Towards Genetically Optimised Multi-Agent Multi-Issue Negotiations

Raymond Y.K. Lau
Centre for Information Technology Innovation
Faculty of Information Technology
Queensland University of Technology
GPO Box 2434, Brisbane, Qld 4001, Australia
E-mail: r.lau@qut.edu.au

Abstract

Classical negotiation models are based on a centralised decision making approach which assumes the availability of complete information about negotiators and unlimited computational resources. These negotiation mechanisms are ineffective for supporting real-world negotiations. This paper illustrates an agent-based distributive negotiation mechanism where each agent's decision making model is independent to each other and is underpinned by an effective evolutionary learning algorithm to deal with complex and dynamic negotiation environments. Initial experimental results show that the proposed genetic algorithm (GA) based adaptive negotiation mechanism outperforms a theoretically optimal negotiation mechanism in environments constrained by limited computational resources and tough deadlines. Our research work opens the door to the development of practical negotiation systems for real-world applications.

Keywords: *Automated Negotiations, Autonomous Agents, Genetic Algorithm, Decision Making Model.*

1 Introduction

As our world is constrained by limited resources, negotiations have long been taken as the essential activities in human society. Negotiations are ubiquitous and conducted in various contexts such as the formation of virtual enterprises [12], marketing, establishing business contracts, managing labour dispute [33], resolving border conflicts, handling hostage crisis, etc. Negotiation refers to the process by which group of agents (human or software) communicate with one another in order to reach a mutually acceptable agreement on resource allocation (distribution) [23]. In general, negotiators strike for maximising their individual payoffs while ensuring that an agreement is reached [20, 30]. Given the ubiquity and importance of negotiations in various contexts, research into negotiation theories and techniques has attracted attention from multiple disciplines such as Distributed Artificial Intelligence (DAI) [19, 17, 33], Social Psychology [2, 27, 28], Game Theory [26, 34, 31], Operational Research [5, 13], and more recently in agent-mediated electronic commerce [8, 12]. Despite the variety of approaches towards the study of negotiation theory, a negotiation model consists of four main elements: *negotiation protocol*, *negotiation strategies*, *negotiation environment*, and *agent environment* [9, 20, 23]. An abstract view of a negotiation model is depicted in Figure 1.

Negotiation Protocol refers to the set of rules that govern the interactions among negotiators. The rules specify the types of participants (e.g., the existence of a negotiation mediator), the valid states (e.g., waiting for bid submission, negotiation closed), the actions that cause negotiation state changes (e.g., accepting an offer, quit a negotiation session).

Negotiation Strategies refer to the decision making apparatus the agents employ to act in line with the negotiation protocol, the negotiation environment, and the agent environment to achieve their objectives. For instance, negotiators should set stringent goals initially and concede first on issue of lesser importance to achieve higher payoffs in a competitive environment [2, 28].

Negotiation Environment refers to factors that are relevant to problem domain. These factors include the number of negotiation issues, number of parties, time constraints, nature of domain (e.g., purely competitive vs. cooperative), etc.

Agent Environment refers to the characteristics of the participants (agents). These characteristics include an agent's attitude (e.g., self-interested vs. benevolent), cognitive limitations (e.g., omniscient agent vs. memoryless agent), goal setting, initial offer magnitude, knowledge and experience (e.g., knowledge about the opponents), etc.

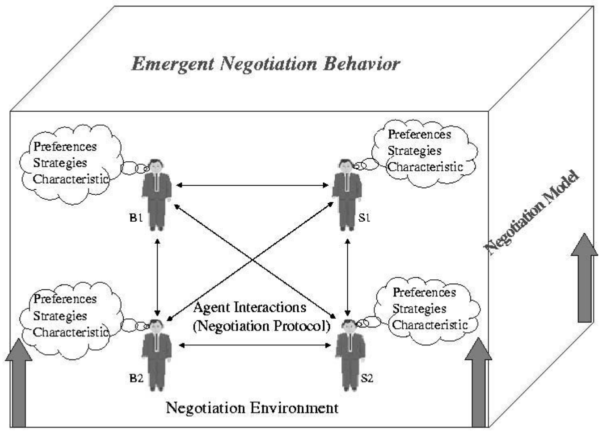

Figure 1. The Elements of a Negotiation Model

A *negotiation mechanism* refers to a particular negotiation protocol and the decision making model that operates under the protocol [23]. Typical negotiation situations (e.g., negotiating business contracts) are characterised by huge negotiation spaces involving multiple parties and dozens of issues (i.e., negotiation attributes). Under such circumstance, even the most experienced human negotiators will be overwhelmed by the non-linear negotiation dynamics. Consequently, sub-optimal rather than optimal deals are often reached and the phenomenon of "leaving some money on the table" may occur [28]. It is generally believed that automated negotiation systems are more efficient than human negotiators in combinatorially complex negotiation situations [20, 23]. However, a centralised negotiation decision making mechanism is unable to model the realistic negotiation scenarios where each agent is characterised by its own preferences and employs distinct strategies (i.e., decision making models) to search for solutions. To this end, the notion of autonomous agents [14] provides an effective and robust approach for the development of practical negotiation systems.

Autonomous agents are encapsulated computer systems situated in some environments and are capable of flexible, *autonomous* actions in that environment to meet their design objectives [14]. In an agent-based negotiation system, a group of software agents communicate and autonomously make negotiation decisions on behalf of their human users. There is not a centralised decision making mechanism which assumes to have complete knowledge about every agent's characteristics and preferences. Instead, the *emergent* negotiation behavior among a group of agents is revealed due to the interactions among the individual autonomous agents (i.e., the negotiation dynamics) as highlighted in Figure 1. Emergent behaviour is the overall behaviour that results from lower level factors such as individuals and their interactions [20]. It is based on the view that the nonlinear and dynamic aspects of organisational behaviour are in fact the result of the interaction of adaptive decision makers. This paper illustrates an efficient agent-based negotiation mechanism. In particular, it focuses on a negotiation agent's decision making model which is underpinned by a genetic algorithm (GA).

