An Evolutionary Learning Approach for Adaptive Negotiation Agents

Raymond Y.K. Lau, Maolin Tang, On Wong and Stephen W. Milliner
Centre for Information Technology Innovation
Faculty of Information Technology
Queensland University of Technology
GPO Box 2434, Brisbane, Qld 4001, Australia
E-mail: {r.lau, m.tang, o.wong, s.milliner }@qut.edu.au

Yi-Ping Phoebe Chen
School of Information Technology
Deakin University
221 Burwood Highway, Burwood, Victoria 3125, Australia
E-mail: phoebe@deakin.edu.au

Abstract. Developing effective and efficient negotiation mechanisms for real-world applications such as e-Business is challenging since negotiations in such a context are characterised by combinatorially complex negotiation spaces, tough deadlines, very limited information about the opponents, and volatile negotiator preferences. Accordingly, practical negotiation systems should be empowered by effective learning mechanisms to acquire dynamic domain knowledge from the possibly changing negotiation contexts. This paper illustrates our adaptive negotiation agents which are underpinned by robust evolutionary learning mechanisms to deal with complex and dynamic negotiation contexts. Our experimental results show that GA-based adaptive negotiation agents outperform a theoretically optimal negotiation mechanism which guarantees Pareto optimal. Our research work opens the door to the development of practical negotiation systems for real-world applications.

Keywords: Genetic Algorithms, Evolutionary Learning, Adaptive Negotiation Agents.

1. Introduction

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) [30]. Indeed, negotiation is one of the key stages with reference to the Business-to-Business Transaction (BBT) model [19]. If the negotiation spaces are large as found in many Business-to-Business (B2B) trading situations where dozens of issues (i.e., attributes) are involved, even experienced human negotiators will be overwhelmed. Under such circumstance, sub-optimal rather than optimal deals are often reached. Consequently the phenomenon of "leaving some money on the table" may occur [37]. Software agents are encapsulated computer systems situated in some environments such as the Internet and are capable of flexible, *autonomous* actions in that environment to meet their design objectives [21, 48]. The notion of agency can be applied to build robust architectures for automated negotiation systems within which a group of software agents communicate and autonomously make negotiation decisions on behalf of their human users. Recent research in intelligent agent mediated electronic commerce has highlighted the importance and the benefits of agent-based negotiation support for e-Business [17, 19]. These *negotiation agents* can considerably reduce human negotiation time and identify optimal or near optimal solutions from combinatorially complex negotiation spaces. This paper focuses on one of our negotiation models, a

Genetic Algorithm (GA) based adaptive negotiation agent model, implemented on our Web-based negotiation server.

1.1. Requirements of practical negotiation systems

Practical negotiation mechanisms must be computationally efficient. It implies that negotiation agents should be developed based on the assumption of *bounded* rather than *perfect rationality* [30]. These agents have only limited time and resources to deliberate negotiation solutions. As indicated by many authors that agents' knowledge about the negotiation spaces is very limited in business environments [11, 21]. An agent knows its own preferences (utility function) but normally does not know the preferences of its opponents because this information is often kept private by individual business. Accordingly, negotiation agents should be designed to deal with incomplete and uncertain information found in most negotiation scenarios. In particular, these agents should be empowered by effective and efficient learning mechanisms to gradually learn the opponents' interests based on their moves in the negotiation processes. Moreover, given the fact that the preferences of an agent and that of its opponents may change over time because of the ever changing negotiation environment, effective negotiation mechanisms must be able to deal with *dynamic* negotiation scenarios. For example, a negotiation agent should be able to observe the changing preferences of its opponents and adapt to the opponents' changing behaviour by initiating the corresponding actions (e.g., conceding more or less in subsequent rounds). To support various negotiation scenarios of real-world applications, 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, real-world negotiations such as negotiations in e-Business often involve many issues (e.g., price, quantity, product quality, shipment time, payment method, etc.), practical negotiation mechanisms must be able to handle multi-issue negotiations.

1.2. Justifications of the proposed approach

Negotiations in the real-world are often characterised by combinatorially complex negotiation spaces which involve many issues. In addition, negotiators are bounded by limited computational resources, time, and information about the negotiation spaces. Classical negotiation models based on operational research methods [9, 20] or traditional game-theoretic models [34, 47] need to be further developed to support negotiations in these realistic situations. GAs have long been taken as heuristic search methods to find optimal or near optimal solutions from large search spaces [16]. GAs can identify feasible solutions (e.g., mutually acceptable offers) efficiently without requiring detailed structural information about the search space as input. In the context of automated negotiations, our proposed GA is used to drive the heuristic search over a set of potential negotiation solutions (i.e., agreements). Given a negotiation situation, each negotiation agent will employ its own GA to conduct the distributed heuristic search in parallel. Therefore, our proposed GA-based negotiation method is more efficient than other negotiation methods based on exhaustive search. As a result, our negotiation approach is feasible for solving large and complex negotiation problem. In addition, the negotiation agents' searches for the mutually acceptable solution is not conducted in a totally independent manner. Instead, a co-evolution [7] is actually performed since the intermediate search result (i.e., a tentative solution) obtained by an agent will be passed to its negotiation opponents to stimulate their evolution processes. Thereby, a group of negotiation agents work collaboratively to solve complex negotiation problems.

It has been shown that a GA-based negotiation model can lead to optimal negotiation outcome as it is generated by a game-theoretic model under certain conditions [13]. In addition, 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. As a matter of fact, GAs have been successfully applied to develop automated negotiation systems [6, 27, 29] before. GA-based negotiation approaches fulfil the general requirements of developing practical negotiation systems in terms of computational efficiency, bounded agent rationality, and the assumption of limited information about the negotiation spaces. Therefore, a GA-based adaptive negotiation model is proposed to support negotiations for real-world applications.

1.3. Contributions of the paper

Although many negotiation models have been reported in the literature [18, 25, 35, 39, 50], these models have limited use in realistic negotiation environments because they are either assuming complete knowledge about the negotiation spaces, computationally inefficient, ignoring the time pressure, ineffective in learning changing negotiation contexts, or dealing with single issue only. This paper illustrates our GA-based adaptive negotiation agents which address all of the above issues. In particular, three main contributions of the work reported in this paper are:

1. Reviewing existing learning approaches for automated negotiations, particularly GAs-based adaptive negotiation mechanisms;

2. Designing and developing GA-based negotiation agents to support real-world applications such as negotiations for e-Business;

3. Quantitatively evaluating the performance of our GA-based adaptive negotiation agents.

1.4. Outline of the paper

The rest of the paper is organised as follows. Section 2 highlights the basic concepts with respect to a generic negotiation framework; Section 3 illustrates a basic negotiation model which guarantees Pareto optimal; Our GA-based adaptive negotiation mechanism is then illustrated in Section 4; Section 5 reports our experimental procedures and our empirical results;Section 6 compares our method with other related negotiation approaches; The final section highlights possible extensions of current work and concludes with a discussion of our GA-based adaptive negotiation agents.

2. A Generic Framework for Automated Negotiation

Given its ubiquity and importance in many different contexts, research into negotiation theories and techniques has attracted attention from multiple disciplines such as

Distributed Artificial Intelligence (DAI) [24, 26, 46], Social Psychology [3, 36, 37], Operational Research [9, 20], and Game Theory [34, 47]. 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 [14, 27, 30].

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). For example, English Auction is a well-known protocol for single issue (e.g., price) negotiation in an ascending open-cry environment.

Negotiation Strategies refer to the decision making apparatus that 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 [3, 37]. A *negotiation mechanism* refers to a particular negotiation protocol and the corresponding decision making model for the formulation of negotiation strategies.

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.

This paper focuses on our GA-based adaptive negotiation mechanism. In general, negotiators strive to increase their individual payoffs while ensuring that an agreement is feasible [27, 39]. In other words, negotiation agents have a common interest to cooperate, but have conflicting interests over exactly how to cooperate [14]. The main problem that confronts negotiation agents is to decide how much to concede in order to achieve the two conflicting goals of maximising its own payoff while ensuring an agreement to be made as soon as possible. In short, the most prominent issues that must be addressed in a negotiation mechanism are:

- How to represent negotiators' preferences and offers;
- How to compute concession and generate an offer;
- How to evaluate an incoming offer;
- How to learn the opponents' preferences.

Figure 1 depicts a simple negotiation process which involves a buyer and a seller. In this paper, it is assumed that only a finite set of agents P participates in a negotiation process. The negotiation process can be understood with reference to the simple

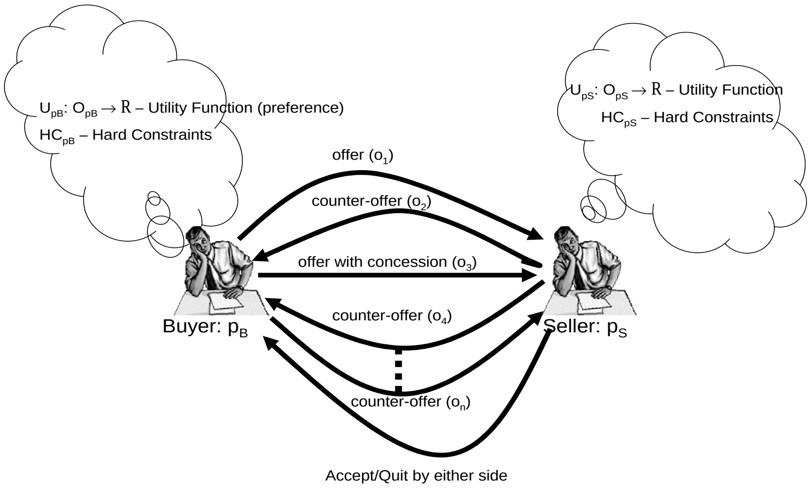


Figure 1. A Simple Negotiation Process

monotonic concession protocol [38]. 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 may make a concession or put forward the same offers again. If there is no agreement after the deadline is reached, an agent decides to quit and the negotiation ends with a conflict. With reference to Figure 1, the buyer agent makes an offer o_1 first. Such an offer is driven by her preferences (e.g., the utility function $U^o_{p_B}$). The seller agent $p_S \in P$ evaluates o_1 according to her own preferences $U^o_{p_S}$. If the offer o_1 produces a payoff (utility) higher than or equal to the seller's expected payoff at this round, the seller will accept the offer; otherwise a counter-offer o_2 is proposed by the seller. Similarly, the buyer agent $p_B \in P$ will evaluate o_2 according to her own preferences. If o_2 is not acceptable, the buyer agent may make a *concession* based on her original offer o_1 (e.g., raising the price) and generate a counter-offer o_3. This process continues until an offer and a counter-offer overlap, or either side decides to quit.

