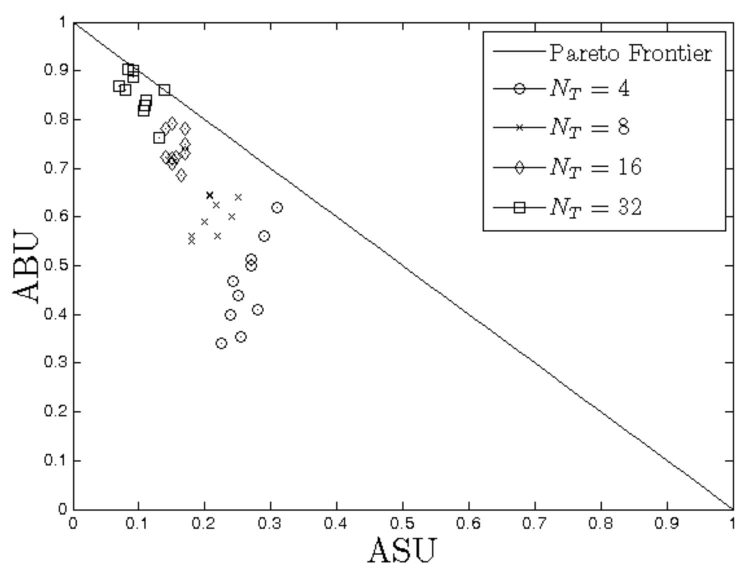

Figure 3. Optimality of the proposed model**.**

This article can be cited as K. Panagidi, K. Kolomvatsos and S. Hadjiefthymiades, An Intelligent Scheme for Concurrent Multi-Issue Negotiations, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 129-149, 2014.

Let us now study the effect of the proposed FL controller in the buyer behavior. We define the following metric:

$$D = \frac{D_{TARGET} - D_{BASE}}{D_{BASE}} \cdot 100\% ,$$

where the indication *TARGET* stands for the use of the FL controller and the indication *BASE* stands for the basic functionality of our proposed model (without the use of the FL controller). D is applied to the above discussed metrics for the buyer side i.e., AG, ABU.

We examine the performance of the system for different I values (I $\in$ {4, 8, 16, 32}). The performance of our system is shown in Figure 4 and Figure 5. We can notice that the integration of the FL controller generally improves the performance of the proposed architecture. Results of the Simplex method are improved as the FL controller increases the negotiation time. The increased negotiation time could lead to better agreement results. The AG is increased more than 5% and the ABU more than 40% especially when I → 32. The AHP methodology leads to a large increase in the AG values only when I → 4. For large I values, the difference remains at lower levels. Concerning the ABU, the use of the FL controller combined with the AHP does not lead to any important variations. Actually, the adoption of the FL controller does not affect the performance of the system. The FL controller combined with the PSO method raises the AG when I ≤ 8. For issues more than 8, there is a rise on the number of PSO particles trying to conclude the best agreement and, in general, the complexity of their movement in space increases as well. Finally, the FL controller positively affects the ABU in the PSO scenario. The greater the I is, the greater the ABU difference becomes. For instance, when I → 32, the use of the FL controller leads to 35% (approximately) more utility for the buyer.