1.1 The problems

Realistic negotiation situations are characterised by combinatorially complex negotiation spaces which involve multiple parties and many issues. In addition, negotiators are bounded by limited computational resources (i.e., *bounded rationality*), time, and limited information about the opponents. While classical game-theoretic negotiation models [34, 31] provide excellent theoretical analysis of the optimal outcomes (e.g., Nash equilibrium) given the somewhat restricted scenarios (e.g., bi-lateral negotiations), these normative theories fail to advise the course of actions that a negotiator can follow to reach the optimal outcome in real-world negotiations. One main concern for the practical use of these normative theories is that the search space of considering all the possible strategies and interactions in order to identify the equilibrium solutions grows exponentially. It means that the problem of finding an optimal strategy is in general computationally intractable. Another problem is that classical game theories assume that complete information about every agent is available to a *centralised* decision making mechanism. It turns out that such an assumption does not hold in most real-world negotiation situations [9, 20]. Newer models of games which are based on incomplete information about the opponents (e.g., in the form of priori probability distributions about others' strategies) have been proposed [11]. Nevertheless, it is still problematic in finding an automated means of acquiring or estimating accurate probability distributions given a negotiation situation.

1.2 Requirements of practical negotiation systems

Practical negotiation mechanisms must be computationally efficient and should not be based on a centralised decision making approach. In other words, negotiation agents should be developed based on the assumption of *bounded* rather than *perfect rationality* [23]. Moreover, these agents must be sensitive to *time pressure* present in most real-world negotiation situations. As an agent only knows its own preferences (utility function) but not the preferences of its opponents [15, 7], negotiation agents must be able to make sensible decision based on incomplete and uncertain information. One of the ways to do this is to equip negotiation agents with effective learning mechanisms to gradually learn good strategies by observing the opponents' moves in a negotiation process. Moreover, since the preferences of an agent and its opponents may change over time, effective negotiation mechanisms must be *adaptive* to the changing agent environment. To support a variety of real-world negotiation scenarios, negotiation agents should be able to act flexibly and mimic a wide spectrum of negotiation attitudes (e.g., from fully self-interested to fully benevolent) rather than limited by a few pre-defined negotiation attitudes. Finally, as real-world negotiations often involve multiple parties who will exploit many issues (e.g., price, quantity, product quality, etc.) simultaneously, practical negotiation mechanisms must be able to support multi-lateral multi-issue negotiations.

1.3 Related work

It has been argued that the challenge of research in negotiation is to develop models that can track the shifting tactics of negotiators [20]. Accordingly, a genetic algorithm based negotiation mechanism is developed to model the dynamic *concession matching behaviour* arising in bilateral negotiation situations [20]. The set of feasible offers of a negotiation agent is represented by a population of chromosomes. The fitness of each chromosome (i.e., a feasible offer) is measured by the fitness function developed based on Social Judgement Theory. By employing the standard genetic operators such as selection, crossover, and mutation, a population of chromosomes evolves over time. After a pre-defined number of evolutions, the fittest chromosome from the current generation is chosen as a tentative solution (i.e., a counter-offer). The main weakness of this negotiation model is that a subjective estimation of the opponent's utility function is required to compute member fitness and to evaluate an incoming counter-offer. The negotiation agents cannot learn the opponents' preferences and adapt their strategies accordingly. Furthermore, the proposed model is only evaluted in bilateral negotiation scenarios.

Rubenstein-Montano and Malaga have also reported a GA-based negotiation mechanism for searching optimal solutions for multiparty multi-objective negotiations [30]. Basically, a negotiation problem is treated as a multi-objective optimisation problem. Apart from the standard genetic operators such as selection, crossover, and mutation, the GA is enhanced with a new operator called *trade*. The trade operator simulates a concession making mechanism which is often used in negotiation systems. However, the main problem of their particular GA-based negotiation mechanism is that the preferences (i.e., the utility functions) of all the negotiation parties are assumed available to a central negotiation mechanism. Moreover, the preferences of the negotiation parties are assumed unchanged during a negotiation process.

Genetic algorithm has also been applied to learn effective rules to support the negotiation processes [25]. A chromosome represents a negotiation (classification) rule rather than an offer. The fitness of a chromosome (a rule) is measured in terms of how many times the rule has contributed to reach an agreement. In order for the system to determine if an agreement is possible, each negotiator's preferences including the reservation values of the negotiation attributes are assumed known or hypothesised. Therefore, this approach also suffers from the same problem as that of the other methods which assume complete information about negotiation spaces.

Instead of using the evolutionary approach to develop an agent's decision making model, a GA has been used to learn optimal negotiation tactics given a particular negotiation situation (e.g., a predefined amount of negotiation time) [24]. Even though such a theoretical anlysis helps identify optimal negotiation parameters, the proposed negotiaton model is not applicable to build practical negotiation system because the model is based on a centralised decision making architecture where complete information about each player is available. Similarly, a GA is also used to study the bargaining behaviour of boundedly rational agents in a single issue (e.g., price) bi-lateral negotiation situation [8]. One of the findings is that if both the buyer agent and the seller agent have a negative utility discount factor (i.e., worse off for delayed agreement), the stable outcome generated by the GA model always matches the equilibrium outcome identified by the game-theoretic model. Other approaches such as Bayesian learning [35], Case-Based Reasoning (CBR) [33], Belief Revision [22], and Argumentative Logic [18] have also been explored to model the non-linear negotiation dynamics present in most real-world negotiation scenarios.

