UNIVERSITY OF SOUTHAMPTON

A heuristic model for concurrent bilateral negotiations in incomplete information settings

Thuc Duong Nguyen

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

May 2005

ABSTRACT

by Thuc Duong Nguyen

Multi-agent systems, in which autonomous agents interact in flexible ways, are an important new approach for developing software systems for a range of real-world problems. Here the notion of an agent (a computer program that is capable of autonomously working in its environment and interacting with other agents) is the core building block of the system. These software agents interact with one another in order to achieve their individual goals or to manage the dependencies that ensue from being situated in a common environment. Now, there are many different types of interaction that can occur in such systems. However *negotiation*, the process of collaborating agents coming to an agreement on a specific matter, is one of the most important. It is so important because it offers a means for the agents to make a mutual selection of actions which, in turn, is simply the de facto means of interaction between autonomous components.

Against this background, this research developed a model that software agents can use to drive their participation in bilateral (pairwise) encounters. Specifically, we consider the case in which the agents negotiate over multiple issues (such as the price, quality and time of delivery) and where they can engage in multiple, concurrent encounters in order to procure the same good or service. The model is targeted at realistic trading scenarios (including web service procurement and virtual organization management) and so has to be computationally efficient and be able to operate effectively with minimal information about its negotiation opponents.

To this end, we have developed a heuristic-based concurrent model that allows an agent to effectively handle simultaneous negotiations with other agents. A versatile coordination mechanism has been created to ensure that all the negotiations are inter-related to each other to ensure that only a single high value deal is reached at the end of the bargaining process. A commitment model has also been integrated to allow the agents to have more flexible behaviors and to stimulate different agents to participate in the negotiation process. Finally, an adaptive negotiation strategy has been introduced, which makes use of the information gained during the process, to improve the performance of the model in certain scenarios.

Contents

List of Figures

List of Tables

Acknowledgements

During the duration of my Ph.D., I have received a lots of advice, support and help from many people.

First and foremost, I would like to thank my supervisor, Professor Nick Jennings, for his guidance, support and constant encouragement. His indisputable knowledge in the field, his beyond duty interest in the research and his always valuable suggestions have always inspired and encouraged me in numerous ways. I have learnt a lot under my time with you. Thank you.

Second, I am grateful to BT Exact for financing my work. In particular, I would like to thank Dr. Simon Thompson for his support during my time in BT Exact as well as throughout my Ph.D. I also want to express my gratitude to all the members of the CONOISE project for their help and support.

I would also like to thank all my friends in the IAM group for their friendship, kindness, help and hospitality. They have made my years of study memorable and enjoyable. Specially, I would like to thank Dr. Steve Gunn, Dr. Sarvapali Ramchurn (Gopal) and Steve Munroe for their help both inside and outside my research time.

This thesis would not have been done if it were not because of my mother and my beloved grandparents, who were always there for me and supporting me in every step that I took. There are not enough words in any language to express my love for you.

Last, but not least, I own a big thank to my friends in UK, France and Vietnam, who made me feel their warmth and narrowed the gap imposed by the physical distance. In particular, I would like to thank Thuy Linh and my housemates for their help and support. Finally, a big thank you to you all, my friends.

This thesis is dedicated to my beloved grandparents, Johan and my mother.

Chapter 1

Introduction

The work presented in this thesis is concerned with automated negotiation between software agents. Our interests focus on a subset of the domain, called *concurrent bilateral negotiation*, in which an agent engages in a number of parallel pairwise negotiations with other agents in order to come to a mutually acceptable agreement on some matter. These negotiations may influence one another in order for that agent to gain more benefit from the bargaining process.

The structure of this chapter is as follows: the background of agents and the automated negotiation problem are described in section 1.1, followed by a number of motivating examples in section 1.2. Next, the contributions of this work are given in section 1.3, followed by the list of publications in section 1.4. Finally, section 1.5 outlines the overall structure of this thesis.

1.1 Negotiating Agents

In recent years, agent-based computing has emerged as an important new approach in computer science. This can be explained by the natural fit between agent-based concepts and those required for modelling, designing and building complex distributed systems [Jennings, 2001]. While the exact definition of agency is still a point of some debate, we use the following definition, which has also been adopted by an increasing number of researchers:

Definition 1.1. an agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives [Wooldridge, 1997].

A number of distinguishing characteristics of agent-based approaches are worth highlighting: (i) agents are *autonomous*: this means that they have control over their own actions and they can act without human intervention; (ii) agents are *reactive*: they sense the changes in their surrounding environment and react in a timely fashion to those changes; and (iii) agents are *proactive*: they do not just respond to the changes in their environment, they are able to take the initiative in order to satisfy their goals [Wooldridge and Jennings, 1995].

Although these aforementioned characteristics help individual agents to act flexibly in order to achieve their design objectives, much of the power of the agent-based approach relies on the *social*[1] characteristic of the agents [Wooldridge and Jennings, 1995; Alonso, 2002]. This characteristic is particularly important because most of the real-world problems that can benefit from using agent-based approaches require or involve multiple agents [Bond and Gasser, 1988]. Thus, as several agents are situated in a common environment, there will necessarily be some dependencies between them [Castelfranchi, 1998] (e.g. if the agents have mutually conflicting goals, the effects of the action of one agent will impinge in some way on the goals of another or if they need to access a shared and limited resource then, again, the individual action taken will affect the other agents). Given this, the agents will need to interact to manage these dependencies.

The systems in which these interactions take place are termed *multi-agent systems* (MAS). Some example systems are: *auction houses* in which the price of a good is obtained as a result of several agents competing against each other [Wurman *et al.*, 2001; Sandholm, 1999a], *grid environments* where agents negotiate to optimize computing resource allocation over time [Shen *et al.*, 2002], *information sharing environments* where agents negotiate with each other for the right to access a common source of information [Huhns and Stephens, 1999] and *e-commerce environments* where agents buy and sell goods [Maes *et al.*, 1999].

In these exemplar systems, *coordination* can be used as the generic term for all types of social interaction [Excelente-Toledo *et al.*, 2001; Calisti, 2001]. However, because it covers a broad range of topics, it is viewed differently by different researchers. For example, Bond and Gasser's view of *coordination*, as a property of interaction among some set of agents performing some collective activity [Bond and Gasser, 1988], concerns the *outcome* of the coordination only. On the other hand, Malone and Crowston's view of *coordination*, as the act of managing the inter-dependencies between activities

[1]Here, social means that the agents interact with one another (via some kind of *agent communication language* [Genesereth and Ketchpel, 1994]) in order to pursue their own objectives or to benefit the wider community.

performed [Malone and Crowston, 1990], concerns the *process* of the coordination. Regardless of the viewpoint, however, a successful attempt to coordinate agents' actions should result in a coherent overall system performance in which actions do not conflict with one another. If, however, coordination is unsuccessful then the system may behave incoherently (e.g. the community may quickly degenerate into a collection of chaotic, incohesive individuals [Jennings, 1996]). In the aforementioned *auction house*, for example, if the coordination attempt succeeds (the agents propose their offers and counter-offers in a specific manner that is determined by the auction protocol) then the auction is considered legal and the final price will be accepted by all the parties. On the other hand, unsuccessful coordination could result in the wasting of time and resources and no final price being agreed upon. In the *information sharing* example, successful coordination results in each participating agent being able to access the information it needs. However, if coordination fails, the agents may duplicate unnecessary activities or block one another from accessing their target information [Huhns and Stephens, 1999].

While *coherence* is one reason why *coordination* is needed, it is not the only one[2]. Other motivating factors include the fact that [Jennings, 1996]:

- *global constraints exist*: there are global conditions that all the agents must satisfy (e.g. the total allowed money for a project is fixed or no more than two agents are allowed to access the resource at the same time). If the agents are allowed to act individually and each tries to maximize its own profit then the global constraints are unlikely be met.

- *solving the problem requires the cooperation of the agents*: due to the limited information, resources and skills of a single agent, it is unlikely that a problem can be solved by only one individual (e.g. building a house requires architects, builders and electricians).

As can be seen, coordination is needed for a variety of reasons and it encompasses a variety of behaviors. Consequently, a number of different techniques have been developed to coordinate the behaviors of multi-agent systems [Excelente-Toledo, 2003]. These include *iteratively exchanging partial global plans* until all the constraints are satisfied [Durfee and Lesser, 1991], instituting social laws in order to avoid harmful interactions [Shoham and Tennenholtz, 1995], using contracting protocols to allocate

[2]Even when coordination is not needed (e.g. agents' actions are independent and resources are plentiful), it may still be beneficial if the agents are coordinated [Faratin, 2001]. For example, in searching for information, an independent discovery by one agent can help reduce the complexity of another by sharing its discovery and so pruning its search space [Decker, 1995].

task [Smith, 1980; Davis and Smith, 1983], using *authority hierarchies* to form organizational structures [Carley and Gasser, 1999; Fox, 1981], *exchanging bids* in a market place to allocate tasks [Wellman, 1993] and *negotiating* to pursue agreements [Lomuscio *et al.*, 2003; Jennings *et al.*, 2001].