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 $U^o_p \in U$ 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: $U^o_p : D_{a_1} \times D_{a_2} \times \ldots \times D_{a_n} \mapsto \mathbb{R}$, where $\mathbb{R}$ is the set of real numbers. Each agent p has a deadline $t^d_p \in T$. 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 according to its own payoff function U^o_p.

3. A Basic Negotiation Model

The basic negotiation model illustrated in this section is based on multi-attribute utility theory (MAUT) [23] and is first discussed in [2]. It is a variant of the monotonic concession protocol with Zeuthen strategy [38, 51]. This simple model can guarantee Pareto optimal if an agreement zone exists in a negotiation space. Therefore, we use it as a baseline model to evaluate the performance of our GA-based negotiation model.

3.1. Representing offers

An *offer* $\overrightarrow{o} =< d_{a_1}, d_{a_2}, \ldots, d_{a_n} >$ 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 $D_{a_1} \times D_{a_2} \times \cdots \times D_{a_n}$. 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 $\overrightarrow{o}_i =< 20 - 30(K), 1 - 2(years), 10 - 30(days), 100 - 500(units) >$.

3.2. 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* [2, 22, 40]:

$$U_p^o(o) = \sum_{a \in A} U_p^A(a) \times U_p^{D_a}(d_a)$$

Non-linear utility function may also be used in ranking offers [6]. For example, if an agent p's valuations for the attributes are: $U_p^A(price) = 0.9$ and $U_p^A(quantity) = 0.1$, and its valuations for the attribute intervals are: $U_p^{D_a}(20 - 30K) = 0.8$ and $U_p^{D_a}(100 - 200units) = 0.5$, then an offer $\overrightarrow{o}_i =< 20 - 30(K), 100 - 200(units) >$ has utility 0.77 (i.e., $U_p^o(o_i) = 0.9 \times 0.8 + 0.1 \times 0.5 = 0.77$) from the agent p's perspective.

3.3. 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. In a negotiation round, a new offer with concession o_{new} can be determined based on the total order $o_x \preceq_p o_{new}$, where $o_x \preceq_p o_{new}$ denotes that the new offer o_{new} is more preferable than an arbitrary offer $o_x \in \{O_p - O_p^{'}\}$. The term $\{O_p - O_p^{'}\}$ represents the difference of the sets O_p (the set of feasible offers pertaining to an agent p) and $O_p^{'}$ (the set of offers proposed by an agent p before). The preference relation $\preceq_p$ is a total ordering induced by the utility function

U_p^o described in Section 3.2 over a set of offers. In other words, the set of feasible offers of an agent p are ranked in descending order of utility driven by $(\preceq_p, \{O_p - O_p^{'}\})$. A new offer with concession is picked up from the top of the ranking in each negotiation round.

3.4. 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 previously proposed offers $O_p^{'}$.

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 [2]. The advantage of such a basic negotiation model is that it does not require a pre-defined negotiation threshold nor the information about the opponents' utility functions. However, one of the problems of the basic negotiation model is that it may take a long time to sequentially evaluate all the candidate offers before a *Pareto optimal* solution is found. Given a high dimensional multi-issue negotiation space normally found real-world negotiation situations, this kind of blind sequential search may not be feasible. Another problem of the basic model is that it assumes the preferences of the agents remaining unchanged during a negotiation process, and therefore a Pareto optimal solution can be identified a priori. Nevertheless, agents' preferences (i.e., the utility functions) often change over time in real-world business environment. Therefore, the Pareto optimal solution identified by the model may not even be an acceptable solution because it is derived based on inaccurate and out-dated information.

4. GA-Based Adaptive Negotiation Agents

Development of our GA-based adaptive negotiation agents is driven by the basic intuition that negotiators tend to maximise their individual payoffs while ensuring that an agreement is reached [11, 27, 39]. This intuition is taken as the basis to develop the high-level negotiation strategy of our GA-based adaptive negotiation agents. The advantage of such an approach is that it does not rely on the detailed assumptions for a particular kind of negotiation environment, and so it is general enough to deal with a variety of negotiation situations. On the other hand, low-level negotiation strategy (also called tactics [31, 32]) is developed according to our proposed genetic algorithm. The low-level negotiation strategy of an agent determines which offer to be made in each negotiation round, and such a strategy is adaptive with respect to the opponents' possibly changing negotiation behaviour. To make an agent's low-level negotiation strategy adaptive, an effective learning mechanism, which can take into account the agent's own payoff as well as the opponents' payoffs (to increase the chance of making an agreement), is required. The learning mechanisms of our negotiation agents are underpinned by a genetic algorithm. The proposed GA ensures that negotiation agents conducted their heuristic searches (i.e., the negotiation processes) in an effective and efficient manner.

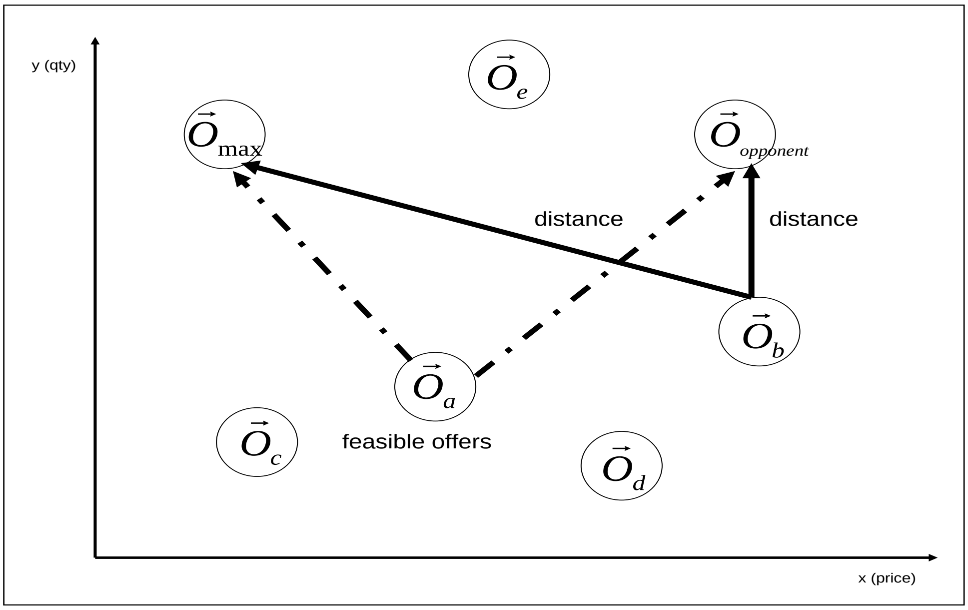


Figure 2. A Geometric Negotiation Space

To estimate the opponents' payoffs, an agent needs to *learn* the preferences of its opponents through their previous encounters. There are two possible approaches as reported in the literature; one is to learn the opponent's utility function directly [6, 50], and the second approach is to estimate an opponent's preferences indirectly according to its previous counter-offers [11, 52]. We adopt the latter approach since it is simpler and computationally tractable. If the first approach is adopted, it is quite difficult, if not totally impossible, to assign prior probabilities to the set of possible outcomes given that we are dealing with practical negotiation problems involving many negotiation issues [11]. On the other hand, an offer can be considered as a whole or the individual issues of an offer can be considered separately when a concession is computed [14]. Our proposed concession generation mechanism considers an offer as a whole since we believe that this is a more efficient approach than conceding on an issue by issue basis.

Essentially, each agent is empowered by the proposed GA to evaluate the set of feasible offers pertaining to its negotiation space. The fittest member according to the fitness function applied in the GA is chosen from a population (i.e., a subset of the set of feasible offers O_p) to form a negotiation solution in a negotiation round. According to the basic intuition, an offer is considered fit if it tends to generate maximal payoff (i.e., close to the maximal offer) and it is also similar to the opponent's recent counter-offer. Evaluation of offers can be analysed with respect to a geometric negotiation space (Figure 2) pertaining to an agent. In Figure 2, offers $o_a \ldots o_e$ are the subset of feasible offers under consideration. 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. All these offers are represented by the corresponding offer vectors in the geometric negotiation space. The distance among the vectors can then be measured based on standard distance function such as the weighted Euclidean distance [8]. In general, the fitter offers of a population are those having minimal distances to both $\overrightarrow{o}_{max}$ and $\overrightarrow{o}_{opponent}$. In each negotiation round, the offer vector $\overrightarrow{o}_{opponent}$ may change, and so are the offers considered fit by the agent. In other words, an agent is learning the opponent's preferences gradually. According to the evolution principle, the population of feasible offers is dominated by fit members gradually. Therefore, an agent's proposal is moved closer to its opponent in each negotiation round. Finally, the conflict between an agent and its opponent is resolved and an agreement is reached. Another advantage of the GA-based negotiation model is that it does not assume that the preferences of the agents remain unchanged. In fact, the preferences of the agents may change in each negotiation round and these changes are reflected by the new offer vectors $\overrightarrow{o}_{max}$ and $\overrightarrow{o}_{opponent}$. The set of feasible offers are then evaluated with respect to these new reference vectors (i.e., the agents' new preferences).