1.4 Justifications of the proposed approach

GAs have long been taken as heuristic search methods to find optimal or near optimal solutions from large search spaces [10]. As a matter of fact, GAs have been successfully applied to develop adaptive information agents [21] and employed to build negotiation systems [3, 20]. For most real-world negotiations, only very limited information about the negotiation spaces (e.g., the opponents' preferences) is available. GAs are just able to search for feasible solutions efficiently without requiring detailed structural information about the search space as input. It has been shown that a GA can identify the same optimal solutions as that derived from a game-theoretic model under certain conditions [8]. Since GAs are based on the evolution principle of "natural selection", they are effective in modelling *dynamic negotiation environments* where good negotiation strategies evolve according to negotiators' changing preferences. In general, GA-based negotiation approach fulfils the general requirements of developing practical negotiation systems in terms of computational efficiency, scalability, adaptability, sensitivity to deadlines, and the ability of making decisions based on incomplete information.

1.5 Contributions of the paper

Although many negotiation models have been reported in the literature [11, 18, 30, 31, 35], these models are ineffective for solving real-world negotiation problems because they are either assuming a centralised decision making mechanism with complete knowledge about the negotiation spaces, computationally intractable, ignoring the time pressure, incapable of learning opponents' preferences, or restricted in dealing with single issue bi-lateral negotiations only. This paper discusses a novel agent-based multi-party multi-issue negotiation mechanism which alleviates all the above problems. In particular, a negotiation agent's decision making model is underpinned by an effective and efficient genetic algorithm. In addition, rigorous quantitative evaluation of the proposed negotiation mechanism is performed.

2 A Basic Negotiation Mechanism

A sequential alternate-offering negotiation protocol, a variant of the *monotonic concession protocol* [29], is employed by our negotiation system in which negotiation proceeds in a discrete series of rounds. In each round, each agent puts forward an offer in alternate. If these offers overlap, it means that an agreement is reached. If the offers do not overlap, negotiation proceeds to the next round where the agents make a concession. If there is no agreement after the deadline is reached, an agent decides to quit and the negotiation ends with a conflict. The basic negotiation mechanism illustrated in this section is based on multi-attribute utility theory (MAUT) [16] and is first discussed in [1].

A negotiation space $Neg = < P, A, D, U, T >$ is a 5-tuple which consists of a finite set of negotiation parties (agents) P, a set of attributes (i.e., negotiation issues) A understood by all the parties $p \in P$, a set of attribute domains D for A, and a set of utility functions U with each function $U_p^o \in U$ for an agent $p \in P$. An attribute domain is denoted D_{a_i} where $D_{a_i} \in D$ and $a_i \in A$. An utility function pertaining to an agent p is defined by: $U_p^o : D_{a_1} \times D_{a_2} \times \ldots \times D_{a_n} \mapsto [0, 1]$. Each agent p has a deadline $t_p^d \in T$. It is assumed that information about P, A, D is exchanged among the negotiation parties during the *ontology sharing* stage before negotiation actually takes place. A *multi-lateral* negotiation situation can be modelled as many one-to-one *bi-lateral* negotiations where an agent p maintains a separate negotiation dialog with each opponent. In a negotiation round, the agent will make an offer to each of its opponent in turn, and consider the most favourable counter-offer from among the set of incoming offers evaluated according to its own payoff function U_p^o.

2.1 Representing offers

An *offer* $\overrightarrow{o} = < d_{a1}, d_{a2}, \ldots, d_{an} >$ is a tuple of attribute values (intervals) pertaining to a finite set of attributes $A = \{a_1, a_2, \ldots, a_n\}$. An offer can also be viewed as a vector of attribute values in a geometric negotiation space with each dimension representing a negotiation issue. Each attribute a_i takes its value from the corresponding domain D_{a_i}. Generally speaking, a finite set of candidate offers O_p acceptable to an agent p (i.e., satisfying its hard constraints) is constructed via the Cartesian product $D_{a1} \times D_{a2} \times \cdots \times D_{an}$. As human agents tend to specify their preferences in terms of a range of values, a more general representation of an offer is a tuple of attribute value intervals such as $o_i =< 20 - 30(\$), 1 - 2(years), 10 - 30(days), 100 - 500(units) >$.

2.2 Representing negotiation preferences

The *valuations* of individual attributes and attribute values (intervals) are defined by the valuation functions $U_p^A : A \mapsto [0, 1]$ and $U_p^{D_a} : D_a \mapsto [0, 1]$ respectively, whereas U_p^A is an agent p's *valuation function* for each attribute $a \in A$, and $U_p^{D_a}$ is an agent p's valuation function for each attribute value $d_a \in D_a$. In addition, the valuations of attributes are assumed normalised, that is, $\sum_{a \in A} U_p^A(a) = 1$. One common way to quantify an agent's preference (i.e., the utility function U_p^o) for an offer o is by a linear aggregation of the *valuations* [1, 32]: $U_p^o(o) = \sum_{a \in A} U_p^A(a) \times U_p^{D_a}(d_a)$.