For the purposes of this research, however, we focus on *negotiation* between agents that are self-interested (i.e. that they try to do as well as they can for themselves). We do so because this is the method humans use when attempting to reach an agreement on a variety of issues [Raiffa, 1982; Pruitt, 1981; Hiltrop and Udall, 1995] and we believe it will be the dominant operation mode for autonomous agents when facing similar situations [Lomuscio *et al.*, 2003; Jennings *et al.*, 2001]. Such negotiation will become ubiquitous because it allows the agents to communicate in a structured manner that should enable them to come to an agreement while minimizing the amount of information that needs to be revealed to each other[3]. Furthermore, automated negotiation among self-interested agents is becoming increasingly important in MAS because of the following reasons [Sandholm, 1999b]:

- *technology push*: with the advent of standardized communication infrastructures (the Internet, Grid, Semantic Web), negotiation allows agents from different organizations with different designs to interact in real time in an open environment so that they can carry out different transactions.
- *application pull*: at the decision making level, there is an increasing demand for computer application support for negotiation. Relevant examples include the introduction of transactional e-commerce on the Internet for purchasing goods [Maes *et al.*, 1999] or information [Kalakota and Whinston, 1996] and the resource-scheduling negotiation system in grid environments [Shen *et al.*, 2002].
- *efficiency*: automated negotiation among agents does not require the intervention of humans, thus saving time for human negotiators. Furthermore, in many cases, the agents can be more effective in finding strategically beneficial contracts than humans in complex negotiation settings [Sandholm, 1999b].

In more detail, negotiation is here defined as:

Definition 1.2. the process by which a group of agents communicate with one another to try to reach agreement on some matter of common interest [Lomuscio *et al.*, 2003].

[3]When all the agents are self-interested, any information about the evaluation criteria of one agent can be a huge beneficial advantage to another agent (see chapter 2 for more details).

In the context of this research, the matter of common interest or the subject of the negotiation is termed a *service*. It is, in an abstract way, the capability of an agent that can be beneficial to the society or to the other agents. This definition is a common concept in the domains e-commerce, grid and web services; indeed it is fundamental to the general area of service-oriented computing [Singh and Huhns, 2005; de Roure *et al.*, 2003; Faratin *et al.*, 2003; Payne *et al.*, 2002]. Now, there are a variety of service examples, ranging from a simple capability such as data retrieval, onto more advanced capabilities such as automatic bidding for goods on the Internet. However, in this context, we are primarily interested in e-commerce like scenarios and so term the agent that is capable of providing the service the *seller* and the agent that wants to purchase the service the *buyer*.

In more detail, an agent's interest in a service at any specific time period of the negotiation is represented by a *proposal*. In most negotiation settings, the agents iteratively exchange proposals with one another until one proposal is agreed by all participants (in this case, the negotiation successfully terminates with an agreement) or one agent opts out (in this case, the negotiation terminates without an agreement). The final proposal in a successful negotiation (an agreement) is considered as a statement of the rights and obligations of each party to a transaction or transactions [Bannock *et al.*, 1992]. Depending on the specifics of the negotiation, a proposal can cover a single issue (e.g. price) or multiple issues (e.g. price, quality and quantity).

In this service-oriented context, there are three broad types of negotiation interactions depending on the number of participants:

- *many-to-many*: this is the most complex negotiation setting, where there are multiple sellers and multiple buyers participating. This type of negotiation is typically dealt with using a complex auction settings such as a double auction [Friedman and Rust, 1992; Wurman *et al.*, 2001]
- *one-to-many*: there are two cases of this type of negotiation: (1) one seller and multiple buyers and (2) one buyer and multiple sellers. The former is the standard setting of many auctions and it is very popular on the Internet (e.g. `http://www.ebay.co.uk`, `http://auctions.yahoo.com`) while the latter is the standard case for price comparison engines (e.g. `http://www.dealtime.co.uk`, `http://www.pricewatch.com`) [Lomuscio *et al.*, 2003; Anthony, 2003].
- *one-to-one* or *bilateral*: this is a very common type of negotiation [Jennings *et al.*, 2001; Lomuscio *et al.*, 2003] and it involves a single seller and a single buyer. Unlike the previous two negotiation types, bilateral negotiation cannot

be efficiently handled by auction techniques because these techniques rely on competition; thus, they will not be effective in this situation. Furthermore, the participating agents are able to exchange information both from the seller to the buyer and from the buyer to the seller (e.g. the seller can express its opinion about the buyer's proposal via some form of counter-offer, which the buyer can then use to refine its offer to send back to the seller) [Lomuscio *et al.*, 2003; Nguyen and Jennings, 2005]. This information exchange allows the agents to flexibly express their preferences and, thus, requires more flexible techniques than those typically found in an auction to search for offers in the agreement space.

To this end, this research focuses on applying bilateral negotiation techniques into one-to-many service oriented negotiations between self-interested agents. In particular, we aim to use multiple concurrent bilateral negotiations as an alternative to both standard bilateral negotiations and to traditional auction techniques that are normally used in one-to-many scenarios. The motivation behind this is twofold. In comparison to standard bilateral techniques, the agents can come to the agreement sooner without being in a worse position. In comparison to standard auction techniques, the participating agents can be more flexible in bargaining with one another and so can avoid the rigidness of the standard auction protocols. Consequently, we believe the outcome of the negotiation can be more beneficial to the agents than that achieved using either the standard bilateral or the standard auction approaches. Furthermore, by using bilateral negotiation techniques, the negotiation can be tailored to each individual opponent, potentially giving the negotiating agent more bargaining power and, thus, allowing it to obtain better outcomes.

Given this background, section 1.2 will give more insight into the motivations and the benefits of applying such techniques in our chosen setting.

1.2 Research Motivations and Requirements

Bilateral negotiation is an important research area because it is a common form of negotiation in a range of different domains (including e-commerce, the grid and the semantic web) [Lomuscio *et al.*, 2003]. To this end, a number of different models have been developed (see chapter 2 for more details). These models can be broadly characterized by the following nomenclature [Jennings *et al.*, 2001]:

- *game theory*: using these techniques, a negotiation is considered as a game of two players. Here, a game is informally defined as the rules of an encounter between players that have strategies and associated payoffs. This method normally guarantees to find the optimal solution for all players. However, game theory methods are often based on unrealistic assumptions (e.g. the players' evaluation criteria, deadlines, pricing structures and so on are often assumed to be commonly known) and in many cases the computation required is intractable (since these methods normally need to take into account all the possible states of the environment, as well as potential behaviors of the players in order to make a decision, this requirement typically requires unbounded computational capability). Refer to section 2.1 for a more detailed review.

- *heuristic*: these methods are typically built upon realistic assumptions (e.g. each agent does not known its opponent's evaluation criteria, deadlines and so on) and they take into account the computational requirements of realizing the models. As a result, these methods are applicable in more general situations than game theory based techniques (e.g. realistic negotiation scenarios, where such assumptions are inapplicable to the participating agents). However, the outcomes generated are often sub-optimal and they often require extensive empirical analysis in order to ascertain their operational characteristics. Refer to section 2.2 for a more detailed analysis of heuristic models.

- *argumentation-based*: these methods are a relatively new addition to the automated negotiation area. They allow more information (such as threats or appeals) to be exchanged during negotiation than their game-theoretic and heuristic counterparts. In general, in argumentation-based encounters, the agents can explicitly state their opinions with supportive information (e.g. an explanation of why a contract is not acceptable). However, this approach has high computational complexity since agents have to reason about arguments (both when sending proposals and when receiving them) in addition to the basic negotiation proposals [Rahwan *et al.*, 2003].

In this research, we aim to develop a negotiation model that targets realistic scenarios. Specifically, we want our model to be applied in a variety of negotiation contexts that are not based on unrealistic assumptions and do not require unrestricted computational capabilities. As can be seen from the aforementioned description, heuristics are the most suitable method in this regard and thus this is the approach adopted in this work[4]. We do not adopt one of the standard auction models because in such cases:

[4]Even though argumentation-based models can give the agents additional bargaining power, they also require the agents to reason about common knowledge (normally represented in an ontology) and the computational requirements are much higher than their heuristic counterparts. Furthermore, the scenarios we target typically do not require the agents to have such bargaining power in order to obtain good results.

- The communication of offers and counter-offers are normally limited to flowing in one direction. For example, in a typical auction, the seller agent will have a number of proposals from the buyers and it has to select one of these proposals as its final contract. Thus, there is no way for the seller agent to modify a particular proposal to suit its needs.
- Most of the auction techniques have the problem of rigidness and high structuralism. For example, the time it takes to complete the bargaining process is typically either fixed or undetermined. Normally, an auction only ends if the deadline has passed or when no bid is received within a predefined amount of time. This is a problem when the seller agent needs to make an agreement within a specified time frame.

To this end, we propose concurrent bilateral negotiations as an alternative. Specifically, instead of using one way communication (only buyers make bids), our approach allows two way simultaneous communication of offers and counter-offers. Thus, instead of just submitting bids from one agent to another agent, a complete bilateral negotiation will be carried out between these two agents (here in this work, this particular negotiation is called a *thread*). Then, when all the negotiations have finished, the agreement that has the highest benefit to our agent will be selected as the final one. In such scenarios, the agents can now exercise various bilateral negotiation strategies to suit their interests. These abilities also allow the agents to possibly increase the social welfare since such negotiation allows the agents to come to an agreement that satisfies all of their interests. Furthermore, the time to achieve an agreement can potentially be decreased. In auctions, only the search spaces of the seller agents are reduced throughout the negotiation. In contrast, by using multiple concurrent negotiations, our model enables all the participating agents to narrow their search spaces simultaneously to find possible offers. This simultaneously space narrowing can result in both parties coming to an agreement quicker if one exists.

We also do not adopt standard bilateral negotiation models since they are specifically designed to handle two agent negotiations. In our target environment, in which there is more than one seller, these models do not provide an appropriate solution because they only consider negotiation with one of the potential sellers. It can be argued that by negotiating sequentially with all the sellers, an agreement can be reached. However, organizing the negotiation in this manner will lead to lengthy negotiations and there is a problem of ordering[5]. Furthermore, if the buyer's negotiation deadline is restricted,

[5]In sequential negotiation, the result of a successful negotiation can be used as the base for the subsequent negotiations. Thus, different orders in which the buyer agent negotiates with the seller agents will result in different outcomes of the encounter.

it may not be able to bargain with all the sellers in time. Thus, the buyer agent may miss some high value outcomes. In contrast, by negotiating concurrently, the encounter will be shorter and the ordering problem is eliminated. Specifically, the buyer agent can still use the result of a successful negotiation to influence other ongoing negotiations. Moreover, since all the negotiations with the sellers will be carried out concurrently, there is less of a problem of finishing within the agents' deadlines.