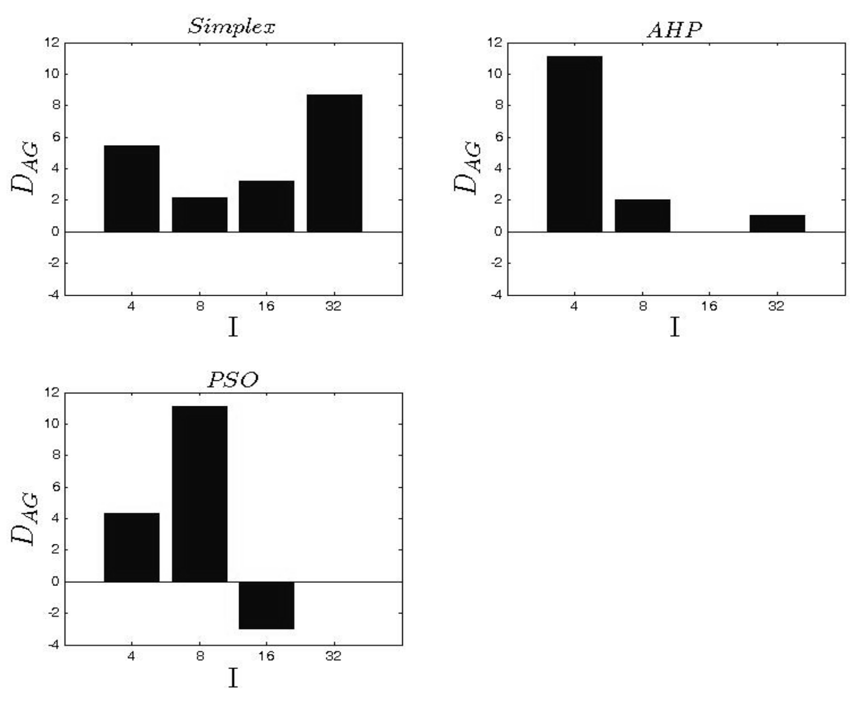

Figure 4. Results for the D_{AG} metric.

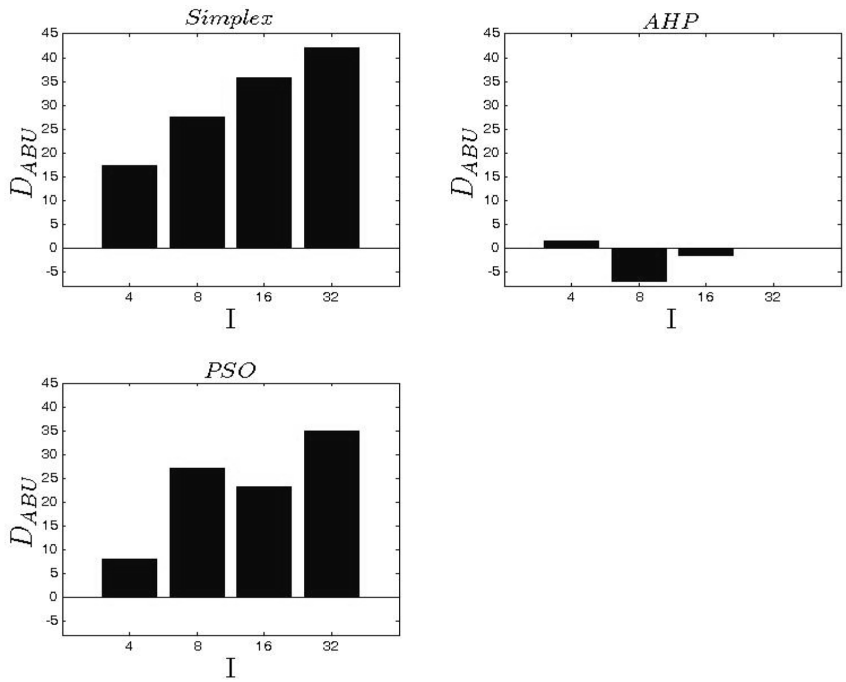

Figure 5. Results for the D_{ABU} metric.

In Table 5, we present our results for different N_T values. The Simplex methodology is that leading to the greater ABU value and seems to be very profitable for the buyer (for N_T = 50). As N_T increases, values of the examined metrics increase as well. When $N_T \rightarrow 50$, the AG reaches to the maximum value, indicating a 100% agreement percentage, in the entire set of the experiments. This stands for the three examined methodologies. Among the three, the PSO requires the greater AR value meaning that entities remain till the end of the negotiation. For instance, when $N_T \rightarrow 50$, the proposed PSO approach requires 71% of the total negotiation time T. Additionally, in the PSO case, the ASU remains at the lowest level which indicates that the discussed methodology is not profitable for the seller. It is Simplex that leads to a 'fair' negotiation result.

Let us now compare the difference in the performance of the proposed models when we use the FL controller and for different N_T values. Table 6 depicts our results. In general, the FL controller leads to a better performance. In the Simplex and the PSO case, the smaller the N_T is the greater the performance becomes. As $N_T \rightarrow 50$ the difference becomes smaller than in the previous case. The opposite stands for the AHP model. For N_T = 5, the use of the FL controller is not profitable while if we get N_T = 50, the use of the FL controller leads to better performance. PSO is more affected, however, the increase in the ABU reaches the 88% for small threads number (N_T = 5). Another interesting observation is that the FL controller increases the AR (number of rounds required to conclude the negotiation) except in the Simplex case. This stands for N_T = 5. The Simplex follows the rest of the

This article can be cited as K. Panagidi, K. Kolomvatsos and S. Hadjiefthymiades, An Intelligent Scheme for Concurrent Multi-Issue Negotiations, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 129-149, 2014.

models, when N_T = 50. In the PSO case, the increment in the performance demands a very large increment in the number of rounds. The greater the N_T is, the greater the difference becomes.