2.3 Computing concessions and generating offers

If an agent's initial proposal is rejected by its opponent, it needs to propose an alternative offer with the least utility decrement (i.e., computing a concession). An agent will maintain a set $O_p^{'}$ which contains the offers it has proposed before or the offer to be proposed in the current round. In a negotiation round, an alternative offer with a concession can be determined based on:

$$\exists_{o_{counter} \in \{O_p - O'_p\}} \forall_{o_x \in \{O_p - O'_p\}} : [o_x \preceq_p o_{counter}]$$

$o_x \preceq_p o_y$ denotes that an offer o_y is more preferable than another offer o_x. The above statement defines that the alternative offer $o_{counter}$ is the most preferable offer from among the set of feasible offers which have not been proposed before. The preference relation $\preceq_p$ is a total ordering induced by an agent p's utility function U_p^o over the set of feasible offers O_p. In other words, the feasible offers of an agent p are ranked in descending order of utility driven by $(\preceq_p, O_p)$. An alternative offer with concession is picked up from the top of the list ranked by $(\preceq_p, \{O_p - O_p^{'}\})$ in a negotiation round.

2.4 Evaluating incoming offers

When an incoming offer o is received from an opponent, an agent p first evaluates if $o \in O_p$ is true (i.e., the offer satisfying all its hard constraints). To do this, an *equivalent offer* $o_{\simeq}$ should be computed. $o_{\simeq}$ represents agent p's interpretation about the opponent's proposal o. Once $o_{\simeq}$ for o is computed, acceptance of the incoming offer o can be determined with respect to p's own preference $(\preceq_p, O_p)$. An offer $o_{\simeq} \in O_p$ is equivalent to o iff every attribute interval of $o_{\simeq}$ *intersects* each corresponding attribute interval of o. Formally, any two attribute intervals d_x, d_y intersect if the intersection of the corresponding sets of points is not empty (i.e., $\{d_x\} \cap \{d_y\} \neq \emptyset$). The acceptance criteria for an incoming offer o (i.e., the equivalent $o_{\simeq}$) is defined by:

1. If $\forall_{o_x \in O_p} \; o_x \preceq_p o_{\simeq}$, an agent p should accept o since it produces the maximal payoff.

2. If $o_{\simeq} \in O_p^{'}$ is true, an agent p should accept o because $o_{\simeq}$ is one of its previous proposals or is the one to be proposed in the current round.

It has been proved that if each participating agent $p \in P$ employs their preference ordering $(\preceq_p, O_p)$ to compute concessions and uses the offer acceptability criteria described above to evaluate incoming offers, *Pareto optimal* is always found if it exists in a negotiation space [1]. A solution is Pareto optimal if it is impossible to find another solution such that at least one agent will be better off but no agent will be worse off [29]. However, Pareto optimal does not necessarily lead to maximal joint payoff. Joint payoff simply refers to the sum of each agent's payoff obtained at the end of a negotiation.

3 Genetically Optimised Negotiation Agents

The decision making models of the GA-based adaptive negotiation agents are developed based on the basic intuition that negotiators tend to maximise their individual payoffs while ensuring that an agreement is reached [7, 20]. In each negotiation agent, a population of chromosomes is used to represent a subset of feasible offers. Since an agent knows its own utility function, an offer o_{max} representing the offer with the maximal payoff can be identified. In addition, an offer $o_{opponent}$ represents the opponent's most recent counter-offer. According to the basic intuition, a feasible offer is considered *fit* if it is close to o_{max} and $o_{opponent}$. The distance from a feasible offer o_i to o_{max} and $o_{opponent}$ can be measured based on standard distance function such as the *Weighted Euclidean distance* [4]. In each negotiation round, the offer vectors $o_{opponent}$ and o_{max} may change, and so are the offers considered fit by the agent. In other words, an agent is learning and adapting to the opponent's preferences gradually. For multi-lateral negotiations, the best incoming offer evaluated based on an agent's private utility function is taken as the $o_{opponent}$ by the GA. Thereby, each agent only needs to maintain one population of chromosomes representing a set of relatively good offers.

3.1 Offer encoding

Each GA-based negotiation agent p utilises a population of chromosomes to represent a subset of feasible offers $O_p^{feas} \subseteq O_p$. Each chromosome consists of a fixed number of fields. The first field uniquely identifies a chromosome, and the second field is used to hold the fitness value of the chromosome. The other fields (genes) represent the attribute values of a candidate offer. Figure 2 depicts the decimal encoding (Genotype) of some chromosomes and a two-point crossover operation. The genetic operators such as crossover and mutation are only applied to the genes representing the attribute values of an offer.

3.2 The fitness function

The top t chromosomes (candidate offers) with the highest fitness are selected from a population to build the solution set S. If the size of S is 1, it means that the fittest chromosome from a population is chosen as an offer. In general, a stochastic selection function is applied to the solution set to choose a member as the solution (i.e., the current offer) in a particular negotiation round. Ideally, a fitness function should reflect the *joint payoff* of each candidate offer. Unfortunately, the utility functions of the opponents are normally not available for real-world applications. Therefore,

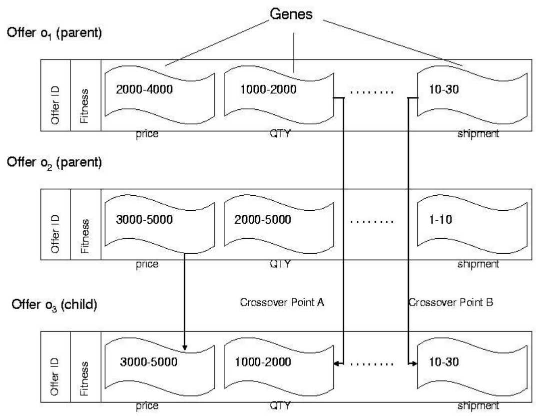

Figure 2. Encoding Candidate Offers

the proposed fitness function approximates the ideal function and captures three important issues: an agent's own payoff, the opponent's *partial preference* (e.g., the most recent counter offer), and time pressure:

$$fitness(o) = \alpha \times TP(t) \times \frac{U_p^o(o)}{U_p^o(o_{max})} + (1 - \alpha \times TP(t)) \times (1 - \frac{dist(\overrightarrow{o}, \overrightarrow{o}_{opponent})}{MaxDist(|A|)}) \quad (1)$$

where o_{max} represents an offer which produces the maximal payoff based on an agent p's current utility function; $\overrightarrow{o}_{opponent}$ is the offer vector representing the most recent counter-offer proposed by an opponent. The parameter $\alpha \in [0, 1]$ is the *trade-off factor* to control the relative importance of optimising one's own payoff or reaching a deal (e.g., by considering the opponent's recent offer). In other words, α is used to model a wide spectrum of agent attitudes, from fully self-interested ($\alpha = 1$) to fully benevolent ($\alpha = 0$).