Having presented this approach, we now outline two use-case scenarios to motivate the discussion further and to illustrate the idea of *concurrent negotiation*.

1. Henry decides that he needs a short break to sort out his stress problems at work. The weekend is approaching and he decides he will go to Ibiza during this occasion. He prefers the cheapest package and he is willing to spend no more than £1000 on this trip. Previously, he would do this task via phone; negotiating with each travel agent he knows about the price and, later, selecting the cheapest one. However, he is currently very busy at work and, furthermore, there is not enough time to do this conventionally. Therefore, he would like his personal agent, called *sigma*, to help him sort out this matter.

   Given Henry's requirements, *sigma* needs to find a travel agent that provides the cheapest package in one day, more precisely, within six hours due to the fact that no travel agent is working after office hours. From the Yellow Pages, *sigma* finds ten available travel agents (called $agent_1, agent_2, \ldots, agent_{10}$) and each of them needs about one hour to find and confirm the details of the package. The communication among the agents is freely available and secure via the Internet.

   Conventionally, if *sigma* negotiates with a single travel agent until it finishes and then moves onto the next one, it can only query six out of the ten agents and it will miss the other four. This means there is a possibility that *sigma* could miss the best deal. Furthermore, there is also a problem of the order in which the travel agents should be queried (e.g. *sigma* does not know which order the travel agents should be processed: should it start with $agent_1$ or $agent_2$?).

   On the other hand, *sigma* could negotiate with all the travel agents concurrently (see figure 1.1). After all of them finish, *sigma* selects the agent that provided the cheapest package. This allows *sigma* to complete the given task in time and still achieve the desired package (as he does when he does it conventionally). Moreover, if the search with one travel agent finishes with an agreement while there are still other searches going on, *sigma* can use this agreement's value to influence the other ongoing searches. In some cases, this influencing can lead to an agreement with a better value. For example, having obtained an agreement with the

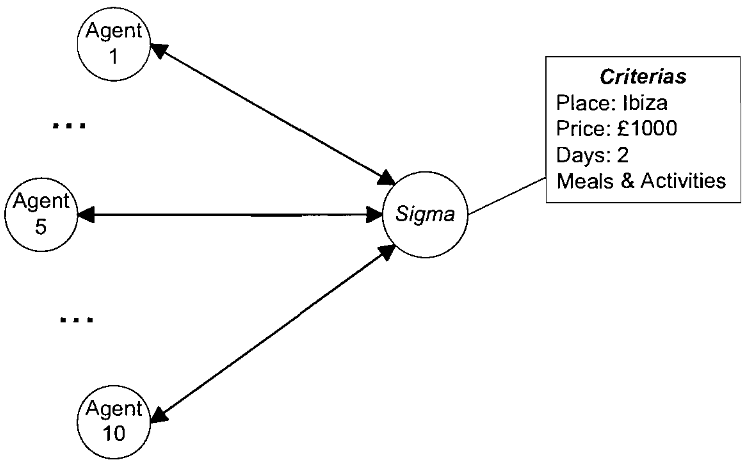


FIGURE 1.1: The holiday scenario.

$agent_4$, *sigma* may change its strategy with $agent_5$ to be tougher. Consequently, $agent_5$ observes that *sigma* suddenly becomes a tougher negotiator and so it may change its strategy to be more concessionary in the hope of finding a deal. By so doing, the offer that $agent_5$ generates may have a higher utility value to *sigma* than if $agent_5$ had not changed its strategy. This, in turn, results in a deal that has a higher utility value for *sigma*.

Comparing these two methods, it is obvious that by negotiating concurrently, *sigma* is able to complete the given task successfully without any loss in the final result and, moreover, it may possibly achieve a better result.

2. The second scenario concerns virtual organizations (VO), in which a number of agents, with different problem solving capabilities and resources, come together to form a coalition to provide a compound service to an end user [Norman *et al.*, 2003]. Now, there are two possible situations in this scenario. The first happens when a VO (composed of 3 agents $\{X, Y, Z\}$) has been formed and Z drops out (for whatever reason). The current VO should not be dismissed because its main objective has not yet been fulfilled. Instead, another agent needs to be summoned to replace Z. Given this, the agent that takes charge of the VO, called RA, then has to find the new agent within the minimum time and cost (see figure 1.2). The second situation happens when a VO has been formed and is operating. Then imagine that a new requirement is introduced that the current VO is not capable of handling. Again, one or more agents needs to be added to the grouping.

   In both scenarios, since the goals of the agents may conflict with each other (e.g. there may be different opinions about price, roles, etc.), changing the formation of the VO is achieved via the process of negotiation. Specifically, when RA needs to

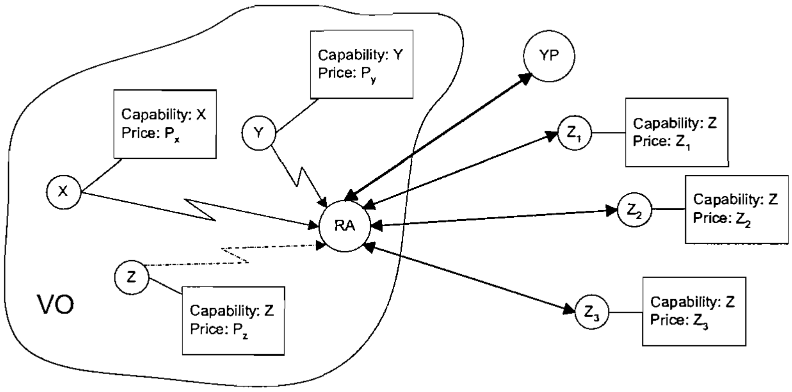


FIGURE 1.2: The VO reformation scenario.

find a particular agent for a specific requirement, it first requests the list of capable agents from a yellow page agent (YP). From this list, RA then needs to negotiate with each of the agents in order to find the most suitable candidate. Due to the requirement of the VO formation, these negotiations need to be finalized with the minimum possible operational time and cost. Thus, only by negotiating concurrently with all the possible candidates can RA be sure that it gets the best deal in the shortest time. In contrast, if this process was to be handled by a traditional auction method, a solution for this complex problem may not be guaranteed (e.g. the bids from the provider agents might not satisfy all the requirements of RA and in such cases, there is no way RA can modify the bids to show its preference. On the other hand, if RA negotiates with the candidates simultaneously, its chance of finding a mutual agreement with a particular candidate is stronger since both of them can implicitly show their preferences to one another.).

These scenarios are but two illustrations of the various practical scenarios that would benefit from using concurrent negotiations. There are clearly many others. Generalizing from this, the rest of this thesis focuses on an abstract scenario in which one *buyer* agent seeks to purchase a service that can be provided by a number of potential *sellers* agents. Moreover, we assume the agents all have time limits (hard *deadlines*) by when the negotiation must be completed (which is typical of many negotiations in e-commerce settings [Wurman *et al.*, 2001; Anthony, 2003]). For reasons of simplicity, we also assume the service will only be provided by a single provider (meaning no partial offers will be considered). In such cases, the buyer will negotiate simultaneously with all the potential providers and each of these encounters will be handled as a single bilateral negotiation (*thread*). At the end of the encounter, the agreement that has the highest value to the buyer is selected as the final one.

Now, as agreed above, by using heuristics as the search method, the agents are able to operate in incomplete information settings (e.g. they do not need to know how their opponents evaluate their proposals; meaning that when these agents select the offers to propose to one another, they do not know explicitly whether these offers will be accepted or rejected). However, in some cases, it is not uncommon for the agents to have some general information about the other agent (e.g. agent A is typically easy in negotiating or agent B tends to be a tough negotiator) [Fatima *et al.*, 2001; Zeng and Sycara, 1998]. Thus, if such information is available, it should be exploited in order to benefit the negotiating agent. For example, if agent a knows that agent b is typically easy in negotiating, a will try to be harder when negotiating with b so that if it succeeds, the utility value it gains will be higher than if it is softer. In general, there are many different types of information that might be available to our negotiating agent, however, in this work, we consider one special type of information which is related to how the other agents generate their counter-offers. Thus, if such information is available, our model will try to exploit it in order to try and gain a better outcome for our agent. To this end, section 2.5 discusses some of the current learning techniques that are capable of exploiting such information and chapter 5 details the method we devised to exploit such information and the results we achieved.

Another important issue to consider is how the concurrent negotiations should be carried out. Naturally, each negotiation is a fully featured single bilateral bargaining process and can be carried out as an independent encounter. However because they occur simultaneously, the negotiations could influence one another during the encounter. For example, if one negotiation finishes early with an agreement, the agent will have something in the bag and it can switch its negotiation stance in the other ongoing negotiations to something tougher (i.e. only conceding very slightly). Thus, if it succeeds, the result achieved will be higher than if it did not change its negotiation stance [Nguyen and Jennings, 2003b]. Similarly, towards the end of the encounter, if no agreement has been reached, the agent could switch its negotiation stance to something that concedes more rapidly in order to try and clinch a deal [Nguyen and Jennings, 2004a]. Therefore, we believe that such mutual influence among the concurrent negotiations should provide our agent with more bargaining power which should, in turn, improve its outcome in the bargaining process. Chapter 3 details this aspect of our negotiation model and presents the corresponding empirical results.