4.1. Offer encoding

The proposed GA-based negotiation model utilises a population of chromosomes to represent a set of feasible offers $O_p^{feas} \subseteq O_p$ for an agent 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 3 depicts the decimal encoding (Genotype) of some chromosomes and a crossover operation. The genetic operators such as crossover and mutation are only applied to the genes representing the attribute values of an offer.

4.2. 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 E-business applications. Therefore, the proposed fitness function approximates the ideal function and captures two important issues, an agent's own payoff and the opponent's *partial preference* (e.g., the most recent counter offer). With reference to the geometric negotiation space depicted in Figure 2, the fitness of a chromosome (i.e., an offer o) is defined by:

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

where o_{max} represents an offer which produces the maximal payoff based on an agent's current utility function; $\overrightarrow{o}_{opponent}$ is the offer vector representing the most recent counter-

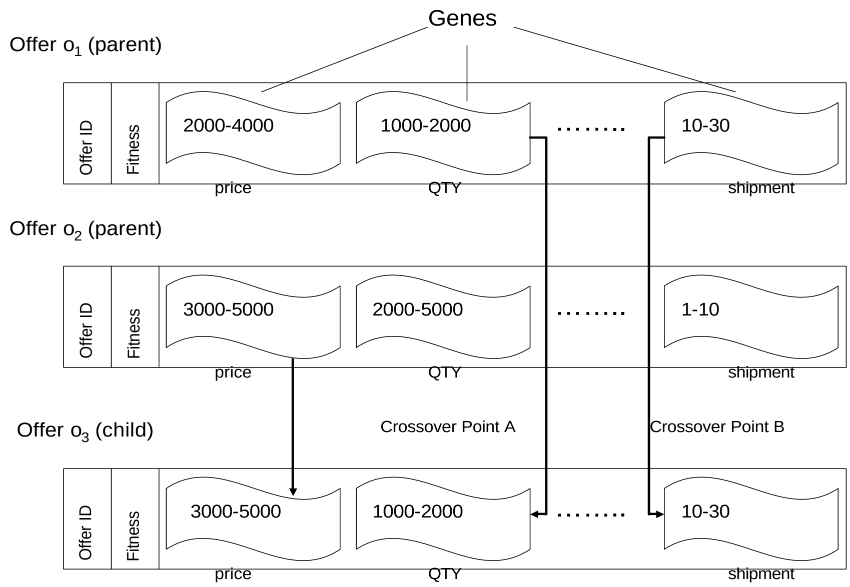


Figure 3. Encoding Candidate Offers

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 normalised in the unit interval. In the very first negotiation round, the agents will use $MaxDist(|A|)$ to replace the actual distance $dist(\overrightarrow{o}, \overrightarrow{o}_{opponent})$ in Eq.(1) since $o_{opponent}$ is unknown at this stage. This is a conservative estimation and assuming that the agents' interests are in total conflict at the beginning. In other words, an offer is evaluated based on an agent's own utility function initially. In Eq.(1), we simply use the ratio of the utility generated by a potential offer o to the maximal utility instead of computing their Mahalanobis distance because it is easier to compute the ratio given the utility function of an agent. Nevertheless, it is impossible to compute such a ratio for the opponent because the utility function of the opponent is unknown.

The distance between two offer vectors $dist(\overrightarrow{o}_x, \overrightarrow{o}_y)$ is defined according to the weighted Euclidean distance [8]:

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

where the weight factor $w_i = U_p^A(a_i)$ is an agent's valuation for a particular attribute $a_i \in A$. 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. The mid-point value is then scaled to the unit interval $[0, 1]$ by linear scaling:

$$d_i^{\text{scaled}} = \frac{d_i - d_i^{min}}{d_i^{max} - d_i^{min}} \quad (3)$$

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}.

For practical negotiations arising in business contexts, *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*) [37]. 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 [10, 14, 13].

$$TP(t) = 1 - (\frac{\min(t, t_p^d)}{t_p^d})^{\frac{1}{e_p}} \quad (4)$$

$TP(t)$ 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 in negotiation [41, 42, 43, 45]. 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 $[0, 1]$. When $t = 0$ is the input, the function returns 1. When the deadline is due ($t = t_p^d$), the time pressure function returns 0. The time pressure factor is incorporated into our GA-based negotiation model by the enhanced fitness function:

$$fitness(o) = \alpha \times TP(t) \times \frac{U_p^o(o)}{U_p^o(o_{max})} + (1 - \alpha \times TP(t)) \times (1 - \frac{dist(\overrightarrow{o}, \overrightarrow{o}_{opponent})}{MaxDist(|A|)}) \quad (5)$$

The eagerness factor e_p can be chosen by the user or else a system default is assumed before a negotiation process begins. The time pressure function is applicable for $\alpha > 0$. For the extreme case ($\alpha = 0$), an agent is totally benevolent. Under such circumstance, there is no need to consider time pressure because the agent will give ground completely at the beginning of a negotiation session.

To consider the opponent's concession matching behaviour [12, 27] and to make our negotiation system more robust, the eagerness factor [41, 42, 43, 45] can be adjusted by our system automatically. The adjustment of the eagerness factor depends on the opponent's concession matching behaviour. To support this dynamic concession matching behaviour, an agent needs to maintain a separate population of chromosomes to adapt to the concession behaviour of each of its opponent. In general, our GA-based adaptive negotiation agent tries to mimic its opponent's concession matching behaviour. If the opponent did not concede in the past few rounds defined by a time window λ, an agent would become Boulware too. On the other hand, if the opponent conceded, the agent would become cooperative and start to concede too. The opponent's concession rate $CR_{opp}(\lambda)$ is measured by:

$$CR_{opp}(\lambda) = \frac{U_p^o(o_{opponent}^{t-1}) - Avg_{i=2}^{\lambda+1} U_p^o(o_{opponent}^{t-i})}{U_p^o(o_{opponent}^{t-1})} \tag{6}$$

where $U_p^o(o_{opponent}^{t-1})$ denotes the payoff of the previous offer ($t-1$ step ago) received from the opponent and it is evaluated using agent p's utility function. $Avg_{i=2}^{\lambda+1} U_p^o(o_{opponent}^{t-i})$ represents the average payoff of the past λ offers received from the opponent. If the opponent did not concede at all, $CR_{opp}(\lambda)$ is less than or equal zero; otherwise it is greater than zero. If $CR_{opp}(\lambda) = 0$ is true, $e_p = e_p^{MinBoulware}$ will be set. For our current implementation, $e_p^{MinBoulware} = 0.9$. The maximum e_p^{max} and the minimum e_p^{min} eagerness factors are defined to be 50 and 0.02 respectively. If $CR_{opp}(\lambda)$ is less than zero (i.e., a very tough opponent), $e_p = e_p^{min}$ will be set. If $CR_{opp}(\lambda)$ is greater than zero (i.e., the opponent has conceded), our system adjusts the agent p's eagerness factor [41, 42, 43, 45] e_p by:

$$e_p = \begin{cases} (CR_{opp}(\lambda) + 1) \times Adj_{conceder} & \text{If } e_p \leq e_p^{max} \\ e_p^{max} & \text{otherwise} \end{cases} \tag{7}$$

$Adj_{conceder}$ is the adjustment factor to convert the positive $CR_{opp}(\lambda)$ values to the eagerness values.

4.3. The Genetic Algorithm

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

LET $i = 0$

CREATE the first population P^i 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 $(i \leq MaxE)$

$P^{i+1} = Best(P^i)$;
$MP = Selection(P^i)$;
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
$i = i + 1$

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 each evolution cycle, 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* 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, either *Tournament selection* [32] or Roulette-wheel selection can be used 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. Roulette-wheel selection is analogous to a roulette wheel where the probability that a member is chosen is proportional to its fitness. The roulette-wheel selection function works on the ranks of chromosomes rather than evaluating the fitness values of the chromosomes directly. The reason is that a ranking reflects the relative performance of the chromosomes in a population and hence minimises the effect of large disparities in the fitness values. A ranking pertaining to a population of chromosomes is generated in descending order of the fitness values. For each chromosome o, the probability of selection is computed according to:

$$Pr(o_{selected}) = 1 - \frac{1}{o_{rank}} \times (o_{rank} - 1) \tag{8}$$

where o_{rank} is the rank of a particular chromosome (i.e., a potential offer). Then, a random number in the unit interval is generated for each chromosome. If $Pr(o_{selected})$ of a chromosome is greater than or equal to the random number, the corresponding chromosome is selected and put into the mating pool. This process continues 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. Two point crossover is used to exchange the fields (offer values) between two parents to create two new members. An example of a crossover operation is depicted in Figure 3. Mutation involves randomly replacing some attribute values encoded on a chromosome by other attribute values from the corresponding attribute domains (quantity). Figure 4 shows an example of an attribute domain defined for a buyer agent via the client interface. Therefore, the proposed mutation operation will not generate offer values which are not acceptable to an agent. An evolution process is invoked after every x negotiation round, where x is the evolution frequency defined by the user. The lower the value of x, the more GA-based learning takes place in a negotiation agent. For example, if $x = 1$ is true, an agent will apply the GA to find a tentative offer in each negotiation round. In general, a low value of x leads to the reduction of negotiation rounds to reach an agreement (if a solution does exist) because an agent will learn the opponent's preferences more frequently. There is another parameter ($MaxE$) to define the maximum number of evolutions to be performed in a evolution cycle. An example of genetic parameters is depicted in Figure 5.