The term $MaxDist(|A|)$ represents the maximal distance of a geometric negotiation space. It can be derived from the number of dimensions $|A|$ if each dimension (attribute) is scaled in the unit interval by linear scaling: $d_i^{\text{scaled}} = \frac{d_i - d_i^{min}}{d_i^{max} - d_i^{min}}$, where the scaled attribute value d_i^{scaled} will take on values from the unit interval $[0, 1]$. d_i^{min} and d_i^{max} represent the minimal and the maximal values for a domain D_{a_i}. In the very first negotiation round, an agent uses $MaxDist(|A|)$ to replace the actual distance $dist(\overrightarrow{o}, \overrightarrow{o}_{opponent})$ in Eq.(1) if $o_{opponent}$ is unknown at that stage.

The distance between two offer vectors $dist(\overrightarrow{o}_x, \overrightarrow{o}_y)$ is defined by the weighted Euclidean distances [4]:

$$dist(\overrightarrow{o}_x, \overrightarrow{o}_y) = \sqrt{\sum_{i=1}^{|A|} w_i (d_i^x - d_i^y)^2} \quad (2)$$

where the weight factor $w_i = U_p^A(a_i)$ is an agent's valuation for a particular attribute $a_i \in A$. Figure 3 shows an example of an attribute domain "Quantity" and its attribute weight defined for a buyer agent via the client interface. An offer vector $\overrightarrow{o}_x$ contains an attribute value d_i^x along the ith dimension (issue) in a negotiation space. If an attribute interval instead of a single value is specified for an offer, the mid-point of an attribute interval is first computed.

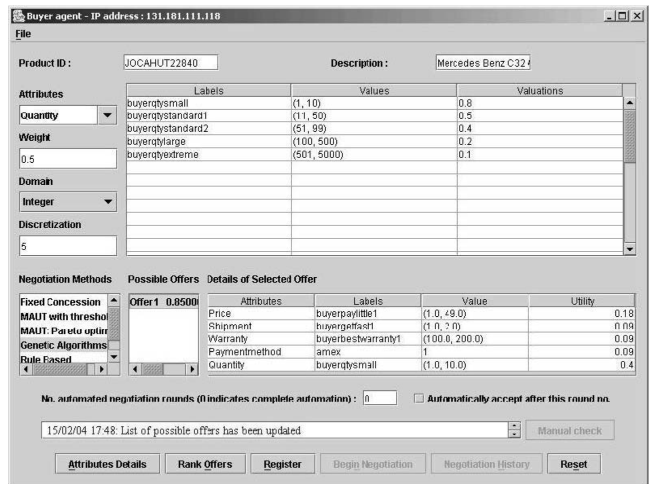

Figure 3. Specifying Agent Preferences via a Client Interface

For real-world negotiations, *time pressure* is often an important factor for concession generation. When the negotiation deadline is close, an agent is more likely to concede in order to make a deal. However, different agents may have different attitudes towards deadlines. An agent may be eager to reach a deal and so it will concede quickly (*Conceder agent*). On the other hand, an agent may not give ground easily during negotiation (*Boulware agent*) [28]. Therefore, a time pressure function TP is developed to approximate a wide spectrum of agent attitudes towards time. Our TP function is similar to the negotiation decision function referred to in the literature [6, 9].

$$TP(t) = 1 - (\frac{\min(t, t_p^d)}{t_p^d})^{\frac{1}{e_p}} \quad (3)$$

where $TP(t)$ denotes the time pressure given the time t represented by the absolute time or the number of negotiation rounds; t_p^d indicates the deadline for an agent p and it is either expressed as absolute time or the maximum number of rounds allowed. The term e_p denotes the agent p's eagerness factor towards negotiation. An agent p is *Boulware* if $0 < e_p < 1$ is set; for a conceder agent, $e_p > 1$ is true. If $e_p = 1$ is established, the agent holds *Linear* attitude towards the deadline. The values of the TP function are within the unit interval $[0, 1]$. The eagerness factor e_p can be chosen by the user or else a system default will be applied when a negotiation process begins.

3.3 The Genetic Algorithm

An agent's adaptive negotiation strategy is developed based on the following genetic algorithm:

CREATE the first population P^0 which consists of o_{max} and $N - 1$ individuals randomly selected from the set $O_p = D_{a1} \times D_{a2} \times \cdots \times D_{an}$;

WHILE NOT (Exit Criteria)
    $P^{i+1} = Best(P^i)$;
    $MP = Selection(P^i)$;
    DO UNTIL $Size(P^{i+1}) = N$
        $I_1 = Crossover(MP)$;
        $I_2 = Mutation(MP)$;
        $I_3 = Cloning(MP)$;
        $P^{i+1} = P^{i+1} \cup I_1 \cup I_2 \cup I_3$;
    END UNTIL

END WHILE

The initial population P^0 is created by incorporating the member o_{max} that maximises an agent p's payoff in the first round, and by randomly selecting the $N - 1$ members from the candidate set O_p, where N is the pre-defined population size. At the beginning of every evolution process, the fitness value of each chromosome is computed based on the most current negotiation parameters (e.g., an agent's utility function and the opponent's counter-offer). *Elitism* [10] is incorporated by executing the $Best$ function to copy the $e\%$ fittest chromosomes from the current generation P^i to the new generation P^{i+1}. By executing the Selection function, *Tournament selection* [24] is invoked to create a mating pool MP. For tournament selection, a group of k members are selected from a population to form a tournament. The member with the highest fitness among the selected k members is placed in the mating pool. This procedure is repeated n times until the mating pool is full.