In addition to the issue of mutual influence amongst the negotiation threads, handling the agreements presents another problem. At the end of the process, only one service provider needs to be selected. Naturally, if the buyer negotiates with various providers, there is a chance that more than one negotiation will result in an agreement. Thus, after all the negotiations finish, the buyer is left with a number of (temporary) deals. This abundance of deals contrasts with the buyer's objective at the beginning of the process which is to obtain the whole desired service from a single provider. Given this, the question is what should the buyer do in this situation? Consider the following example for illustration:

Agent *sigma* needs to find a travel agent that provides the cheapest travel package. From the Yellow Pages, *sigma* finds ten available travel agents (called $agent_1, agent_2, \ldots, agent_{10}$) and negotiates with them simultaneously. Of the 10 negotiations, three were successful with final prices of £200 from $agent_1$, £250 from $agent_4$ and £150 from $agent_6$, respectively.

Since *sigma* only needs one package, it has to select one of the three to become the final agreement. However, what will it do with the remaining two offers? The simplest solution is to accept the package from $agent_6$ and discard the other two. In order to be able to do so in our model, the temporary agreements must be binding on the sellers (see section 3.1). This then allows the buyer agent to simply select the agreement that has the highest value as the final deal at the end of the negotiation and decline the others. This method gives the best outcome to the buyer agent and we discuss it thoroughly in chapter 3.

Although efficient for the buyer agent, this simple solution has the significant drawback of treating the participating agents unequally. Once a deal is agreed upon, no seller agent can renege and, moreover, that deal might not even be finalized at the end of the negotiation process. On the other hand, the buyer agent is freed from being tied to the agreements it made. This limitation reduces the desirability for the seller agents to participate. It also limits the applicability of the negotiation model in realistic negotiation settings (i.e. not all scenarios allow the buyer agent to have such a degree of freedom). Moreover, although a deal can seem profitable for an agent at the time it occurs (viewed ex ante), it might not be so at a latter time (viewed ex post) [Sandholm and Lesser, 2002]. This applies to both buyer and seller agents. Consequently, the model should allow both parties to renege from the deal if necessary (e.g. when presented with a better offer from another source or the current temporary agreement is no longer attractive). However, it does not necessarily mean that an agent can arbitrarily commit and decommit any time it wants to. It does have to be responsible for its action since its commitment has an influence on the other agent's behavior [Sen and Durfee, 1994]. To this end, there are a number of approaches that can enforce these restrictions on the agents' behaviors (see section 2.4 for more details). However, to accommodate this problem, a commitment model needs to be developed. This model needs to not only be capable of treating the agents equally, but to be able to enforce the responsibilities into the agents' decisions. In this vein, chapter 4 introduces our approach to this problem (based upon paying penalties for decommitting) and analyzes the effectiveness of the approach empirically.

To sum up, a number of requirements that need to be satisfied by the negotiation model have been identified in this subsection. These include the fact that it should:

1. be *computationally tractable*;
2. operate with *incomplete information* about the environment and the opponent;
3. exploit *partial information* about the environment and the opponent if it is available;
4. allow agents to have (private) *deadlines* by when they must reach agreements;
5. allow the negotiation threads to *mutually influence* one another;
6. produce *efficient negotiation outcomes*;
7. flexibly handle issues related to *commitments* so that the agents can reach effective deals.

1.3 Research Contributions

The work described in this thesis advances the state of the art in the following ways:

- we have developed a negotiation model that enables an agent to manage multiple concurrent bilateral negotiations [Nguyen and Jennings, 2003b]. The only other model that deals with this situation is that of [Rahwan *et al.*, 2002] and in comparison to this our model:
  - sets the initial strategies for the individual pairwise negotiations in an informed way based upon some knowledge of the prevailing market mix of agent types (see section 3.4.3)
  - enables the agent to change its individual negotiation strategies in response to its assessment of the negotiation behavior of its opponents (see section 3.4.3)

  - enables the agent to negotiate in a time-constrained environment in which there are deadlines (see section 3.4.1.3)

- we have actually implemented an agent that can engage in multiple concurrent negotiations [Nguyen and Jennings, 2003a; Nguyen and Jennings, 2004a] (note that [Rahwan *et al.*, 2002] is an abstract model only). This implementation has been used in two large scale system developments in the application areas of web service procurement and virtual organization management.

- we have developed a commitment model for concurrent negotiation that [Nguyen and Jennings, 2004b; Nguyen and Jennings, 2005]:

  - allows any agent to back down from a committed deal for any reason it deems appropriate by paying a fee to another agent (see section 4.1).
  - enables the buyer agent to have different commitment strategies based on its assessment of its negotiation situation and the behavior of its opponents (see section 4.2).
  - permits the buyer agent to make a trade-off between the number of agreements it reached and the utility value of the final agreement (see section 4.2).

- we have developed an adaptive negotiation strategy that exploits any available partial information about the sellers in order to make more effective counter-offers which, in turn, lead to better final outcomes (see chapter 5).

1.4 Published Papers

The following papers have been published from the work contained in this research:

- T. D. Nguyen and N. R. Jennings. Managing commitments in multiple concurrent negotiations. *Int. J. Electronic Commerce Research and Applications*, 4(3) (*to appear*), 2005.

- T. D. Nguyen and N. R. Jennings. Reasoning about commitments in multiple concurrent negotiations. *Proceedings of the Sixth International Conference on E-Commerce*, pages 77-84, Delft, The Netherlands, 2004 **[Winner of best paper award at this conference]**.

- J. Shao, W. A. Gray, N. J. Fiddian, V. Deora, G. Shercliff, P. J. Stockreisser, T. J. Norman, A. Preece, P. M. D. Gray, S. Chalmers, N. Oren, N. R. Jennings, M. Luck, V. D. Dang, T. D. Nguyen, J. Patel, W. T. L. Teacy and S. Thompson (2004) Supporting Formation and Operation of Virtual Organisations in a Grid Environment. *Proceedings of The UK OST e-Science Second All Hands Meeting 2004 (AHM'04)*, ISBN 1-904425-21-6, Nottingham, UK, 2004.

- T. J. Norman, A. Preece, S. Chalmers, N. R. Jennings, M. Luck, V. D. Dang, T. D. Nguyen, V. Deora, J. Shao, A. Gray, and N. Fiddian. Agent-based formation of virtual organisations. Int. J. Knowledge Based Systems, 17(2-4):103-111, 2004.

- T. D. Nguyen and N. R. Jennings. Coordinating multiple concurrent negotiations. *In Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 1064-1071, New York, USA, 2004.

- T. D. Nguyen and N. R. Jennings. A heuristic model for concurrent bilateral negotiations in incomplete information settings. *Proceedings of the Eighteenth International Joint Conference on AI*, pages 1467-1469, Acapulco, Mexico, 2003.

- T. J. Norman, A. Preece, S. Chalmers, N. R. Jennings, M. Luck, V. D. Dang, T. D. Nguyen, V. Deora, J. Shao, A. Gray, and N. Fiddian. Conoise: Agent-based formation of virtual organisations. *Proceedings of the Twenty-third Annual International Conference of the British Computer Societys Specialist Group on Artificial Intelligence (SGAI)*, pages 353-366, Cambridge, UK, 2003 **[Best Paper Award]**.

- T. D. Nguyen and N. R. Jennings. Concurrent bilateral negotiation in agent systems. *Proceedings of the Fourth DEXA Workshop on e-Negotiations*, pages 839-844, Prague, Czech Republic, 2003.

1.5 Thesis Structure

The remainder of this thesis is structured in the following way:

- Chapter 2 investigates current work in the field of bilateral negotiation models, commitment models and applied learning in negotiation. Specifically, it discusses the requirements of the models, how they perform, and their relative advantages and disadvantages against the requirements identified in section 1.2.

- Chapter 3 presents the core negotiation model that has been developed in this research. This model uses a heuristic approach and is capable of handling multiple concurrent negotiations about a single service. The mechanisms and the components of the model are detailed and we discuss how it performs in specific situations. It is then evaluated by being applied in various negotiation environments. The results are compared against a sequential model and Rahwan et al's model to show its performance and efficiency.

- Chapter 4 details the commitment manager that has been incorporated into the negotiation model. This commitment manager enables the agents to back down from previously committed deals by paying a decommitment fee. We present various commitment tactics and strategies for the buyer agent to apply in specific settings. These strategies are then evaluated empirically and the results highlight which strategies are effective in which circumstances.

- Chapter 5 extends the negotiation model by introducing an adaptive negotiation strategy to help the buyer gain a better outcome in both one-off and repetitive negotiation scenarios. Specifically, if partial information about the participating agents is available, this strategy is able to exploit it to give the buyer additional bargaining power. Through empirically analysis, it is shown that the buyer can indeed obtain better value using this strategy than if it is not doing so.

- Chapter 6 highlights the main conclusions of this thesis and discusses the remaining open questions.

Chapter 2

Related Work

This chapter discusses the current state of the art in the fields of automated negotiation, commitment handling mechanisms and applied learning in negotiation. As noted in section 1.2, there are a number of different approaches to solve the negotiation problem. In the first part of this chapter, the main approaches are detailed. We start with an introduction to game theory in section 2.1, followed by discussions of computational heuristic-based models in section 2.2 and auction protocols in section 2.3. The second part of the chapter details the current existing commitment and applied learning models. First, the most dominant approaches to handling commitments among the agents, including the leveled commitment contracts, are discussed in section 2.4. Next, section 2.5 continues with a description of the different learning techniques that we believe can be useful in our negotiation model. Finally, section 2.6 concludes the chapter.