As the proposed adaptive negotiation mechanism allows an agent to change its valuation functions (i.e., preferences) during a negotiation process, it is possible for an agent to propose the same offer twice in different negotiation rounds. To avoid an endless loop, a maximum number of negotiation rounds (i.e., a deadline) is specified by the user. If the deadline is due or an agreement is reached, the negotiation process will be terminated. Basically an agent's concession generation mechanism is underpinned by the above genetic algorithm because the GA determines which offer to be proposed in the current round. After an offer is determined by the GA, the agent's decision for an incoming counter-offer can also be developed easily. If the incoming counter-offer produces a payoff greater than

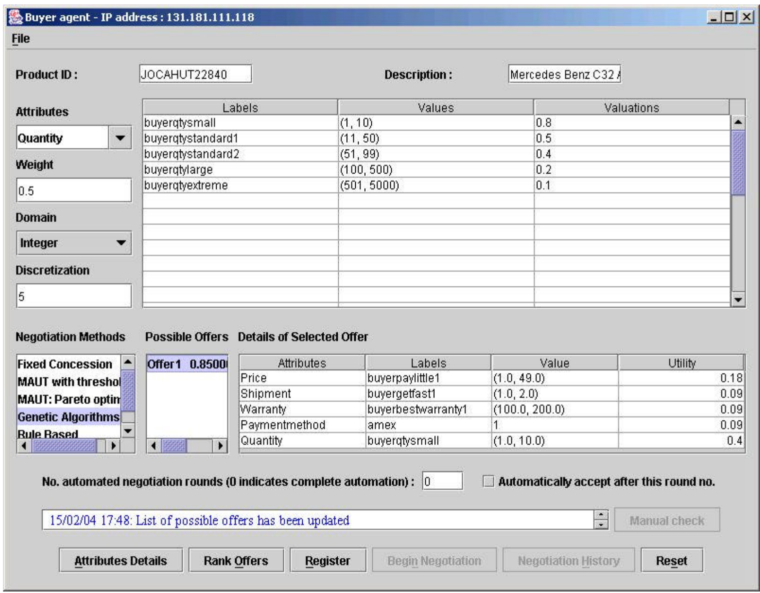


Figure 4. A Buyer Client Interface

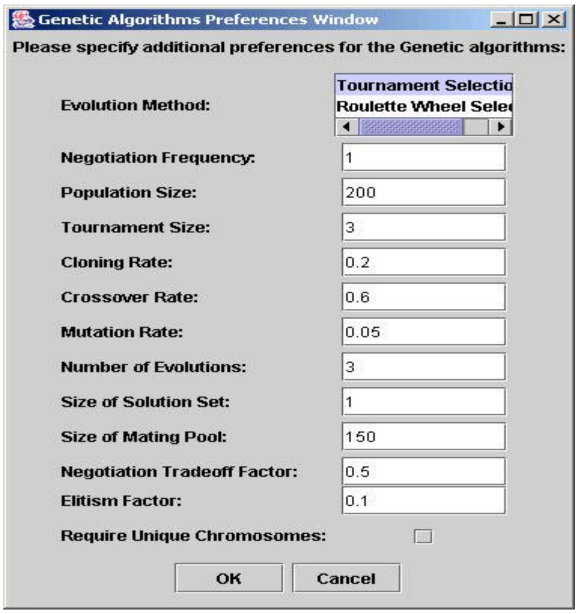


Figure 5. An Example of Genetic Parameters

or equal to that of the current proposal, a rational agent should accept the incoming offer; otherwise the incoming counter-offer should be rejected. Our negotiation agents do not explicitly adjust the *aspiration level* as proposed by [44]. In this context, the aspiration level means the gap (i.e., the difference of two utilities) between the current proposal and the incoming offer [44]. Under tight constraints such as short deadline and tough opponent, our proposed concession mechanism (e.g., the fitness function) will automatically lower the agent's expectation and consider a moderate offer (in terms of perceived payoff) as the current solution. In other words, the aspiration level is lowered.

5. The Experiments

The negotiation spaces Neg of our experiments were characterized by bilateral negotiations between a buyer agent p_B and a seller agent p_S. Each negotiation profile consists of 5 attributes with each attribute domain containing 5 discrete values represented by the natural numbers $D_a = \{1, 2, \ldots, 5\}$. The valuation of an attribute or a discrete attribute value was in the interval of $(0, 1]$. 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, the size of the candidate offer set O_p is $3,125$. 5 negotiation groups with each group containing 10 cases were constructed. For each negotiation case in a group, the negotiation profile (e.g., the preferences) of a buyer was randomly created according to the configuration mentioned before (i.e., 5 attributes with each attribute containing a value $d_{a_i} \in D_{a_i}$). A buyer's profile was then copied to create a seller's profile, and then the seller's profile was modified with various levels of preferential difference introduced (e.g., modifying the valuations of some attribute values).

For the first simulation group, each negotiation case contained identical buyer/seller preferences (i.e., the same weights for the attributes and the same valuations against the same set of attribute values). This group was used as a control group and the other groups were the experimental groups. Each negotiation case in the second group contained 20% preferential difference between a buyer and a seller (e.g., one valuation of an attribute value is different). Each case in the succeeding group was injected a 20% increment of preferential difference. The genetic parameters were: population size = 200, mating pool size = 150, size of solution set = 1, elitism factor = 10%, tournament size = 3, cloning rate = 0.2, crossover rate = 0.6, mutation rate = 0.05, Number of evolutions per cycle = 3, and evolution frequency = 1 (i.e., one evolution cycle per negotiation round). Tournament selection was used for the experiments reported in this paper. The negotiation trade-off factor $\alpha = 0.5$ and the negotiation deadline = 100 (rounds) were set for both the buyer and the seller. One hundred rounds were seen as a moderate deadline given the large negotiation space. For each simulation run, an agent (either the buyer or the seller) was randomly chosen to initiate the negotiation process. The basic performance measures were joint-payoff (JP), success rate (SR), fairness ratio (FR), and negotiation round (RD). Joint-payoff is the sum of the utilities generated from the agreement as computed according to the buyer and the seller's utility functions respectively. If an agreement was not reached after the deadline was due, zero utility was assumed for each agent. Success rate is the percentage of cases with agreements made over the the number of negotiation cases. The fairness ratio represents the degree of equity in the negotiation and is computed by taking the ratio of the seller's individual payoff to the buyer's individual payoff [27].

Experiment 1

The purpose of the first experiment was to study the general performance of the GA-based adaptive negotiation agents when compared with the baseline negotiation model (Section 3) which guarantees Pareto optimal. The main hypothesis was that the performance of the GA-based negotiation agents should be better than that of the baseline negotiation system under realistic negotiation conditions (e.g., limited negotiation time). The first simulation run involved negotiation agents developed according to the baseline negotiation model. The second simulation run involved GA-based adaptive negotiation agents. For the GA-based negotiation agents, both the buyer agent and the seller agent were Boulware agents with eagerness factor $e_p = 0.5$. In other words, the interactions of these agents are symmetry. In this experiment, the fitness function Eq.(5) was used. However, the eagerness factor $e_p = 0.5$ remained the same throughout the experiment. The concession matching mechanism Eq.(6) was not activated in the GA-based negotiation agents. In other words, the agents were adaptive to each other's moves and the negotiation deadline, but they were not responsive to the opponent's concession matching behaviour. All the simulation runs (each case in the 5 negotiation groups) were performed on our negotiation server with the configuration of a single Pentium III 800 MHz CPU and 256MB main memory. Both the GA-based negotiation agents and the baseline negotiation system dealt with exactly the same set of negotiation cases. The following measures were used to evaluate the relative performance of the GA-based adaptive negotiation agents against the baseline negotiation system:.

$$\Delta_{\text{utility}} = \frac{JP_{GA} - JP_{baseline}}{JP_{baseline}} \times 100\%$$

$$\Delta_{\text{rate}} = \frac{SR_{GA} - SR_{baseline}}{SR_{baseline}} \times 100\%$$

$$\Delta_{\text{round}} = \frac{RD_{GA} - RD_{baseline}}{RD_{baseline}} \times 100\%$$

For the baseline negotiation system, the joint-utility ($JP_{baseline}$) represents the sum of the buyer's payoff and the seller's payoff obtained at Pareto optimal point. However, it should be noted that the maximal joint utility of a negotiation session may not necessarily obtained at the Pareto optimal point. $SR_{baseline}$ is the average success rate achieved by the baseline negotiation mechanism depicted in Section 3. If the agents can reach an agreement before the deadline (i.e., 100 negotiation rounds), the negotiation session is considered successful. $RD_{baseline}$ refers to the average number of negotiation rounds consumed by the baseline negotiation mechanism. Table I summarizes the average Δ_{utility}, Δ_{rate}, and Δ_{round} for each negotiation group. A positive Δ_{utility} indicates that the GA-based negotiation agents are more effective than the baseline negotiation system, and a positive Δ_{rate} shows that the GA-based negotiation agents are more efficient because they can finish negotiations on or before deadline. However, a negative Δ_{round} indicates that the GA-based adaptive negotiation agents can finish the negotiation processes in fewer number of negotiation rounds. If an agreement can be reached, the actual number of negotiation rounds consumed by the agents is recorded. If an agreement cannot be reached before the deadline, 100 rounds will be consumed by that particular negotiation session.

An overall results of $\Delta_{\text{utility}} = 10.3\%$, $\Delta_{\text{rate}} = 15.8\%$ and $\Delta_{\text{round}} = -18.2\%$ were obtained and the original hypothesis was confirmed. On average the baseline model consumed 96.5 negotiation rounds ($RD_{baseline} = 96.5$) to deal with the various negotiation sessions, and the GA-based negotiation agents took 78.9 negotiation rounds ($RD_{GA} = 78.9$) to complete the negotiation sessions. The proposed negotiation method ensures that "as little money is left on the table as possible". Since the proposed GA-based negotiation agents can observe their opponents' preferences and continuously learn this information via the opponents' counter-offers, the search for a mutually acceptable offer becomes faster. Moreover, as the GA-based negotiation agents were responsive to time pressure, most of the agreements could be reached on or before the deadline. On the other hand, the basic negotiation system failed to develop some negotiation solutions given limited negotiation time. Although the basic negotiation system can guarantee Pareto optimal solutions, it is less useful for practical negotiations because it is not responsive to the changing negotiation contexts. For the control negotiation group (group 1), both systems could successfully identify all the negotiation solutions given exactly the same preferences of the negotiators. In fact, both systems could identify the solution in the first round for each case in the first negotiation group. In general, the performance gap between these two negotiation systems becomes larger if the preferential differences between the buyer and the seller are bigger. The reason is that the GA-based negotiation agents are equipped with effective learning mechanisms to learn and adapt to the opponents' preferences even though the preferential differences between the parties are big. Therefore, GA-based negotiation agents perform better in these more realistic and more challenging negotiation sessions. The improvement in terms of the average number of negotiation rounds to complete a negotiation session achieved by the GA-based adaptive negotiation agents indicates that these agents are more efficient than those agents developed based on the basic negotiation model. Except for the first negotiation group where both kinds of agents complete every negotiation session after the first round, the GA-based agents generally complete a negotiation session faster when compared to their non-adaptive counterparts.