Standard genetic operators: *cloning*, *crossover*, *mutation* are applied to the mating pool to create new members according to pre-defined probabilities. These operations continue until the new generation of size N is created. Cloning simply means copying a member from the mating pool to the new generation. Two-point crossover is used to exchange the fields (offer attribute values) between two parents to create two new members. In each two-point crossover operation, two points are randomly selected and the fields within the boundary defined by the two points are exchanged. An example of a basic crossover operation is depicted in Figure 2. Mutation involves randomly replacing some attribute values encoded on a chromosome by other attribute values from the permissible attribute domains of an agent. An example attribute domain for "Quantity" is depicted in Figure 3. Therefore, the proposed mutation operation will not generate offer values which are not acceptable to an agent. An evolution cycle is triggered after every x negotiation round(s), where x is the evolution frequency defined by the user. There is another parameter (number of evolutions) to define the number of evolutions to be performed in a evolution cycle. This parameter corresponds to the exit criteria defined in our genetic algorithm.

After an offer with concession is computed, the agent's decision for an incoming offer can also be developed. If the incoming counter-offer produces a payoff greater than or equal to that of the current proposal, a rational agent should accept the incoming offer; otherwise the incoming counter-offer should be rejected.

4 The Experiments

The negotiation environment of our experiments was characterized by multi-lateral negotiations among two buyer agents (B1, B2) and two seller agents (S1, S2) as depicted in Figure 1. These agents negotiated over some virtual services or products described by five attributes (i.e., $|A| = 5$) with each attribute domain containing 5 discrete values represented by the natural numbers $D_a = \{1, 2, 3, 4, 5\}$. The valuation of an attribute or a discrete attribute value fell in the interval of $(0, 1]$. Each negotiation case was defined in terms of the valuation functions U_p^A and $U_p^{D_a}$ for each agent p participating in the negotiation process. Each buyer (seller) participating in a negotiation process was assumed to have a product to buy (sell). A negotiation deadline of 200 rounds was established for each agent. For each negotiation case, an agreement zone always exists since the difference between a buyer and a seller only lies on their valuations against the same set of negotiation issues (e.g., attributes and attribute values). For each agent p, the size of the feasible offer set is: $|O_p| = 5^5 = 3125$.

A synchronised alternate-offering protocol was used in our experiments. At the beginning of every negotiation round, each agent would execute its genetic algorithm to generate an offer for that round. At the message exchange phase, each agent sent the offer messages to its opponents (e.g., $S1 \rightarrow B1$, $S1 \rightarrow B2$). After the message exchange phase, the simulation controller serialised the offer evaluation processes in the sequence of $< B1, B2, S1, S2 >$. Each agent selected the best incoming offer (evaluated according to its private utility function) as the opponent offer $o_{opponent}$ in a negotiation round. If there was a tie, an opponent would be selected randomly. If an agreement was made between a pair, they would be removed from the negotiation table, and the remaining two agents would continue their negotiation until either an agreement was made or the deadline was due.

Agent	(GA) Avg. Payoff	(GA) Avg. Time	(PO) Avg. Payoff	(PO) Avg. Time	Δ Payoff	Δ Time
B1	0.64	168.5	0.08	189	$+0.56$	-20.5
B2	0.54	169.3	0.09	199.9	$+0.45$	-39.6
S1	0.45	168.3	0.07	189	$+0.38$	-20.7
S2	0.41	168.8	0.05	199.9	$+0.36$	-31.1

Table 1. Comparative Negotiation Performance (GA) vs. (PO)

Ten negotiation cases were developed according to various valuation functions U_p^A and $U_p^{D_a}$ assigned to the four agents. The payoff obtained by each agent from every negotiation process was recorded. If no agreement was made after the deadline, the payoff obtained by an agent was zero. The average payoff obtained by each agent and the average negotiation time (in rounds) consumed over the ten negotiation cases are depicted under the (GA) columns in Table 1. The genetic parameters used by the (GA) system were: population size = 80, mating pool size = 50, size of solution set = 1, elitism factor = 10%, tournament size = 2, cloning rate = 0.1, crossover rate = 0.6, mutation rate = 0.05, Number of evolutions per cycle = 10, and evolution frequency = 1 (i.e., one evolution cycle per negotiation round). The negotiation trade-off factors $\alpha = 0.8$ and $\alpha = 0.5$ were applied to the buyer agents and the seller agents respectively. In addition, the eagerness factor $e_p = 1$ was employed by each agent.