2.1 Game Theory

Game theory is a framework designed to model and analyze the decision making mechanism of independent entities in a common environment. Since its conception in [Neumann and Morgenstern, 1944], this framework has been used to study various social subjects such as economics and politics [Romp, 1997; Osborne and Rubinstein, 1994]. In recent years, game theory has been used to study interactions among intelligent agents, including negotiation [Johansson, 1999; Kraus, 2001]. To explore this area, this section first provides a general introduction to game theory and then goes on to discuss in detail its application in modeling automated negotiation.

In more detail, game theory models the decision making mechanism of different entities (which are *intelligent agents* in our case). These agents are assumed to be *rational* (which means that they *reason strategically* [Osborne and Rubinstein, 1994]) and they have *mutual interdependencies* among themselves [Romp, 1997]. Specifically, being *rational* means that the agents act in their own self-interest, being aware of their alternative options, forming expectations about the outcomes of their actions and having clear preferences over these outcomes. The assumption of *reasoning strategically* presupposes that when deliberating about their actions, the agents take their knowledge of expectations about the other agents' behaviors into account. The final assumption, *mutual interdependencies*, is present because game theory mostly considers situations in which the agents are mutually interdependent (i.e. the welfare of one agent is fully or partially determined by the other agents' actions. This, in turn, provides the incentive for each agent to reason strategically to find an optimal action, in order to achieve the most desirable outcome [Romp, 1997]).

Game theory uses mathematics to formally express its concepts and solutions. Each instance of a problem being considered is called a *game*. The basic elements of a *game* are *players*, *actions*, *strategies*, *information*, *pay-offs*, and *equilibria* [Kraus, 2001]. The objective of the game's modeler is to use the *rules of the game* (which are composed of *players*, *actions* and *outcomes*) to find the *equilibria*.

- *players*: the individual entities that make decisions (which are software agents in our case). In any game, there are always at least two players.
- *actions*: the available options for each player to make a choice from.
- *strategies*: a description of how each player can use the actions to play the game.
- *information*: what is available to each player before that player has to make a decision (e.g. the information about the environment or the other player's strategies).
- *pay-offs*: what a player will receive when the game terminates, given the actions of all the players in that game.
- *equilibria*: a set of strategies, one for each player, such that no player has incentive to unilaterally change its action. The concept of specific equilibria will be described later in this section.

Formally, there are two ways in which a game can be represented: *strategic game* and *extensive game*. A *strategic game* is defined as the tuple $\langle \mathrm{N}, (A_i), (\succsim_i) \rangle$. Here, N denotes the set of players. For each player $i \in \mathbf{N}$, $\mathbf{A}_i$ is the set of available actions for i. An action profile a is a set $a = \{a_i\}, i \in \mathbf{N}$, in which $a_i \in \mathbf{A}_i$ is the action of player i. The set of action profiles is called $\mathbf{A}$. The set of pay-offs is called $\mathbf{C}$. The link between $\mathbf{A}$ and $\mathbf{C}$ is given by a function $g : \mathbf{A} \rightarrow \mathbf{C}$, which associates an action profile with a pay-off; and a preference relation $\succsim_i^*$ over $\mathbf{C}$. For any two actions a_1, $a_2 \in \mathbf{A}_i$, the preference relation $\succsim_i$ of each player i over $\mathbf{A}_i$ is defined as follows: $a_1 \succsim_i a_2 \Leftrightarrow g(a_1) \succsim_i^* g(a_2)$. Typically, this preference relation $\succsim_i$ is represented by a *utility function* $u_i : \mathbf{A} \rightarrow \mathbb{R}$; $u_i(a_1) \geq u_i(a_2) \Leftrightarrow a_1 \succsim_i a_2$.

In a strategic game, an *equilibrium* is a solution (an action profile), in which none of the players acting individually, has an incentive to deviate from this solution. One of the most commonly used concepts is the Nash equilibrium and this is defined as follows [Osborne and Rubinstein, 1994]:

Definition 2.1. A Nash equilibrium of a strategic game $\langle \mathbf{N}, (A_i), (\succsim_i) \rangle$ is a profile $a^* \in \mathbf{A}$ with the property that for every player $i \in \mathbf{N}$ we have:

$$(a^*_{-i}, a^*_i) \succsim_i (a^*_{-i}, a_i) \quad \forall\, a_i \in \mathbf{A}_i.$$

where $a^*_{-i} = a^* \setminus a^*_i$. From this definition, it can be seen that for a^* to be a Nash equilibrium, each player i shall not be able to find another action that produces a better outcome than a^*_i, given that every other player j chooses a^*_j. This concept of Nash equilibrium captures a *steady state* of the playing of a strategic game, in which all the players act rationally and hold correct expectations about other players' behaviors [Osborne and Rubinstein, 1994].

To illustrate this concept, consider the classic example of the Prisoner's Dilemma, in which there are two suspects of a crime (players) being questioned in separate cells [Osborne and Rubinstein, 1994]. If both confess, each of them will be sentenced to three years. If both of them deny being involved, each of them will be convicted of a minor offense and be sentenced for one year. If only one of them confesses, he will not be charged but the other suspect will receive the sentence of four years. Table 2.1 represents the pay-offs of the players in this game (in each cell, the first number represents the column player's pay-offs and the second number represents the row player's pay-offs. The lower the pay-off, the better for the player).

As can be seen, the players will benefit best if they both deny (since the sum of their payoff is smallest). However, each of the players does not have the incentive to do so. Instead, no matter what the other player's chosen action, each player prefers *Confess* than *Deny* (for example, if the column player chooses *Deny*, the row player prefers

	Deny	Confess
Deny	1,1	0,4
Confess	4,0	3,3

TABLE 2.1: The pay-off matrix of the Prisoner's Dilemma.

Confess over *Deny* because the pay-off is better (0 compared to 1). Similarly, if the column player chooses *Confess*, the row player prefers *Confess* for the better pay-off (3 compared to 4). Consequently, the action profile (*Confess*, *Confess*) is the Nash equilibrium of this game. However, this outcome is not as good for either player as (*Deny*, *Deny*) – hence the dilemma.

However, the concept of a *strategic game* is too abstract to model problems when the timing of the decisions needs to be taken into account and when the amount of information available to each player is different at each period [Kraus, 2001]. For this reason, a more advanced form of game, called an *extensive game*, is introduced. This form of game explicitly describes the sequential decision problems for the players in a strategic situation. Unlike a *strategic game*, in which the players make their decisions only once, an *extensive game* requires each player to consider its action at multiple times during the course of the game.

Formally, an *extensive game* is represented by the tuple $\langle \mathrm{N}, \mathbf{H}, P, (\succsim_i) \rangle$. Here, N denotes the set of players involved, $\mathbf{H}$ the set of history sequences (it can be either finite or infinite). Each member $h \in \mathbf{H}$ is called a history, which is composed of the actions of the players: $h = (a^k), k \in [1, K]$. h is a *terminal* history if K is finite. The set of terminal histories is denoted by $\mathbf{Z}$. P is a function that maps a member $i \in \mathrm{N}$ to each non-terminal history. $\succsim_i$ is the preference relation for the player i over $\mathbf{Z}$. The initial history, $\varnothing$, is the starting point of the game.

At any non-terminal history $h \in \mathbf{H}$, $\mathbf{A}(h)$ denotes the set of available actions that player $P(h)$ can choose: $\mathbf{A}(h) = \{a : (h, a) \in \mathbf{H}\}$. A strategy s_i of player $i \in \mathrm{N}$ is a function that maps an action $a \in \mathbf{A}(h)$ to h for which $P(h) = i$. The set of strategies for player i is denoted by $\mathbf{S}_i$. A strategy profile s is the order set $s = (s_i) \forall i \in \mathrm{N}$ where s_i is the strategy of player i. Given a strategy profile s, the outcome $O(s)$ is defined as the terminal history that results when each player $i \in \mathrm{N}$ follows s_i. Based on these concepts, the definition of a Nash equilibrium is defined as:

Definition 2.2. A Nash equilibrium of an extensive game $\langle \mathrm{N}, \mathbf{H}, P, (\succsim_i) \rangle$ is a strategy profile s^* with the property that for every player $i \in \mathrm{N}$ we have:

$$O(s^*_{-i}, s^*_i) \succsim_i O(s^*_{-i}, s_i) \quad \forall\, s_i \in \mathbf{S}_i.$$

To illustrate the concept of Nash equilibrium in extensive game form, figure 2.1 presents a sample game [Romp, 1997]. Here, we have N $= \{1, 2\}$; $P(\varnothing) = 1$; $P(h) = 2$ for all non-terminal histories $h \neq \varnothing$; $S_1 = \{L, R\}$; $S_2 = \{L, R,$ do the same as 1, do the opposite of $1\}$. The pay-off matrix of this game is represented in table 2.2.

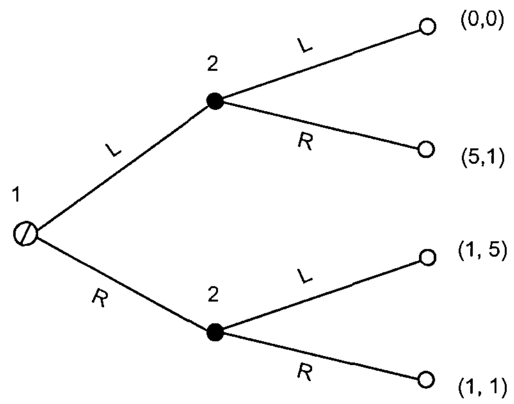


FIGURE 2.1: A sample game in the extensive form.

	L	R	do the same as 1	do the opposite of 1
L	0,0	5,1	0,0	5,1
R	1,5	1,1	1,1	1,5

TABLE 2.2: The pay-off matrix of the sample extensive game.

As can be seen, this sample game has three Nash equilibria:

1. player 2 threatens to always choose L, irrespective of player 1's action. If player 1 believes this threat, he will choose R.