Table I. Comparative negotiation performance GA vs. Baseline

Group	Preferential Difference	Δ_{utility}	Δ_{rate}	Δ_{round}
1	0%	0.0%	0.0%	0.0%
2	20%	−1.6%	0.0%	−36.5%
3	40%	9.8%	11.1%	−31.8%
4	60%	17.2%	25.0%	−15.5%
5	80%	26.1%	42.8%	−7.1%
	Average	10.3%	15.8%	−18.2%

Experiment 2

The purpose of the second experiment was to examine the capabilities of the GA-based adaptive negotiation agents in dealing with dynamic negotiation environments (e.g., the preferences of an agent and its opponent will change during a negotiation process). The main hypothesis was that the GA-based negotiation agents would be responsive to the changing negotiation preferences, and so should demonstrate a bigger performance boost when compared with the baseline negotiation system which cannot take into account any changing negotiation preferences. The same set of negotiation cases and the same agent parameters used in experiment one were adopted in this experiment. The fitness function Eq.(5) was still used in the GA-based negotiation agents. The only difference between experiment 1 and experiment 2 was that a random preferential change (e.g., the valuation of an attribute or attribute value) was injected to both the buyer and the seller at the end of every five negotiation rounds if an agreement had not been established in that round. Similar to experiment 1, the interactions between the buyer and the seller were symmetry since both of them demonstrated changing negotiation preferences with the same frequency. With reference to our client interface (Figure 4), a breakpoint was set to 5 (i.e., a negotiation process continues for 5 rounds and then stops) and then the valuation values were modified. Both the GA-based negotiation agents and the baseline system were exposed to the same set of negotiation cases with preferential changes. For the baseline model, it cannot take into account any changing preferences, and so the negotiation processes were not affected. However, when the payoff of a buyer or a seller was computed at the end of a negotiation process, the most up-to-date utility function (due to the changing preferences of a negotiator) was used to compute an agent's payoff.

Table II summarizes the average Δ_{utility} and Δ_{rate} for each negotiation group. An overall results of $\Delta_{\text{utility}} = 31.6\%$ and $\Delta_{\text{rate}} = 15.8\%$ were obtained and our second hypothesis was confirmed. Apart from the control group (group 1), the performance of the GA-based adaptive negotiation agents was much better than that of the baseline system. For the control group, both systems could successfully finish all the negotiations in round one before any preferential changes were introduced. Therefore, the results of group one was the same as that obtained in experiment one. It was obvious that the baseline system could not find the right solutions for most of the cases of the remaining groups. On the other hand, the GA-based negotiation agents were still performing well because they could adapt to the preferential changes and were able to reach agreements before the deadline for all the cases. Overall, this experiment shows that the GA-based adaptive negotiation mechanism is quite effective under dynamic negotiation environment.

Table II. Comparative Negotiation Performance in Dynamic Environment

Group	Preferential Difference	Δ_{utility}	Δ_{rate}
1	0%	0.0%	0.0%
2	20%	21.6%	0.0%
3	40%	32.8%	11.1%
4	60%	42.3%	25.0%
5	80%	61.5%	42.8%
	Average	31.6%	15.8%

Experiment 3

The purpose of the third experiment was to examine the concession matching mechanisms of the GA-based negotiation agents. This experiment only involved the GA-based adaptive negotiation agents. The control group was the adaptive agents without the concession matching mechanisms Eq.(6) activated, while the experimental group consisted of agents with the concession matching mechanisms activated. The parameter $\lambda = 3$ was set when the agents used the concession matching tactics. Ten negotiation cases from group 4 which were used in experiment one and two were adopted in this experiment. To simulate different encounters such as Boulware-to-Boulware, Boulware-to-Conceder, Conceder-to-

Boulware, and Conceder-to-Conceder, the buyer agent and the seller agent took the respective roles in turn. As a whole, it was a $9 \times 2 \times 10$ factorial design which involved 180 simulation runs since there were 9 types of encounters, 2 kinds of agents (concession matching mechanisms activated or not), and 10 negotiation cases. For a Boulware agent, the eagerness factor [41, 42, 43, 45] $e_p = 0.5$ was set. For a Conceder agent, the eagerness factor was $e_p = 5$ and the Linear agent had $e_p = 1$. In this experiment, the average *fairness ratio* was used to evaluate the performance of our concession matching mechanism. The fairness ratio represents the degree of equity in the negotiation and is computed by taking the ratio of the seller's individual payoff to the buyer's individual payoff [27]. All the genetic parameters and agent parameters were the same as the previous experiments except for those we mentioned above. Our hypothesis in this experiment was that the negotiation agents with the concession matching mechanisms activated should produce more robust results, that is, the fairness ratios should be higher in all kinds of encounters. The relative performance of the agents with concession matching tactics was measured using the following metric:

$$\Delta_{\text{Fair}} = \frac{FR_{CMT} - FR_{NCMT}}{FR_{NCMT}} \times 100\%$$

where FR_{CMT} denotes the average Fairness Ratio for agents with Concession Matching Tactics employed, and FR_{NCMT} represents the average Fairness Ratio for agents without Concession Matching Tactics employed.

Table III depicts the average Δ_{Fair} for the 9 encounters. A positive Δ_{Fair} indicates that a fairer result was achieved. As can be seen, a better result (in terms of a higher fairness ratio) was produced when our concession matching mechanisms were applied to negotiations, and so our third hypothesis was confirmed. The reason was that the agents tried to mimic and adapt to each other during negotiation. Consequently, a fairer outcome was produced. For example, when a Boulware agent negotiated with a Conceder agent, the Boulware agent learned to be less Boulware. Meanwhile, the Conceder agent learned to be tougher (i.e., less conceding). Therefore, their concession making behaviour converged gradually and the resources were more evenly distributed. If both agents were of the same type, their positions remained the same, but the tougher Boulware became less Boulware while the tender Boulware became tougher. As a result, their concession behaviour also converged.

Table III. The Effect of Learning Concession Matching Behaviour

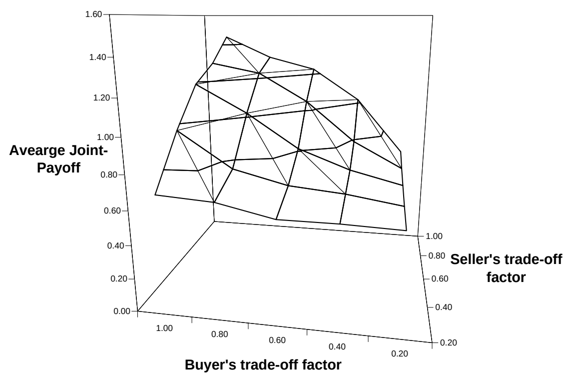
Agent Trait	Boulware	Conceder	Linear
Boulware	1.3%	9.5%	3.3%
Conceder	10.4%	0.9%	1.5%
Linear	3.1%	1.8%	0.6%

Experiment 4

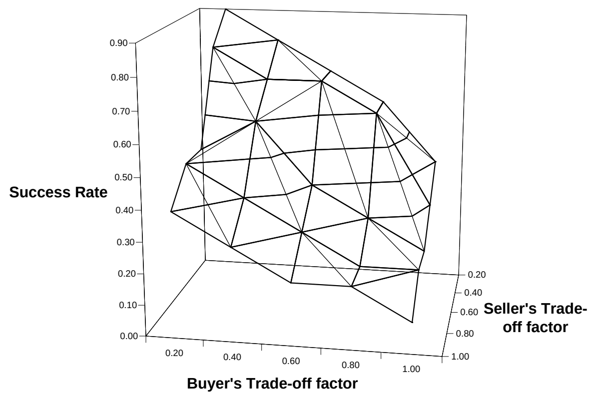
The purpose of the fourth experiment was to examine the interactions among different kinds of agent attitudes (e.g., self-interested to benevolent). In addition, we would like to evaluate if our adaptive negotiation mechanism Eq.(5) could improve the negotiation outcome under extreme agent attitudes. Our hypothesis was that the adaptive negotiation mechanism which could take into account approaching deadline would lead to higher success rate in extreme encounters such as fully self-interested attitude to fully self-interested attitude. From the literature, it has been indicated that fully self-interested agents do not necessarily achieve good negotiation outcomes (e.g., the Prisoner's Dilemma) [37, 47]. Through our simulated negotiations, it is interesting to see if certain agent attitudes would generally lead to better outcome under realistic conditions (e.g., limited information, time pressure). Ten negotiation cases in group 4 which were used in experiment one and two were employed in this experiment. For the first series of simulation runs, the fitness function Eq.(1) was employed so that the agents' attitudes were fixed in each negotiation session. For the second series of simulation runs, the fitness function Eq.(5) was employed so that the agents' attitudes were adjusted with respect to the time pressure. Basically, it was a $5 \times 5 \times 2 \times 10$ factorial design. There were 5 buyer attitudes ($\alpha = 0.2, \ldots, \alpha = 1.0$), 5 seller attitudes, and for each interaction, there were 10 negotiation cases. This experimental setting was repeated twice, one for non-adaptive agents, and the another for agents that were responsive to time pressure.