The same set of simulated negotiations was conducted by the agents developed based on the basic negotiation mechanism (PO) defined in Section 2. The results of these agents are listed under the (PO) columns in Table 1. From Table 1, it is obvious that the GA-based negotiation agents out-perform the negotiation agents which guarantee Pareto optimal in terms of both average payoff (effectiveness) and average negotiation time (efficiency). The main reason is that the (PO) agents cannot find negotiation solutions (i.e., agreements) in most of the cases under a tough deadline of 200 negotiation rounds. On the contrary, even though the (GA) agents cannot guarantee Pareto optimal in general, they are sensitive to negotiation deadlines and are adaptive with respect to their opponents' negotiation behaviour. Therefore, a solution is always found in each of the ten cases. Figure 4 compares the average payoffs of the offers proposed by the B1 agents (in the (GA) and the (PO) systems) per 20 negotiation rounds in one typical negotiation case. It is not difficult to see that the concession making process of the B1 agent in the (GA) system is more sensitive to the deadline (e.g., a bigger drop of utility when the deadline is approaching) than that of the B1 agent in the (PO) system.

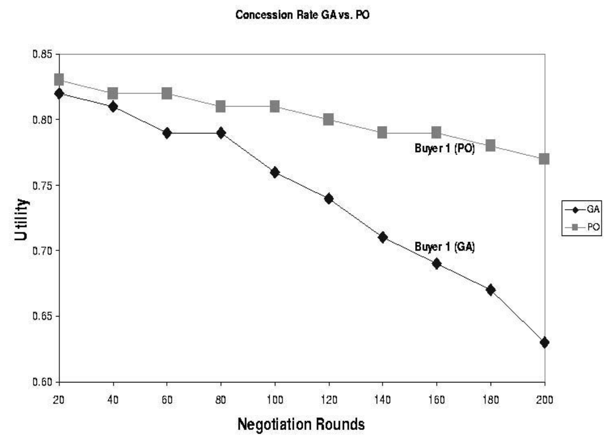

Figure 4. Comparative Concession Rate (GA) vs. (PO)

In order to examine the stability and the convergence of the proposed GA which underpins an agent's decision making process, the ten simulated negotiations were repeated five times. Figure 5 shows the average payoff of each agent in the (GA) system over five iterations. As can be seen, the average payoff obtained by an agent in each iteration is quite close. It indicates that the proposed GA can always converge under the same negotiation conditions. Moreover, the buyer agents obtained better payoffs than that of the seller agents since they put more weights on optimising self payoffs (e.g., a higher trade-off factor value). It confirms the findings of previous studies in that agents with stringent goals and starting with aggressive initial offers achieve better payoff in general [20].

The (PO) system requires a deadline of 1291 rounds such that each agent is able to find an agreement in every negotiation case. The average joint-payoff achieved by the (PO) system at the Pareto optimal point is depicted in Figure 6. To analyse the quality of the solutions generated by the (GA) system (e.g., how close the solutions to the optimal), the average joint-payoffs obtained by the society of agents in the (GA) system given various negotiation dead-

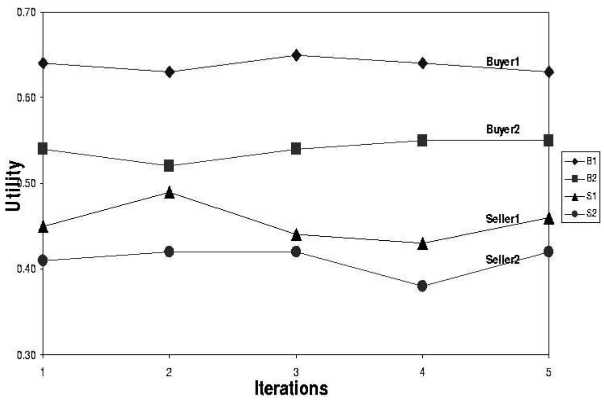

Figure 5. Agent Payoffs in the (GA) System Over 5 Iterations

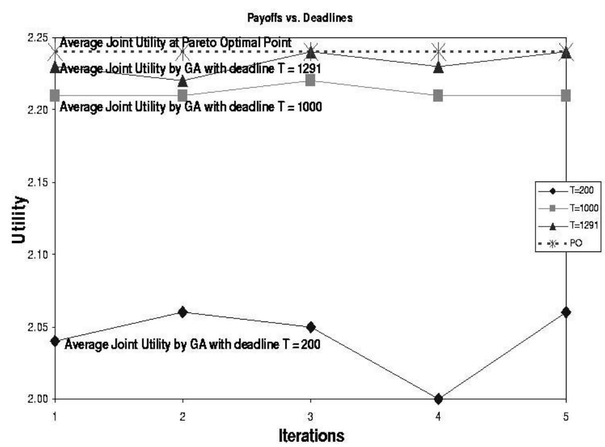

Figure 6. GA-based Agent Performance Under Various Deadlines

lines are plotted in Figure 6. The results plotted by the curves T=1291 and T=1000 were achieved when the parameters $e_p = 0.2$ and $\alpha = 0.95$ were applied to each agent in the (GA) system. As can be seen, when the negotiation deadline is set to 1291 rounds, the (GA) system can produce an average joint-payoff very close to that obtained at Pareto optimal point for each iteration. It indicates that the GA-based adaptive negotiation agents can really produce near-optimal negotiation results. In fact, even under great time pressure (e.g., deadline = 200 rounds), the GA-based negotiation agents can also find good solutions (in terms of average joint-payoff) because the agents' decision making models are responsive to negotiation deadline as well as the opponents' negotiation behaviour.