2. player 2 promises to always choose R, irrespective of player 1's action. If player 1 believes this promise, he will choose L.

3. player 2 promises to always do the opposite of player 1's action. If player 1 believes this promise, he will choose L.

However, the first two equilibria are not plausible. In the first equilibrium, player 2 threatens to always choose L, irrespective of player 1's action. This is optimal only if player 1 chooses R. However, if player 1 chooses L, choosing L is not in the best interest of player 2 since its pay-off will be lower than if he chooses R. Therefore, this threat of player 2 is not credible to player 1. Similarly, the promise of player 2 in the second equilibrium is, also, not credible to player 1 (because if player 1 chooses R, choosing R is not optimal for player 2). Only the third equilibrium provides the optimal solution to both the players.

To overcome this problem, the concept of *subgame perfect equilibrium* or *SPE* is introduced. In order to explain this concept, we first have to define the concept of a subgame:

Definition 2.3. The subgame of the extensive game $\Gamma = \langle \mathrm{N}, \mathrm{H}, P, (\succsim_i) \rangle$ that follows the history $h \in \mathrm{H}$ is the extensive game $\Gamma(h) = \langle \mathrm{N}, \mathrm{H}|_{\mathrm{h}}, P|_h, (\succsim_{i|_h}) \rangle$ in which $\mathrm{H}|_{\mathrm{h}}$ is the set of sequences h' of actions for which $(h, h') \in \mathrm{H}$; $P|_h$ is defined by $P|_h(h') = P(h, h')$ for each $h' \in \mathrm{H}|_{\mathrm{h}}$ and $\succsim_{i|_h}$ is defined by $h' \succsim_{i|_h} h'' \Leftrightarrow (h, h') \succsim_i (h, h'')$ for each $h', h'' \in \mathrm{H}|_{\mathrm{h}}$.

Given this definition of a subgame, a *subgame perfect equilibrium* is defined as:

Definition 2.4. A SPE of an extensive game $\Gamma = \langle \mathrm{N}, \mathrm{H}, P, (\succsim_i) \rangle$ is a strategy profile s^* with the property that for every player $i \in \mathrm{N}$ we have:

$$O_h(s^*_{-i|_h}, s^*_{i|_h}) \succsim_{i|_h} O(s^*_{-i|_h}, s_i) \quad \forall\ s_i \in \mathbf{S}_i.$$

Basically, a subgame perfect equilibrium of an extensive game Γ is a Nash equilibrium s^* if and only if for each subgame $\Gamma(h)$, $s^*(h)$ is also a Nash equilibrium of that subgame. This concept of *subgame perfect equilibrium* eliminates the Nash equilibria that are not credible to the players. For example, the first two Nash equilibria of the extensive game described in figure 2.1 are not *subgame perfect equilibria* (e.g. (L,L) is not a Nash equilibrium in the former case and (R,R) is not a Nash equilibrium in the latter case). The only *subgame perfect equilibrium* of this game is that player 1 chooses L and player 2 chooses R. This is reasonable given the fact that player 1 is the first player to make the decision.

As previously mentioned, these discussions about game theory assume that all the players have complete information about the environment and the history of the game. However, these situations are not the only concern of game theory. In fact, game theory also studies situations in which the players' actions, as well as their planned future moves, are not revealed. The solution that game theory proposes for each player in this situation uses probability theory to form expectations of the other players' actions. Specifically, these expectations are used as the probability measures to help each player to select an appropriate strategy [Osborne and Rubinstein, 1994]. This type of incomplete information game is usually complex and the outcome depends very strongly on the specifics of the problem at hand. Thus, a detailed review of incomplete information games is not presented in this introduction (see [Fudenberg and Tirole, 1991; Gibbons, 1992] for more details of this type of game).

Another aspect of these discussions is that the players are considered to be individual self-interested entities, that try to maximize their own utility function. This is the characteristic of *non-cooperative game theory* [Romp, 1997]. The other type of game theory with respect to this characteristic is *cooperative game theory*, in which the players are allowed to enter into binding and enforceable agreements with one another [Osborne and Rubinstein, 1994]. In cooperative games, the players make binding agreements to coordinate their strategies, whereas in non-cooperative games the players use their strategies to maximize their own benefit, irrespective of the other players' actions. Therefore, in our opinion, *non-cooperative game theory* is a more suitable basis for studying realistic negotiation scenarios and is the approach that we focus on in this section.

Against this background, we present our view on some of the most popular negotiation models that uses game theory approaches.

2.1.1 Kraus's Negotiation Model

This section reviews Kraus's game theory based bilateral negotiation model [Kraus, 2001; Kraus and Schechter, 2003]. This model considers the situation in which there are two agents bargaining over the right to use a resource (e.g. printers or satellites). One agent, called the Attached Agent (A), is already using the resource that the other agent, called the Waiting Agent (W), needs. Now, W starts the negotiation process to obtain access to use the resource. If no agreement can be found, W opts out and causes some damage to the resource (such as destroying it), which will affect A. During the time of negotiation, A still uses the resource for its own purpose. The resource is composed of M units and the purpose of the negotiation is to find an agreement, which is an ordered pair (s_A, s_W) where $s_A + s_W = M$ and s_i is agent i's portion of the resource. The set of possible agreements is $\mathcal{S} = \{(s_A, s_W)|s_A, s_W \in \mathbb{N}^+, s_A + s_W = M\}$.

The negotiation protocol used in this model is based on Rubinstein's model of alternating offers [Rubinstein, 1982]. For each agent $i \in \{A, W\}$, i can only take actions in the negotiation at certain pre-determined times in the set $\mathcal{T} = \{0, 1, 2, ...\}$. At each time $t \in \mathcal{T}$, the agent whose turn it is to make a proposal at time t will propose its offer. The other agent can either accept the offer, reject it, or opt-out of the negotiation. If the proposed offer is accepted by the other agent, the negotiation terminates with an agreement. If the other agent opts out, the negotiation terminates with a conflict outcome (*Opt*). If the other agent has decided to reject the offer but not opt-out, the negotiation moves to the next time period, $t + 1$, where it is the turn of the other agent. If the negotiation continues forever without reaching an agreement and without any of the agents opting out, the outcome of this negotiation is called a *disagreement*. This protocol is called the *simultaneous response protocol alternating offers*.

In more detail, each agent $i \in \{A, W\}$ has a utility function over all possible outcomes U^i:

$$U^i : \{| \mathcal{S} \bigcup \{Opt\} \mid \times \mathcal{T}\} \bigcup \{Disagreement\} \rightarrow \mathbb{R}.$$

This utility function has the following assumptions:

1. **Disagreement is the worst outcome**: The agents prefer any possible outcome over disagreement.

$$\forall x \in \{| \mathcal{S} \bigcup \{Opt\} \mid \times \mathcal{T}\}, i \in \{A, W\} : U^i(Disagreement) < U^i(x).$$

2. **The resource is valuable**: For agreements that are reached within the same time period, each agent prefers to get a larger portion of the resource.

$$\forall t \in \mathcal{T}; r, s \in \mathcal{S}; i \in \{A, W\} : r_i > s_i \Rightarrow U^i((r, t)) > U^i((s, t)).$$

3. **Cost/benefits over time**: It is assumed that the agents have a utility function with a constant cost or gain due to delay. At each period, A has a constant gain $c_A > 0$ and W has a constant loss $c_W < 0$.

$$\forall t_1, t_2 \in \mathcal{T}; s \in \mathcal{S} : t_1 < t_2 \Rightarrow \begin{cases} U^W((s, t_1)) \geq U^W((s, t_2)) \\ U^A((s, t_1)) \leq U^A((s, t_2)) \end{cases}$$

4. **Agreement costs over time**: It is assumed that W prefers to obtain the resource sooner rather than later, whereas A prefers to obtain the resource later rather than sooner.

5. **Cost of opting out over time**: At time $t \in \mathcal{T}$, if there is an agreement $s \in \mathcal{S}$ that A prefers over W's opting out in the next period $t + 1$, A may agree to s. This agreement is denoted $\hat{s}^{W,t}$. The set of agreements that are not worse for any agent than opting out is denoted $Possible^t$.

$$\forall t \in \mathcal{T} : \begin{cases} U^W((Opt, t)) > U^W((Opt, t + 1)) \\ U^A((Opt, t)) < U^A((Opt, t + 1)) \end{cases}$$

6. **Range for agreement**: For every $t \in \mathcal{T}$,

   - The property of the non-emptiness of $Possible^t$ is monotonic; thus, if $Possible^{t+1} \neq \varnothing$, then $Possible^t \neq \varnothing$.
   - If it is still possible to reach an agreement in the next time period, W prefers to opt-out at period t or agree to $\hat{s}^{W,t}$ than to wait until the next time period $t+1$ and agree to $\hat{s}^{W,t+1}$. However, A's preference is reversed and A prefers to agree to $\hat{s}^{W,t+1}$ at period $t+1$ than to agree to $\hat{s}^{W,t}$ at period t.

$$\text{If } Possible^t \neq \varnothing \text{ then } \begin{cases} U^W((\hat{s}^{W,t}, t)) \geq U^W((\hat{s}^{W,t+1}, t+1)) \\ U^W((Opt, t)) \geq U^W((\hat{s}^{W,t+1}, t+1)) \\ U^A((\hat{s}^{W,t+1}, t+1)) \geq U^A((\hat{s}^{W,t}, t)) \end{cases}$$

   - If it is still possible to reach an agreement then A prefers $\hat{s}^{W,t}$ at period t to opting out in the next period $t+1$:

$$\text{If } Possible^t \neq \varnothing \text{ then } U^A((\hat{s}^{W,t}, t)) \geq U^A((Opt, t+1)).$$

7. **Possible agreement**: In the first two time periods, 0 and 1, there is an agreement that is preferable to both A and W over opting out: $Possible^0 \neq \varnothing$ and $Possible^1 \neq \varnothing$.