Joint-payoff (JP) and success rate (SR) were used to measure the agents' performance with respect to various agent attitudes. For this experiment, if an agreement could not be reached before the deadline (100 rounds), the case was ignored in computing the average joint-payoff since we would like to clearly examine if a particular interaction of agent attitudes promoted joint-payoff. The genetic parameters were the same as those used in the previous experiments, but the trade-off factor varied in the range of $[0.2, 1]$. For each pair of agent attitude such as $\alpha_{buyer} = 0.2$ and $\alpha_{seller} = 0.2$, the 10 negotiation cases were executed to obtain the average joint-payoff and the average success rate within 100 negotiation rounds. Figure 6.a and Figure 6.b show the average joint-payoff and the average success rate for various interactions of agent attitudes when the agents were not responsive to time pressure. In addition, Figure 6.c and Figure 6.d show the average joint-payoff and the average success rate for the same set of interactions while the agents were responsive to time pressure.

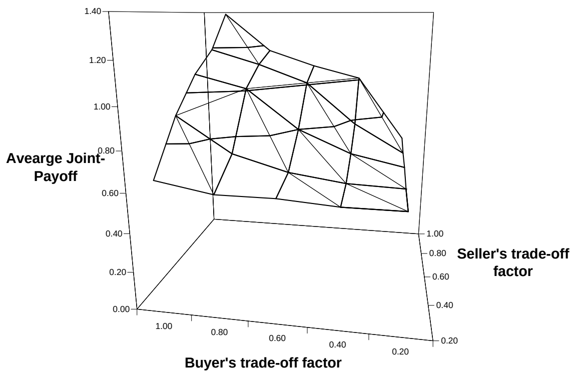
By comparing Figure 6.a and Figure 6.b, it should not be difficult to find that the average joint-payoff was the lowest when both agents were benevolent ($\alpha_{buyer} = \alpha_{seller} = 0.2$). However, the average success rate was the highest at this point. On the other hand, when both agents were fully self-interested ($\alpha_{buyer} = \alpha_{seller} = 1$), the average joint-utility was the highest although the average success rate was the lowest (average success rate = 0.1). So, our experiment confirmed the findings reported in the literature. Even though there is a temptation for agents to be fully self-interested (e.g., achieving a high payoff as shown in our simulation), they may in fact be worse off since most of these encounters cannot lead to agreements. From Figure 6.c and Figure 6.d, it also confirms that our proposed GA-based adaptive negotiation mechanism, which can take into account the time pressure, leads to better outcome in terms of success rate. The average success rates were substantially improved under various encounters while the joint-payoffs remained more or less the same when compared with the results produced by the agents which were not responsive to time pressure (Figure 6.a and Figure 6.b). Therefore, the fourth hypothesis is supported based on this experiment. In fact, not only the outcomes of the extreme encounters was improved by employing the proposed adaptive negotiation mechanism that takes into account of time pressure Eq.(5) but also the outcomes of most of the agent encounters.



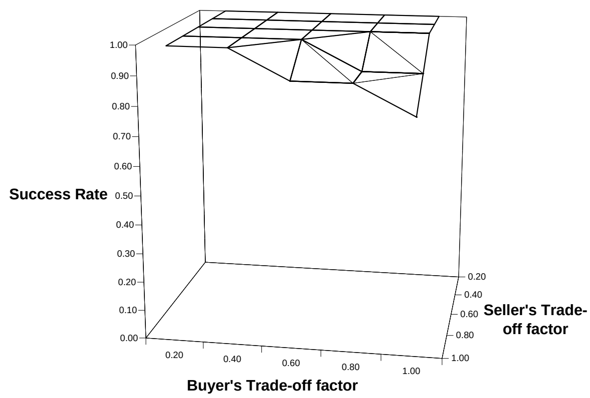
a. Average Joint-Payoff (Non-adaptive)



b. Average Success Rate (Non-adaptive)



c. Average Joint-Payoff (Adaptive)



d. Average Success Rate (Adaptive)

Figure 6. The Interactions of Agent Attitudes

Experiment 5

The purpose of the fifth experiment was to explore multi-lateral negotiation situations which consists of two buyer agents (B1, B2) and two seller agents (S1, S2). 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 as described before. Each negotiation case in this experiment 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). Similar to the previous experiments, an agreement zone always exists in a negotiation case.

A synchronised alternate-offering protocol was used in this experiment. 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 each of its opponents (e.g., $S1 \rightarrow B1$, $S1 \rightarrow B2$) according to a pre-defined order. After the message exchange phase in each negotiation round, the simulation controller would randomly serialise the offer evaluation processes in a particular sequence (e.g., $< B2, B1, S1, S2 >$). Then, 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. As a result, each agent can determine if the $o_{opponent}$ should be accepted or not with reference to its proposal in the current round. 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.

Table IV. Comparative Performance (GA) vs. (Baseline) in Multilateral Negotiations

	(GA)		(Baseline)	
Agent	Average Payoff	Average Time (rounds)	Average Payoff	Average Time (rounds)
B1	0.62	67.9	0.18	96.2
B2	0.59	67.6	0.19	95.8
S1	0.47	67.9	0.13	95.9
S2	0.48	68.1	0.11	96.1

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 IV. 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 [41, 42, 43, 45] $e_p = 1$ was employed by each agent.

The same set of negotiation cases was assigned to the agents developed based on the basic negotiation mechanism defined in Section 3. The results of these agents are listed under the (Basic) columns in Table IV. From Table IV, it is obvious that the GA-based adaptive negotiation agents again out-perform the negotiation agents which guarantee Pareto optimal in terms of both average payoff (effectiveness) and average negotiation time (efficiency). One of main reasons is that the the non-learning agents cannot find negotiation solutions (i.e., agreements) in many cases under a tough deadline of 100 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, agreements can be made in most of the negotiation cases.

6. Related work

Many researchers in automated negotiation [14, 25, 49] have stressed the necessity and importance of developing adaptive negotiation mechanisms which allow negotiation agents to learn their opponents' preferences and concession making behaviour in dynamic negotiation environments. However, existing learning approaches for automated negotiation such as Bayesian learning [5, 50] and Case-Based Reasoning (CBR) [46, 52] are still primitive in terms of what can be learned (e.g., price only) and their sensitivity to the changing negotiation contexts. Recently, there have been several proposals and implementations which employ genetic algorithms to empower negotiation agents with learning and adaptation capabilities [1, 6, 15, 27, 28, 39].

Naive Bayesian classifier has been proposed to develop the learning mechanisms of negotiation agents in multiagent systems [5]. It is argued that learning capabilities of autonomous agents are essential since agents may not be willing to share their preferences and the communication costs of exchanging full preferences among the agents could be high. Negotiation is treated as a process in which the agents refine their joint agreement set from the initial set of all feasible agreements to a single final agreement. Basically, such a refinement process is modelled as hill-climbing search in an agreement tree. The root node of the agreement tree contains all feasible agreements and all the leaf nodes are singleton sets. The set of all children of an intermediate node is a partition of that node. Negotiation is a coordinated search through the tree to find a leaf (agreement) acceptable by all the agents. The search involves two stages, asynchronous refinement and collective refinement. In the asynchronous refinement stage, an agent tries to maximise the social welfare (global utility) based on its own refinement bias function (i.e., a utility function). If all agents propose exactly the same refinement (i.e., agreement set), the particular refinement will be adopted as the solution at the collective refinement stage; otherwise the agents exchange their preferences and attempt to find an alternative refinement to maximise the group's payoff at the collective refinement stage. This process may involve a back-tracking to a node at a higher level if all the nodes below the current level do not lead to a joint agreement.

As can be seen, the main issue lies on the refinement bias function used by an agent at the asynchronous refinement stage. A refinement bias function (i.e., a utility function) can be expressed by: $U^a_{group}(o) = U_a(o) + \sum_{b \in \{A-a\}} U_b(o)$, where $U_{group}(o)$ is an agent a's perception of the joint payoff for an offer o, and $U_a(o)$ is agent a's own utility function. $\sum_{b \in \{A-a\}} U_b(o)$ represents agent a's view of the joint payoff perceived by other agents and A is the set of agents participating in negotiation. In the extreme case, an agent may adopt the ignorance approach such that $U^a_{group}(o) = |A| \times U_a(o)$ is computed. In other words, the agent only used its own utility function $U_a(o)$ to replace other agents' utility functions. However, this approach may lead to much conflict at the collective refinement stage. An agent can learn others' preferences based on their previous negotiation history. In particular, the Naive Bayesian classifier $Pr(c_i|\mathbf{x}) = \frac{\prod_j Pr(e_j|c_i)Pr(c_i)}{Pr(\mathbf{x}}$, where c_i represents the discretised utility value (i.e., a class) and e_j is one of the features of the object (i.e., offer) $\mathbf{x}$. In other words, given an offer $\mathbf{x}$, the probability of it generating a payoff c_i for an agent is approximated by the prior probabilities (e.g., $Pr(e_j|c_i)$ and $Pr(c_i)$) which could be obtained from the past negotiation history. The expected payoff of an offer as perceived by an agent b is: $U^{exp}_b(o) = \sum_i Pr(c_i|o) \times \overline{c_i}$, where $\overline{c_i}$ is the mean of the discretised utility c_i. Through simulated experiments, it was found that the agents with learning mechanisms underpinned by the Naive Bayesian classifier performed better than those agents without the capabilities of learning others preferences in terms of reduced solution cost (i.e., higher joint payoff) and reduced number of messages exchanged during the negotiation process. Nevertheless, one important issue which was not explained clearly in the paper [5] is how the prior probability $Pr(e_j|c_i$ is obtained. The prior probability $Pr(e_j|c_i$ implies that the opponents need to disclose their preference (utility) values for some negotiation issues during the previous negotiation process. Such a requirement may contradict the original assumption of the negotiation process in that agents tend to keep their preference information private.