Table 5: Results for different N_T values (use of the FL controller).

N_T	Simplex Optimization				AHP				PSO			
	AG	**ABU**	**ASU**	**AR**	**AG**	**ABU**	**ASU**	**AR**	**AG**	**ABU**	**ASU**	**AR**
5	0.70	0.40	0.37	0.17	0.57	0.34	0.22	0.16	0.73	0.47	0.30	0.49
50	1.00	0.89	0.48	0.19	0.97	0.75	0.32	0.17	1.00	0.86	0.26	0.71

Table 6: Performance difference with the use of the FL controller.

		N_T = *5*	N_T = *50*
Simplex	D_{AG}	37.25%	4.17%
	D_{ABU}	21.21%	1.14%
	D_{ASU}	54.17%	20.00%
	D_{AR}	-10.53%	18.75%
AHP	D_{AG}	-27.85%	1.04%
	D_{ABU}	-35.85%	4.17%
	D_{ASU}	-38.89%	-25.58%
	D_{AR}	23.08%	13.33%
PSO	D_{AG}	65.91%	7.53%
	D_{ABU}	88.00%	13.16%
	D_{ASU}	87.50%	-43.48%
	D_{AR}	122.73%	153.57%

6. CONCLUSION AND FUTURE EXTENSIONS

In this paper, we focus on one-to-many, concurrent negotiations. The proposed solution is an automated process of a dynamic and independent change of the adopted strategy of the IAs involved. We present our proposal for optimizing negotiations focusing on the buyer's side. Specific models for the dynamic adaptation of utility function weights are analyzed. We also propose a PSO approach to lead the buyer to the optimal solution. A large number of experiments (negotiations) show that the Simplex method leads to an increased buyer utility. The PSO approach is mainly affected by the number of threads, however, it is an efficient technique when threads number is small. By using PSO, when issues number increases, the buyer utility decreases. Moreover, we propose the use of Fuzzy

Logic in the buyer side. We define a Fuzzy Logic controller responsible to update the buyer behavior during negotiation. The controller is based on the known Kernel Density Estimator for predicting the seller strategy for specific issues and adopting its results the controller decides changes in the deadline for each thread. Threads are fully adapted in the course of the negotiation as well as in the agreements concluded in other threads. The novelty of our work is the combination of optimization techniques with computational intelligence models in order to provide an intelligent and efficient methodology for handling uncertainty in negotiations. The strengths of the collective behavior (Swarm Intelligence) is combined with the advantages of the KDE and Fuzzy Logic. Our model could be applied in real life scenarios where uncertainty on the behavior of entities is the common case. Future work involves the definition of relevant function for weights adaptation in the seller side and extensive experiments in order to reveal the advantages of the proposed model.

7. REFERENCES

Ali, M. Z., Alkhatib, K., Tashtoush, Y., 2013, Cultural Algorithms: Emerging Social Structures for the Solution of Complex Optimization Problems, *International Journal of Artificial Intelligence*, vol. 11(A13), 20-42.

An, B., Sim, K., Tang, L., Li, S. Q., Cheng, D. J., 2006, Continuous-time Negotiation Mechanism for Software Agents, *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 36, 1261-1272.

Baarslag, T., Fujita, K., Gerding, E., Hidriks, K., Jennings, N., Jonker, C., Kraus, S., Lin, R., Robu, V., Williams, C., 2013, Evaluating Practical Negotiating Agents: Results and Analysis of the 2011 International Competition, *Artificial Intelligence*, 198, 73-103.

Bui, H., Venkatesh, S., Kieronska, D., 1995, *An Architecture for Negotiating Agents that Learn*, Technical Report, Department of Computer Science Curtin University of Technology, Perth, Australia.

Chatterjee, K., Samuelson, L., 1988, Bargaining Under Two-Sided Incomplete Information: The Unrestricted Offers Case, *Operations Research*, 36(4), 605-638.