The model considers two situations: *complete* and *incomplete* information games. In the first situation, the agent has complete information about the other agent including its utility function. If the agent uses *subgame perfect equlibrium* strategies and the aforementioned assumptions are satisfied then *an agreement will be reached in the first or second period.* This agreement is Pareto optimal[1]:

- **W loses more than A can gain**: If $|c_W| > c_A$ and $\hat{s}_A^{W,1} + c_A \leq M$ then W will offer $(\hat{s}_A^{W,1} + c_A, \hat{s}_W^{W,1} - c_A)$ in the period 0 of the negotiation and A will accept this offer.

- **W loses less than A can gain**: If $|c_W| < c_A$, any offer made by W at the first time period will be rejected by A. In the next time period, A will make a counter-offer $(\hat{s}_A^{W,1}, \hat{s}_W^{W,1})$ and W will accept this counter-offer.

In the *incomplete* information game, the agents do not have information about the other agent's utility function. To solve this problem, one more assumption is made, namely

[1]A solution is Pareto optimal if there is no other solution that is better for one agent and not worse for the other [Sandholm, 1999b].

that there is a finite set of possible types of agent, where each type defines the utility function of the agent. Each agent only belongs to a specific type. However, W and A do not know the type of each other. Instead, each of them has a belief system, which is a probability distribution, about the type of the opponent. Similar to the complete information game, if the agents use *subgame perfect equlibrium* strategies and the aforementioned assumptions are satisfied then *the negotiation ends in the second period*. However, unlike the complete information game, an agreement is not guaranteed. Thus, there is a possibility that W will opt-out if A's offer does not meet W's demand.

Against the requirements stated in section 1.2, this model is:

- *computationally tractable*: since the negotiations in both the complete and incomplete information games finish within two negotiation periods and the offer generation mechanism is clearly defined and requires simple calculation, this model satisfies this criterion.
- *incomplete information*: this model performs best with complete information games (as would be expected). However, the situation is not the same in incomplete information games. To be able to cope with this specific game type, the model puts forward the assumption about the agents' types and the probability distribution of this type of the agent. When these assumptions are satisfied, the model guarantees to terminate within two negotiation periods but is not able to guarantee that the agreement reached is an equilibrium.
- *partial information*: this model exploits the information about the opponent in searching for offers.
- *negotiation deadline*: because the negotiation will always terminate after at most two negotiation periods, this model is able to satisfy this requirement.
- *concurrent negotiations*: this model is not designed for and, thus, is incapable of handling multiple concurrent negotiations.
- *efficient outcomes*: in the complete information game, the model guarantees to provide a Pareto optimal solution. However, this is not the case in the incomplete information game, in which this claim cannot be guaranteed in all situations.
- *commitments*: this model is not designed for and, thus, is incapable of handling commitments amongst the participating agents.

As can be seen, the most important disadvantage of this model is that it has a number of unrealistic assumptions. In the complete information negotiation game, the agents must know everything about the other agent including its utility function. This situation is not common in typical negotiation scenarios. In the incomplete information negotiation game, the possible types of agent are limited, which leads to an approximation of the complete information case. This assumption is the core requirement and the model is not able to perform without it. As a result, this model is not applicable in practical situations.

The second disadvantage of this model is the obsolete negotiation process. The negotiation always terminates after a fixed number of rounds and the result can be determined even before the negotiation starts. Consequently, the whole negotiation process could be efficiency reduced to a simple choice functions with both agents having no incentive to deviate from the chosen result.

2.1.2 Fatima's Negotiation Model

Fatima et al. define a bilateral negotiation model that focuses on a single-issue item (price) [Fatima *et al.*, 2001]. The two agent participants in this model are assumed to have fixed negotiation deadlines and they do not have complete information about each other. However, this model does assume that each agent has partial information about the opponent (this information is private and only available to it), which includes:

- a finite set of possible values for the reservation price of the opponent and a binary probability distribution over these values.
- a finite set of possible values for the negotiation deadline of the opponent and a binary probability distribution over these values.

Each agent does not have information about what type of strategy the opponent uses in the negotiation. Under such uncertainty, the aim of this model is to determine how each agent can exploit the partial information to select the best strategy that will maximize its utility.

Formally, one of the two agents plays the role of a buyer (b) whereas the other plays the role of a seller (s). b prefers to pay less for the good whereas s prefers to sell the good for a high price. The negotiation deadlines for b and s are T^b and T^s, respectively. The range of prices that are acceptable for an agent a is denoted as $[P^a_{min}, P^a_{max}]$. There exists an interval, called *price-surplus*, which is $[P^s_{min}, P^b_{max}]$ that contains the prices that are acceptable to both b and s. At time t of the negotiation, $p^t_{b\rightarrow s}$ denotes the price offered by b to s. When s receives this offer, it uses its utility function U^s to evaluate the offer. If $U^s(p^t_{b\rightarrow s})$ is greater than the value of the counter-offer $p^{t'}_{s\rightarrow b}$ that s is ready to send to b at time $t' > t$ then s will accept this offer. Otherwise, s will send its counter-offer $p^{t'}_{s\rightarrow b}$ to b. The action A that s will take at time t is defined as:

$$A^s(t', p^t_{b\rightarrow s}) = \begin{cases} \mathit{Quit} & \text{if } t > T^s \\ \mathit{Accept} & \text{if } U^s(p^t_{b\rightarrow s}) > U^s(p^{t'}_{s\rightarrow b}) \\ p^{t'}_{s\rightarrow b} & \text{otherwise} \end{cases}$$

For each agent a, at time $t < T^a$, the value of the offer/counter-offer that a generates is calculated using equation 2.1:

$$p^t_{a\rightarrow b} = \begin{cases} P^a_{min} + F^a(t)(P^a_{max} - P^a_{min}) & \text{for the buyer} \\ P^a_{min} + (1 - F^a(t))(P^a_{max} - P^a_{min}) & \text{for the seller} \end{cases} \quad (2.1)$$

where $F^a(t)$ is the $\alpha^a_j(t)$ function in Faratin's model (see section 2.2.1). The strategy for each agent a is a tuple $\langle P^a_{max}, P^a_{min}, F^a(t), T^a \rangle$.

Given this information, each negotiation environment for b is defined as the tuple[2]:

$$\langle \mathcal{T}^s, \alpha^s, \alpha^s_c, \mathcal{P}^s, \beta^s, \beta^s_c, T^b, P^b, U^b \rangle$$

where:

- $\mathcal{T}^s = \{T^s_1, T^s_2 | T^s_1 < T^s_2\}$ is a two element vector that contains possible values for s's deadline.
- α^s is the probability that s's deadline is T^s_1. Thus, $1 - \alpha^s$ is the probability that s's deadline is T^s_2.
- α^s_c is the value of α^s on the basis of which the buyer selects an optimal strategy.
- $\mathcal{P}^s = \{P^s_1, P^s_2 | P^s_1 < P^s_2\}$ is a two element vector that contains possible values for s's reservation price.

[2]The environments for s can be defined analogously.

- β^s is the probability that s's reservation price is P_1^s. Thus, $1-\beta^s$ is the probability that s's reservation price is P_2^s.
- β_c^s is the value of β^s on the basis of which the buyer selects an optimal strategy.
- T^b, P^b and U^b are the buyers deadline, reservation price and utility function, respectively.

$\mathcal{T}^s, \alpha^s, \mathcal{P}^s$ and β^s are the private information that b has about s. There are 6 possible environments that are defined based on the values of T_1^s, T_2^s, T^b and U^b:

- E_1^b: when $T_2^s < T^b$ and b gains utility with time (U^b is an increasing function of time).
- E_2^b: when $T_1^s < T^b < T_2^s$ and b gains utility with time.
- E_3^b: when $T^b < T_1^s$ and b gains utility with time.
- E_1^b: when $T_2^s < T^b$ and b loses utility with time (U^b is a decreasing function of time).
- E_2^b: when $T_1^s < T^b < T_2^s$ and b loses utility with time.
- E_3^b: when $T^b < T_1^s$ and b loses utility with time.

For each agent a, a strategy S^a is a function that maps its negotiation environment E^a and time T to the counter-offer vector at time $T+1$ (see equation 2.1). Assume O is the outcome that has resulted from S^a then S^a is the optimal strategy for a if it maximizes $U^a(O)$.

Given each environment E_i^b for b, the aim of the model is to find the corresponding optimal strategy S_i^a. There are two situations:

- When $\beta^s = 1$:
  - E_1^b: There are two possible strategies:

    $s_1^b = \langle P_{min}^b, P_1^s, \text{Boulware}^3, T_1^s \rangle$ if $T \leq T_1^s$

    $s_2^b = \langle P_{min}^b, P_1^s, \text{Boulware}, T_2^s \rangle \forall T$.

[3]This tactic starts with a high value offer and maintains its value until the time is almost exhausted, whereupon it concedes up to the reservation value. For more information on Boulware tactics, see section 2.2.1.

Let α_c^s denote the value of α^s below which s_2^b is better than s_1^b and above which s_1^b is better than s_2^b. α_c^s depends on θ_s^t, which is the length of the time interval between T_1^s and T_2^s : $\theta_s^t = T_2^s - T_1^s$.

In this environment, the optimal strategy S_1^b is s_1^b when $\alpha^s > \alpha_c^s$ and s_2^b otherwise.