5 Conclusions and Future Work

Real-world negotiations are characterised by combinatorially complex negotiation spaces, tough negotiation deadlines, limited information about the opponents, and volatile negotiator preferences. Therefore, practical negotiation systems must be able to address these issues. The proposed GA-based adaptive negotiation mechanism fulfils most of the requirements of practical negotiation systems because it supports multi-party multi-issue negotiations based on a distributive decision making model which can deliberate negotiation solutions with incomplete and uncertain information. Furthermore, the proposed mechanism can mimic a wide spectrum of agent negotiation attitudes and generate near optimal solutions efficiently. Our initial experiments show that under time pressure, the GA-based adaptive negotiation mechanism outperforms a theoretically optimal negotiation model which guarantees Pareto optimal. Our research work opens the door to the development of practical negotiation mechanisms for real-world applications. Future work includes the enhancement of our existing genetic algorithm by exploring adaptive genetic operators and by taking into account the market-oriented factors such as competition and opportunity in multi-lateral negotiations. More empirical evaluation of the system will be performed based on real negotiation cases from various contexts.

References

[1] M. Barbuceanu and W.-K. Lo. Multi-attribute utility theoretic negotiation for electronic commerce. In F. Dignum and U. Cortés, editors, *Agent-Mediated Electronic Commerce III — Current Issues in Agent Based Electronic Commerce Systems*, number 2003 in Lecture Notes in Artifical Intelligence, pages 15–30. Springer-Verlag: Heidelberg, Germany, 2001.

[2] M. Bazerman, T. Magliozzi, and M. Neale. Integrative bargaining in a competitive market. *Organization Behavior and Human Decision Processes*, 35:294–313, 1985.

[3] S. P. M. Choi, J. Liu, and S.-P. Chan. A genetic agent-based negotiation system. *Computer Networks*, 37(2):195–204, Oct. 2001.

[4] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, New York, 1973.

[5] H. Ehtamo, E. Ketteunen, and R. Hamalainen. Searching for joint gains in multi-party negotiations. *European Journal of Operational Research*, 130:54–69, 2001.

[6] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.

[7] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. *Artificial Intelligence*, 142(2):205–237, 2002.

[8] S. Fatima, M. Wooldridge, and N. R. Jennings. Comparing Equilibria for Game-Theoretic and Evolutionary Bargaining Models. In *Proceedings of the International Workshop on Agent-Mediated Electronic Commerce V*, pages 70–77, Melbourne, Australia, 2003.

[9] S. Fatima, M. Wooldridge, and N. R. Jennings. An agenda based framework for multi-issues negotiation. *Artificial Intelligence*, 152(1):1–45, 2004.

[10] D. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, Reading, Massachusetts, 1989.

[11] J. Harsanyi and R. Selten. A generalised nash solution for two-person bargaining games with incomplete information. *Management Sciences*, 18(5):80–106, 1972.

[12] M. He, N. R. Jennings, and H. Leung. On agent-mediated electronic commerce. *IEEE Trans. on Knowledge and Data Engineering*, 15(4):985–1003, 2003.

[13] P. Heiskanen. Decentralized method for computing pareto solutions in multiparty negotiations. *European Journal of Operational Research*, 117:578–590, 1999.

[14] N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.

[15] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, methods and challenges. *Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

[16] R. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, Cambridge, UK, 1993.

[17] S. Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94(1–2):79–97, 1997.

[18] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.

[19] P. Krause, S. Ambler, M. Elvang-Goransson, and J. Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, 11:113–131, 1995.

[20] R. Krovi, A. Graesser, and W. Pracht. Agent behaviors in virtual negotiation environments. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(1):15–25, Feb. 1999.

[21] R. Lau. The State of the Art in Adaptive Information Agents. *International Journal on Artificial Intelligence Tools*, 11(1):19–61, Mar. 2002.

[22] R. Lau, B. Essam, S. Chan, and Z. Huang. Belief Revision for Adaptive Negotiation Agents. In *Proceedings of the 3rd IEEE/WIC International Conference on Intelligent Agent Technology*, pages 196–202, Halifax, Canada, Oct. 13–17 2003. IEEE Computer Society.

[23] A. R. Lomuscio and N. R. Jennings. A classification scheme for negotiation in electronic commerce. *Journal of Group Decision and Negotiation*, 12(1):31–56, 2003.

[24] N. Matos, C. Sierra, and N. R. Jennings. Determining successful negotiation strategies: an evolutionary approach. In Y. Demazeau, editor, *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS-98)*, pages 182–189, Paris, France, 1998. IEEE Press.

[25] S. Matwin, T. Szapiro, and K. Haigh. Genetic algorithm approach to a negotiation support system. *IEEE Transactions on Systems Man and Cybernetics*, 21(1):102–114, Jan.-Feb. 1991.

[26] J. F. Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.

[27] D. Pruitt. *Negotiation Behaviour*. Academic Press, 1981.

[28] H. Raiffa. *The Art and Science of Negotiation*. Harvard University Press, 1982.

[29] J. Rosenschein and G. Zlotkin. Task oriented domains. In *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*, pages 29–52. MIT Press, Cambridge, Massachusetts, 1994.

[30] B. Rubenstein-Montano and R. A. Malaga. A Weighted Sum Genetic Algorithm to Support Multiple-Party Multi-Objective Negotiations. *IEEE Transactions on Evolutionary Computation*, 6(4):366–377, Aug. 2002.

[31] A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1):97–109, 1982.

[32] C. Sierra, P. Faratin, and N. R. Jennings. Deliberative automated negotiators using fuzzy similarities. In *Proceedings of the EUSFLAT-ESTYLF Joint Conference on Fuzzy Logic*, pages 155–158, Palma de Mallorca, Spain, 1999.

[33] K. Sycara. Multi-agent compromise via negotiation. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence II*, pages 119–139. Morgan Kaufmann, 1989.

[34] J. von Neumann and O. Morgenstern. *The Theory of Games and Economic Behaviour*. Princeton University Press, 1994.

[35] D. Zeng and K. Sycara. Bayesian learning in negotiation. *International Journal of Human-Computer Studies*, 48(1):125–141, 1998.