It has been argued that the challenge of research in negotiation is to develop models that can track the shifting tactics of negotiators [27]. Accordingly, a genetic algorithm based negotiation mechanism is developed to model the dynamic *concession matching behaviour* arising in bi-lateral negotiation situations [27]. 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: $fitness(o) = P_{self} - \lambda \times P_{opponent}$, which is derived based on Social Judgement Theory [4]. P_{self} and $P_{opponent}$ represent the payoffs of an agent and its opponent respectively. The parameter $\lambda \in \{1, -1\}$ indicates if a negotiation agent is situated in a competitive or a cooperative situation. The parameter λ is established according to the opponent's concession matching behaviour in previous negotiation rounds. For example, if the opponent does not concede at all in previous rounds, it indicates that an agent is faced with a highly competitive situation and so λ should be set to 1 to penalise the opponent. Based on 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). If an agent receives an incoming offer that is perceived to yield a higher payoff than that of its opponent, it will accept the offer. The main drawback of this negotiation model is that it requires the knowledge about the opponent's utility function to compute $P_{opponent}$ and to evaluate an incoming (counter-)offer. The system is not equipped with a learning mechanism to learn the opponent's preferences; it is assumed that an agent and its opponent have the same utility function. The GA-based multi-agent multi-issue negotiation mechanism proposed in this paper does not rely on the assumption of the opponents' negotiation preferences. Moreover, the proposed negotiation mechanism is *adaptive* in the sense that negotiators' changing preferences and negotiation behaviour are taken into account by the proposed GA.

An agent-based multi-issue negotiation model has been developed based on a genetic algorithm [6]. Essentially, this negotiation model is mainly based on Krovi's work [27] and focuses on learning an opponent's concession matching behaviour. However, the main difference between these systems lies on the computation of the λ parameter. Choi et. al.'s negotiation mechanism [6] seems to demonstrate a large departure from the Social Judgment Theory. Though a learning mechanism based on stochastic approximation is proposed to learn the opponent's negotiation preferences, the given formulas do not show clearly how the weight of a negotiation issue is derived. To strike for a better balance between exploitation-oriented and exploration-oriented genetic search, adaptive mutation operator is proposed.

Rubenstein-Montano and Malaga have also reported a GA-based negotiation mechanism for searching optimal solutions for multiparty multi-objective negotiations [39]. 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. A chromosome is used to represent a payoff matrix (e.g., each column represents a negotiation party and each row represents a negotiation issue in the matrix). The negotiation performance of the GA enhanced by the trade operator is compared to other approaches such as a similar GA without the trade operator, a nonlinear programming method, a hill-climber, and a random searcher. Their experimental results show that the GA with the trade operator performs better than the other approaches in terms of maximising the joint payoffs in *distributive* negotiation scenarios involving four issues. However, the main problem of this 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. Because of these assumptions, such a negotiation mechanism seems to have very limited use in real-world negotiation environments such as e-Business where a negotiator's preferences are often kept private. Our proposed negotiation mechanism does not assume complete knowledge about a negotiation space. Instead, the GA-based adaptive negotiation mechanism can observe and gradually learn the opponents' negotiation preferences.

Genetic algorithm has also been applied to learn effective rules to support the negotiation processes [33]. A chromosome represents a negotiation (classification) rule rather than an offer. Each classification rule has the pattern $< p, s >:< p^{'}, s^{'} >$, where p and s represent the purchaser's current offer and the seller's current offer respectively, and $p^{'}$ and $s^{'}$ are the subsequent offer of the purchaser and the subsequent counter-offer of the seller. The fitness of a chromosome (a rule) is measured in terms of how many times the rule has contributed to reach an agreement. If a rule is fired and subsequently it leads to an agreement, one unit of fitness is added to the rule. Standard genetic operators such as roulette wheel selection, crossover, mutation are used to evolve populations. 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 concession generation mechanism, a GA has been used to learn optimal negotiation tactics given a particular negotiation situation (e.g., a predefined amount of negotiation time) [32]. The negotiation tactics such as time-dependent tactics, resource-dependent tactics, and behaviour-dependent tactics are examined. A negotiation agent and its tactics, in particular, the various parameters associated with these tactics, are represented by a chromosome. To compute the fitness of a chromosome, an agent needs to negotiate with each partner in a market-place to obtain the average joint-payoff. Then, such a joint-payoff is compared to that obtained at the theoretical Nash equilibrium point. It is found that using a weighted combination of tactics to determine the concession value of each negotiation issue leads to more robust results (e.g., in terms of stable joint-payoffs) under varying negotiation situations. Similar approach has also been applied to find the optimal negotiation parameters for a set of pre-defined negotiation tactics under a case-based negotiation architecture and a fuzzy rule-based negotiation architecture respectively [31]. Nevertheless both of these approaches [32, 31] suffer from the serious limitation in that they only work under a centralised decision making mechanism where complete information about each negotiator's preferences is available. On the contrary, the approach discussed in this paper is based on the assumption of a decentralised decision making model where each agent makes its own negotiation decision according to the respective GA. Such an assumption better reflects the real-world negotiation situations such as negotiations for e-Business.

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. The empirical results produced by the GA are compared to the equilibrium outcomes generated by a game-theoretic analysis [13]. Each chromosome encodes an agent's strategy which consists of four elements: an agent's initial price, the final price, the eagerness factor simulating a wide spectrum of concession behaviour (e.g., extreme Boulware to extreme Conceder), and the agent's negotiation deadline. The opponent's reservation price is assumed known and is used to establish an agent's initial price. Only the agent's final price and the eagerness factor will be evolved in a competitive co-evolution process which involves two sub-populations representing a set of buyers and sellers respectively. The average utility achieved by an agent (after applying its particular strategy encoded on a chromosome) is used to measure the fitness of a chromosome (i.e., a strategy). In fact, the GA is not used to build the agents' decision-making mechanisms but to learn the negotiation strategies which lead to optimal joint utilities. The conclusion is that a stable state produced by the evolutionary model does not always match an equilibrium outcome generated by the game-theoretic model. However, 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.

In many real-life negotiation settings such as business process management and e-Commerce, negotiation agents have only limited information about their opponents and limited computational resources (bounded rationality) to deliberate solutions. Under such circumstance, it is impossible to predict or specify equilibrium strategies a priori. Therefore, employing a heuristic learning approach to develop the adaptive decision making mechanisms of negotiation agents is desirable. A fuzzy similarity-based trade-off mechanism is developed to search for near optimal negotiation solutions under the constraints imposed by real-world applications [40, 11]. Conceptually, the set of feasible offers $\text{iso}_\alpha(\theta) = \{\mathbf{x} | U_\alpha(\mathbf{x}) = \theta\}$ lying on the indifference curve for a given utility aspiration level θ of an agent α is first identified. Then, a hill-climbing algorithm is applied to learn the trade-off strategy, that is, an offer $\mathbf{z} = \text{trade-off}_\alpha(\mathbf{x}, \mathbf{y}) = \arg\max_{\mathbf{z} \in \text{iso}_\alpha(\theta)} \{Sim(\mathbf{z}, \mathbf{y})\}$, such that z is most similar to the opponent's latest offer $\mathbf{y}$ to maximise the chance of reaching an agreement. The usual interpretation of a fuzzy similarity function $Sim_j(x_j, y_j)$ is: $Sim_j(x_j, y_j) = \min_{1 \le i \le m}(h_i(x_j) \leftrightarrow h_i(y_j))$, where x_j, y_j are the issues subject to comparison, and $h_i(x_j), h_i(y_j)$ are the corresponding criteria functions. The notation $\leftrightarrow$ represents a fuzzy equivalence operator. Nevertheless, Faratin et. al.'s implementation [11] adopts a weighted mean approach to define the fuzzy similarity operation: $Sim_j(x_j, y_j) = \sum_{1 \le i \le m} w_i \times (h_i(x_j) \leftrightarrow h_i(y_j))$. Experimental results confirm that the heuristic algorithm is effective in generating trade-offs in a range of negotiation contexts.

7. Conclusions

Real-world negotiation scenarios such as those found in B2B environment 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 equipped with effective learning mechanisms to automatically acquire domain knowledge from the negotiation environments and continuously adapt to the dynamic negotiation contexts. Our proposed GA-based adaptive negotiation agents are empowered by the effective evolutionary learning mechanisms such that these agents can learn the opponents' preferences gradually and continuously adapt to the changing negotiation contexts. The design of our GA-based adaptive negotiation agents fulfils all the requirements of practical negotiation systems because these agents can mimic a wide spectrum of negotiation attitudes, identifying near optimal solutions based on limited information about the negotiation spaces, continuously learning the opponents' preferences, and adapting to the changing negotiation contexts. Our empirical study shows that under realistic negotiation conditions, the GA-based adaptive negotiation agents outperform a theoretically optimal negotiation model which generally guarantees Pareto optimal. In addition, the proposed GA-based adaptive negotiation mechanism that takes into account of time pressure can improve the negotiation outcomes (e.g., joint payoff and success rate) under various agent encounters. 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 using adaptive genetic operators and by taking into account the market-oriented factors in multi-lateral negotiations [41, 42, 43, 45].