Chen, Y. M., Huang, P. N., 2009, Agent-Based Bilateral Multi-Issue Negotiation Scheme for E-Market Transactions, *Applied Soft Computing*, 9, 1057-1067.

Cheng, D., Chan, C., Lin, C., ‘Buyer-supplier negotiation by fuzzy logic based agents’, 3rd Int. Conference on Information Technology and Applications, 2005, pp. 137-142.

Crampton, P. C., 1984, Bargaining With Incomplete Information: An Infinite-Horizon Model With Two-Sided Uncertainty, *Review of Economic Studies*, 579-593.

Dagdeviren, M., Yüksel., I., 2008, Developing a fuzzy analytic hierarchy process (AHP) model for behavior-based safety management, *Information Sciences*,178(6), 1717–1733.

Da-Jun, C., Liang-Xian, X., 2002. A Negotiation Model of Incomplete Information Under Time Constraints, *AAMAS*, Bologna, Italy, 128-134.

Engelbrecht, A. P., 2007, Computational Intelligence. An Introduction, *2nd Edition, John Wiley and Sons Ltd*.

Faratin, P., Sierra, C., Jennings, N. R., 1998, Negotiation Decision Function for Autonomous Agents, *International Journal of Robotics and Autonomous Systems*, 24 (3-4), 1998, 159-182.

Fatima, S. S., Wooldridge, M., Jennings, N., 2005, Bargaining with Incomplete Information, *Annals of Mathematics and Artificial Intelligence*, 44(3), 207-232.

Fudenberg, D., Levine, D. K., Tirole, J., 1987, Incomplete Information Bargaining With Outside Opportunities, *The Quarterly Journal of Economics*, 102(1), 37-50.

Gerding, E., van Bragt, D., 2003, Multi-Issue Negotiation Processes by Evolutionary Simulation, Validation and Social Extensions, *Computational Economics*, 22(1), 39-63.

Hong, W.-C., Dong, Y., Chen, L.-Y., Wei, S.-Y., 2011, SVR with Hybrid Chaotic Genetic Algorithms for Tourism Demand Forecasting, *Applied Soft Computing*, vol. 11(2), 1881-1890.

Howard, A., Mataric, M. J., Sukhatme, G. S., 2002, Mobile sensor network deployment using potential field: A distributed scalable solution to the area coverage problem, In Proceedings of the 6th International Conference on Distributed Autonomous Robotic Systems (DARS02), 299–308.

Jazayeriy, H., Azmi-Murad, M., Sulaiman, N., Uzidir, I., 2011, Pareto-Optimal Algorithm in Bilateral Automated Negotiation, *International Journal of Digital Content Technology and its Applications*, 5(3), 1-11.

Jonker, C., van der Meij, L., Robu V., Treur J., 2004, Demonstration of a Software System for Automated Multi-Attribute Negotiation, *AAMAS*, New York City, USA.

Kennedy, J., Eberhart, R. C., 1995, Particle swarm optimization, *Proc. IEEE International Conference on Neural Networks*, 1942–1948.

Kolomvatsos, K., Anagnostopoulos, C., Hadjiefthymiades, S., 2008, On the use of fuzzy logic in a seller bargaining game, *32nd IEEE COMPSAC*, 184-191.

Kolomvatsos, K., Hadjiefthymiades, S., 2008, Implicit Deadline Calculation for Seller Agent Bargaining in Information Marketplaces, *Proceedings of the 2nd International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2008)*, 184-190.

Kong, F. Liu, H., 2005, Applying fuzzy analytic hierarchy process to evaluate success factors of e-commerce, *International journal of information and systems sciences*.

Lau, R.Y.K., 2005, Towards Genetically Optimised Multi-Agent Multi-Issue Negotiations, *Proceedings of the 38th Annual Hawaii International Conference*.

Luo, X., Jennings, N., Shadbolt, N., Leung, H., Lee, J., 2003, A fuzzy constraint based model for bilateral, multi-issue negotiations in semicompetitive environments, *Artificial Intelligence*, 148, 53–102.

Mataric, H. M., Sukhatme, S., 2002, Mobile Sensor Network Deployment Using Potential Field: a distributed scalable solution to the area coverage problem, *Proc. of ICDARS*.

Narayanan, V. Jennings, N. R., 2005, An adaptive bilateral negotiation model for e-commerce settings, *7th International IEEE Conference on E-Commerce Technology*, 34-39.

Nguyen T. D., Jennings N. R., 2003, Concurrent Bi-Lateral Negotiation in Agent Systems, *4th DEXA Workshop on E-Negotiations*, 844-849.

Nguyen, T. D., Jennings, N. R., 2003, A Heuristic Model for Concurrent Bilateral Negotiations in Incomplete Information Settings, *Proceedings of IJCAI*, 1467-1469.

Nguyen, T. D., Jennings, N., 2004, Coordinating multiple concurrent negotiations, *Proc. AAMAS* 1064–1071.

Oliver, J. R., 1997, A Machine Learning Approach to Automated Negotiation and Prospects for Electronic Commerce, *Journal of Management Information Systems*, 13(3), 83-112.

This article can be cited as K. Panagidi, K. Kolomvatsos and S. Hadjiefthymiades, An Intelligent Scheme for Concurrent Multi-Issue Negotiations, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 129-149, 2014.

Oprea, M., 2002, An adaptive negotiation model for agent-based electronic commerce, *Studies in Informatics and Control*, 11 (3), 271–279.

Precup, R.-E., David, R.-C., Petriu, E. M., Preitl, S., Radac, M.-B., 2012, Novel Adaptive Gravitational Search Algorithm for Fuzzy Controlled Servo Systems, *IEEE Transactions on Industrial Informatics*, vol. 8(4), 791-800.

Raeesy, Z., Brzostwoski, J., Kowalczyk, R., 2007, Towards a fuzzy-based model for human-like multi-agent negotiation, *IEEE/WIC/ACM Conference on Intelligent Agent Technology*, 515-519.

Rahwan, I., Kowalczyk, R. Pham, H., 2002, Intelligent Agents for Automated One-to-Many E-Commerce Negotiation, *25th Australian Computer Science Conference*, 197-204.

Raiffa, H., 1982, The Art and Science of Negotiation, *Harvard University Press*, Cambridge.

Robu, V., Somefun, D.J.A., La Poutré, J.A., 2005, Modeling Complex Multi-Issue Negotiations using Utility Graphs, *AAMAS*, New York, USA, 280-287.

Satyala, N. T., Pieper, R. J., 2008, A Unified Approach for Predicting Long and Short-Term Capability Indices with Dependence on Manufacturing Target Bias, *International Journal of Quality, Statistics and Reliability*.

Sun, T., Zhu, Q., Li, S., Zhou, M., 2007, Open, Dynamic and Continuous One-to-Many Negotiation System, *Proceedings of the 2nd International Conference on Bio-Inspired Computing: Theories and Applications*, 87-93.

Torroni, P., Toni, F., 2001, Extending a Logic Based One-to-One Negotiation Framework to One-to-Many Negotiation, *Proc. of the WESAW*, London, UK, 105-118.

Türkay, D., Koray, A., 2012, Modified Even-Swaps: A novel, clear, rational and an easy-to-use mechanism for multi-issue negotiation, *In Computers & Industrial Engineering*, 63(4), 1013-1029.

Wand, M. P., Jones, M. C., 1995, Kernel Smoothing, *Chapman and Hall*.

Williams, C., Robu, V., Gerding, E., Jennings, N., 2012, Negotiating Concurrently with Unknown Opponents in Complex, Real-Time Domains, *20th European Conference on Artificial Intelligence*, Montpellier, France, 834-839.

Wu, M., Weerdt, M., Poutre, H., 2009, Efficient Methods for Multi Agent Multi-Issue Negotiation: Allocating Resources, *12th Intern. Conference on Principles of Practice in MAS*, 97-112.

Yazdani, D., Nasiri, B., Azizi, R., Sepas-Moghaddam, A., Meybodi, M. R., 2013, Optimization in Dynamic Environments Utilizing a Novel Method Based on Particle Swarm Optimization, *International Journal of Artificial Intelligence*, vol. 11(A13), 170-192.

Zeng, D., Sycara, K., 1998, Bayesian Learning in Negotiation, *International Journal of Human-Computer Studies*, 48(1), 125-141.

Zuo, B.-H., Sun, Y., 2009, Fuzzy Logic to Support Bilateral Agent Negotiation in E-Commerce, *Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence*, Shanghai, China, 179-183.