References

1. Baber, V., R. Ananthanarayanan, and K. Kummamuru: 2002, 'Evolutionary Algorithm Approach to Bilateral Negotiations'. In: E. Lutton, J. A. Foster, J. Miller, C. Ryan, and A. G. B. Tettamanzi (eds.): *Proceedings of the 4th European Conference on Genetic Programming, EuroGP 2002*, Vol. 2278 of *Lecture Notes in Computer Science*. Kinsale, Ireland, pp. 203–212.
2. Barbuceanu, M. and W.-K. Lo: 2001, 'Multi-attribute Utility Theoretic Negotiation for Electronic Commerce'. In: F. Dignum and U. Cortés (eds.): *Agent-Mediated Electronic Commerce III — Current Issues in Agent Based Electronic Commerce Systems*. pp. 15–30.
3. Bazerman, M., T. Magliozzi, and M. Neale: 1985, 'Integrative bargaining in a competitive market'. *Organization Behavior and Human Decision Processes* **35**, 294–313.
4. Brehmer, B. and K. Hammond: 1977, 'Cognitive factors in interpersonal conflict'. In: D. Druckman (ed.): *Negotiations: Socio-Psychological Perspectives*. Sage, pp. 216–270.
5. Bui, H. H., S. Venkatesh, and D. Kieronska: 1999, 'Learning Other Agents' Preferences in Multiagent Negotiation using the Bayesian Classifier'. *International Journal of Cooperative Information Systems* **8**(4), 275–293.
6. Choi, S. P. M., J. Liu, and S.-P. Chan: 2001, 'A genetic agent-based negotiation system'. *Computer Networks* **37**(2), 195–204.
7. de Jong, E. D. and J. B. Pollack: 2004, 'Ideal Evaluation from Coevolution'. *Evolutionary Computation* **12**(2), 159–192.
8. Duda, R. and P. Hart: 1973, *Pattern Classification and Scene Analysis*. New York, New York: John Wiley & Sons.
9. Ehtamo, H., E. Ketteunen, and R. Hämäläinen: 2001, 'Searching for joint gains in multi-party negotiations'. *European Journal of Operational Research* **130**, 54–69.
10. Faratin, P., C. Sierra, and N. R. Jennings: 1998, 'Negotiation Decision Functions for Autonomous Agents'. *Journal of Robotics and Autonomous Systems* **24**(3-4), 159–182.
11. Faratin, P., C. Sierra, and N. R. Jennings: 2002, 'Using Similarity Criteria to Make Issue Trade-offs in Automated Negotiations'. *Artificial Intelligence* **142**(2), 205–237.
12. Faratin, P., C. Sierra, N. R. Jennings, and P. Buckle: 1999, 'Designing Responsive and Deliberative Automated Negotiators'. In: *Proceedings of the AAAI Workshop on Negotiation: Settling Conflicts and Identifying Opportunities*. Orlando, FL, pp. 12–18.
13. Fatima, S., M. Wooldridge, and N. R. Jennings: 2003, 'Comparing Equilibria for Game-Theoretic and Evolutionary Bargaining Models'. In: *Proceedings of the International Workshop on Agent-Mediated Electronic Commerce V*. Melbourne, Australia, pp. 70–77.
14. Fatima, S., M. Wooldridge, and N. R. Jennings: 2004, 'An agenda based framework for multi-issues negotiation'. *Artificial Intelligence* **152**(1), 1–45.

15. Gerding, E., D. van Bragt, and H. L. Poutré: 2003, 'Multi-Issue Negotiation Processes by Evolutionary Simulation, Validation and Social Extensions'. *Computational Economics* **22**, 39–63.
16. Goldberg, D.: 1989, *Genetic Algorithms in Search, Optimisation and Machine Learning*. Reading, Massachusetts: Addison-Wesley.
17. Guttman, R., A. Moukas, and P. Maes: 1998, 'Agent-mediated Electronic Commerce: A Survey'. *Knowledge Engineering Review* **13**(2), 147–159.
18. Harsanyi, J. and R. Selten: 1972, 'A generalised nash solution for two-person bargaining games with incomplete information'. *Management Sciences* **18**(5), 80–106.
19. He, M., N. R. Jennings, and H. Leung: 2003, 'On agent-mediated electronic commerce'. *IEEE Trans. on Knowledge and Data Engineering* **15**(4), 985–1003.
20. Heiskanen, P.: 1999, 'Decentralized method for computing Pareto solutions in multiparty negotiations'. *European Journal of Operational Research* **117**, 578–590.
21. Jennings, N. R., P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge: 2001, 'Automated negotiation: prospects, methods and challenges'. *Journal of Group Decision and Negotiation* **10**(2), 199–215.
22. Jonker, C. and J. Treur: 2001, 'An Agent Architecture for Multi-Attribute Negotiation'. In: B. Nebel (ed.): *Proceedings of the seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*. pp. 1195–1201.
23. Keeney, R. and H. Raiffa: 1993, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge, UK: Cambridge University Press.
24. Kraus, S.: 1997, 'Negotiation and Cooperation in Multi-Agent Environments'. *Artificial Intelligence* **94**(1–2), 79–97.
25. Kraus, S., K. Sycara, and A. Evenchik: 1998, 'Reaching Agreements through Argumentation: A Logical Model and Implementation'. *Artificial Intelligence* **104**(1–2), 1–69.
26. Krause, P., S. Ambler, M. Elvang-Goransson, and J. Fox: 1995, 'A logic of argumentation for reasoning under uncertainty'. *Computational Intelligence* **11**, 113–131.
27. Krovi, R., A. Graesser, and W. Pracht: 1999, 'Agent Behaviors in Virtual Negotiation Environments'. *IEEE Transactions on Systems, Man, and Cybernetics* **29**(1), 15–25.
28. Lau, R.: 2002, 'Evolutionary Approach for Adaptive Negotiation Agents in B2B E-commerce'. In: L. Wang, K. C. Tan, T. Furuhashi, J.-H. Kim, and X. Yao (eds.): *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning*. Singapore, pp. 577–581.
29. Lau, R., M. Tang, and O. Wong: 2004, 'Towards Genetically Optimised Responsive Negotiation Agents'. In: *Proceedings of the 4th IEEE/WIC International Conference on Intelligent Agent Technology*. Beijing, China, pp. 295–301.
30. Lomuscio, A. R. and N. R. Jennings: 2003, 'A Classification Scheme for Negotiation in Electronic Commerce'. *Journal of Group Decision and Negotiation* **12**(1), 31–56.
31. Matos, N. and C. Sierra: 1998, 'Evolutionary Computing and Negotiating Agents'. In: P. Noriega and C. Sierra (eds.): *Proceedings of the First International Workshop on Agent Mediated Electronic Trading (AMET-98)*, Vol. 1571 of *Lecture Notes in Artificial Intelligence*. Minneapolis, MN, pp. 126–135.
32. Matos, N., C. Sierra, and N. R. Jennings: 1998, 'Determining successful negotiation strategies: an evolutionary approach'. In: Y. Demazeau (ed.): *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS-98)*. Paris, France, pp. 182–189.
33. Matwin, S., T. Szapiro, and K. Haigh: 1991, 'Genetic Algorithm Approach to a Negotiation Support System'. *IEEE Transactions on Systems Man and Cybernetics* **21**(1), 102–114.
34. Nash, J. F.: 1950, 'The Bargaining Problem'. *Econometrica* **18**, 155–162.
35. Parsons, S., C. Sierra, and N. Jennings: 1998, 'Agents that Reason and Negotiate by Arguing'. *Journal of Logic and Computation* **8**(3), 261–292.
36. Pruitt, D.: 1981, *Negotiation Behaviour*. Academic Press.
37. Raiffa, H.: 1982, *The Art and Science of Negotiation*. Harvard University Press.
38. Rosenschein, J. and G. Zlotkin: 1994, 'Task Oriented Domains'. In: *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. Cambridge, Massachusetts: MIT Press, pp. 29–52.
39. Rubenstein-Montano, B. and R. A. Malaga: 2002, 'A Weighted Sum Genetic Algorithm to Support Multiple-Party Multi-Objective Negotiations'. *IEEE Transactions on Evolutionary Computation* **6**(4), 366–377.

40. Sierra, C., P. Faratin, and N. R. Jennings: 1999, 'Deliberative Automated Negotiators Using Fuzzy Similarities'. In: *Proceedings of the EUSFLAT-ESTYLF Joint Conference on Fuzzy Logic*. Palma de Mallorca, Spain, pp. 155–158.
41. Sim, K.: 2004, 'Negotiation Agents that make prudent compromises and are slightly flexible'. *Computational Intelligence, Special Issue on Business Agents and the Semantic Web* pp. 643–662.
42. Sim, K. and C. Choi: 2003, 'Agents that react to changing market situations'. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics* **33**(2), 188–201.
43. Sim, K. and S. Wang: 2004, 'Flexible Negotiation Agent with Relaxed Decision Rules'. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics* **34**(3), 1602–1608.
44. Sim, K. M.: 2002, 'A Market-driven Model for Designing Negotiation Agents'. *Computational Intelligence, Special Issue in Agent Technology for E-commerce* **18**(4), 618–637.
45. Sim, K. M.: 2003, 'Equilibrium Analysis of Market-driven Agents'. *ACM SIGECOM: E-commerce Exchange* **4.2**, 32–40.
46. Sycara, K.: 1989, 'Multi-agent compromise via negotiation'. In: L. Gasser and M. Huhns (eds.): *Distributed Artificial Intelligence II*. Morgan Kaufmann, pp. 119–139.
47. von Neumann, J. and O. Morgenstern: 1994, *The Theory of Games and Economic Behaviour*. Princeton University Press.
48. Wooldridge, M. and N. Jennings: 1995, 'Intelligent Agents: Theory and Practice.'. *Knowledge Engineering Review* **10**(2), 115–152.
49. Zeng, D. and K. Sycara: 1997, 'Benefits of Learning in Negotiation'. In: *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI-97/IAAI-97)*. Menlo Park, pp. 36–42.
50. Zeng, D. and K. Sycara: 1998, 'Bayesian Learning in Negotiation'. *International Journal of Human-Computer Studies* **48**(1), 125–141.
51. Zeuthen, F.: 1967, *Problem of Monopoly and Economic Warfare*. London, U.K.: Routledge and Kegan-Pail.
52. Zhang, D. and W. Wong: 2000, 'A Web-Based Negotiation Agent Using CBR'. In: R. Kowalczyk and M. Lee (eds.): *Proceedings of the Second Workshop on AI in Electronic Commerce (AIEC 2000)*, Vol. 2112 of *Lecture Notes in Artificial Intelligence*. Melbourne, Australia, pp. 183–197.