- E_2^b: For all values of α^s, the optimal strategy S_2^b is:

$$S_2^b = \begin{cases} \langle P_{min}^b, P_1^s, \text{Boulware}, T_1^s \rangle & \text{if } T \leq T_1^s \\ \langle P_1^s, P^b, \text{Boulware}, (T^b - T_1^s) \rangle & \text{otherwise} \end{cases}$$

- E_3^b: For all values of T, the optimal strategy S_3^b is $\langle P_{min}^b, P^b, \text{Boulware}, T^b \rangle$.
- E_4^b: For all values of T, the optimal strategy S_4^b is $\langle P_{min}^b, P_1^s, \text{Conceder}^4, T_2^s \rangle$.
- E_5^b: The optimal strategy S_5^b is:

$$S_5^b = \begin{cases} \langle P_{min}^b, P_1^s, \text{Conceder}, T_1^s \rangle & \text{if } T \leq T_1^s \\ \langle P_1^s, P^b, \text{Conceder}, (T^b - T_1^s) \rangle & \text{otherwise} \end{cases}$$

- E_6^b: For all values of T, the optimal strategy S_6^b is $\langle P_{min}^b, P^b, \text{Conceder}, T^b \rangle$.

• When $\beta^s \neq 1$:

There are two possible strategies:

$$s_1^b = \langle P_{min}^b, P_1^s, \text{Boulware / Conceder}, T \rangle$$
$$s_2^b = \langle P_{min}^b, P_2^s, \text{Boulware / Conceder}, T \rangle$$

Let β_c^s denote the value of β^s below which s_2^b is better than s_1^b and above which s_1^b is better than s_2^b. β_c^s depends on θ_s^p, which is the difference between P_1^s and P_2^s : $\theta_s^p = P_2^s - P_1^s$.

The optimal strategy S^b is s_1^b when $\beta^s > \beta_c^s$ and s_2^b otherwise.

As can be seen, all the possible negotiation environments can be classified into one of the above situations. After the environment has been classified, the optimal negotiation strategy is chosen accordingly.

Now, against the requirements stated in section 1.2, this model is:

[4]This tactic starts with a high value offer and then quickly concedes up to the reservation value (see section 2.2.1 for more details).

- *computationally tractable*: the optimal strategy specified in any environment is a time-dependant tactic from the responsive mechanism in Faratin's model (see section 2.2.1). Consequently, this requirement is satisfied.

- *incomplete information*: this model copes with incomplete information situations by making assumptions about the opponent's characteristics (the value of the opponent's reservation price and negotiation deadline). If these assumptions are satisfied, this model is able to find the optimal strategy.

- *partial information*: this model exploits the information about the opponent in searching for offers.

- *negotiation deadline*: this model satisfies this requirement because the participating agents all have deadlines.

- *concurrent negotiations*: this model is not designed for and, thus, is incapable of handling multiple concurrent negotiations.

- *efficient outcomes*: this model is able to find the optimal strategy in any environment. Consequently the agreement, once reached, is optimal for both agents.

- *commitments*: this model is not designed for and, thus, is incapable of handling commitments amongst the participating agents.

This model presents a clear evaluation of different situations that could happen in a negotiation episode. Each situation is carefully analyzed to find the optimal negotiation strategy for the buyer agent. However, this model has a number of disadvantages. Firstly, it is assumed that there are a fixed number of different seller's types (in terms of their deadlines and reservation prices). This assumption is the basic requirement, without which the model is not capable of performing at all. Realistically, this assumption is not plausible: not every negotiation episode allows the buyer to acquire the information about the opponent. Thus, to be widely applicable, the model needs to be able to perform without relying on this assumption. Secondly, the assumption that this model makes is that there are only two types of sellers. There are six corresponding environments to be analyzed. Consequently, if the number of seller types is increased to n, the number of corresponding environments to be analyzed will be $2(n+1)$ and the analyzing spaces will be dramatically increased. This increase will eventually reduce the practicability of the model, especially when the number of sellers is increased.

2.1.3 Appraisal of Game Theory

Generally speaking, game theory techniques have been applied in various social sciences and have proved to be successful in modeling social phenomena. This success can be explained by: (1) game theory's capability of conceptualizing a synthetic and formal prototypical context as a game which is open to experimental analysis and (2) its ability to predict and explain these games in a manner, using formal notions [Castelfranchi and Conti, 1998]. Furthermore, game theory has introduced new concepts such as Pareto optimality and Nash equilibrium that can be used in evaluating the efficiency of other negotiation models.

However, when applied in the automated negotiation domain, the game theory approach has a number of limitations [Jennings *et al.*, 2001]:

- Game theory fails to generate a general model that can govern rational choice in inter-dependent situations [Zeng and Sycara, 1997]. Instead, game theory only produces a number of highly specialized models that are only applicable to specific types of inter-dependant decision making. As noted by Binmore: "...conclusions (of non-cooperative models) only apply to one specific game. If the details of the rules are changed slightly, the conclusions reached need no longer be valid" [Binmore, 1992], p.196.

- Most of game theory based models often have the assumption that the agents have *perfect computational rationality*. This means that it is assumed the mutually acceptable solutions within a feasible range of outcomes can be found without any computational requirement. This assumption is not plausible in most real world cases. Knowing that a solution exists does not imply that it can be found. Although this notion of perfect rationality is helpful in designing and proving properties of a negotiation system, it is not useful in practice. As noted by Sandholm: "... future work should focus on developing methods where the cost of search (deliberation) for solutions is explicit, and it is decision-theoretically traded off against the bargaining gains that the search provides. This becomes particularly important as the bargaining techniques are scaled up to combinatorial problems with a multi-dimensional negotiation space as opposed to combinatorially simple ones like splitting the dollar" [Sandholm, 1999a], p.223.

- Similarly, most of the game theory based models only prove that they are useful if the agents have a *perfect knowledge* about the game, including the information about the opponent's strategies and preferences. However, again this assumption does not hold in most real world cases. Typically, the agents are self-interested and only act on their interests. Thus, they will keep their strategies and preferences as private information, in order to avoid being exploited by the opponent. Given this condition, most of the game theory based models fail to operate or the optimality of the outcome is not guaranteed.

Even though game theory based techniques are a useful tool in analyzing a negotiation model, especially when the preferences and possible strategies of the participants can be characterized [Jennings *et al.*, 2001], we do not adopt this approach in our work because of these limitations.

2.2 Heuristic-Based Bilateral Negotiation Models

This section focuses on models in which the participating agents use some form of heuristic search to find the offers or counter-offers to propose to one another. Unlike the assumptions in game theory, these agents do not have perfect rationality; nevertheless, they still try to maximize their benefits, but only within their capabilities. These models generally impose fewer restrictions than the game theory based ones and focus more on the deliberation mechanism of the agents. In this section, we review three of the most prominent models: (1) Faratin's model, which uses heuristic as the search technique, (2) Luo et al's model, which uses Constraint Satisfaction (CSP) as the search technique and (3) Rahwan et al's model, which also uses CSP as the search technique and has the capability of handling concurrent negotiations.

2.2.1 Faratin's Negotiation Model

Faratin et al. develops a model for service-oriented bilateral negotiation [Faratin, 2001; Sierra *et al.*, 2002]. Here, a service has the same intuitive meaning as the one we adopted in this work (see section 1.1). This negotiation model is influenced by two application domains: business process management (ADEPT) [Jennings *et al.*, 1998] and telecommunication service management [Faratin *et al.*, 2000].

In this model, the *negotiation protocol* is a modified version of Rubinstein's model of alternating offers (see section 2.1.1) that allows the agents to iteratively exchange proposals. The protocol terminates either successfully (if both parties agree on a specific solution) or unsuccessfully (if one of the agent reaches its negotiation deadline).

In more detail, each agent i can take one of two possible roles: a *client* (c) or a *server* (s) and i has the deadline t^i_{max} beyond which it cannot continue its negotiation. The object that the agents bargain over is referred to as a *contract*, which is composed of n issues. For each agent i, each issue j is a tuple $\langle D^i_j, w^i_j \rangle$, where $D^i_j = \{min^i_j, max^i_j\}$ is the interval of possible quantitative values for each issue and w^i_j is the weight of this issue, or how i values j. V^i_j is a scoring function $V^i_j : D^i_j \rightarrow [0, 1]$ that gives a value to an issue j. The scoring function of a contract $x = (x_1, \ldots, x_n)$ is then defined as:

$$V^i(x) = \sum_{1 \leq j \leq n} w^i_j V^i_j(x_j).$$

Normally, c and s have opposing interests (e.g. c wants to have a low price for a service, whereas s wants to obtain a high price for it). This typically leads to opposing scoring functions for c and s: given an issue j, if $x_j \geq y_j$ then $V^c_j(x_j) \geq V^c_j(y_j) \Leftrightarrow V^s_j(x_j) \leq V^s_j(y_j)$.

Once the set of negotiation issues is set, the negotiation process starts with an alternate succession of offers and counter-offers of values for these issues. The contract proposed by an agent a to the other agent b at time t is denoted as $x^t_{a \rightarrow b}$ and the value of issue j proposed from a to b at time t is denoted as $x^t_{a \rightarrow b}[j]$. For convenience, the model assumes that there exists a linear set of instances of a common global time and there is no delay in message transmission during negotiation.

Given this background, Faratin introduces the *responsive mechanism* that aids the agents in deliberating during the negotiation process. Specifically, this mechanism helps the agents to decide:

- what initial offers should be sent out
- what is the range of acceptable agreements
- what counter offers should be generated
- when negotiation should be abandoned
- when an agreement is reached

In more detail, the responsive mechanism generates offers and counter-offers using linear combinations of simple functions (called *tactics*). Tactics generate an offer for a single negotiation issue using a single criterion (such as time or resource available to the agents) which is motivated by the computational and informational bounds of the agents. When a receives an offer $x^t_{b\to a}$ from b at time t, a evaluates this offer using its scoring function. If the value of this offer is greater than the value of the counter-offer $x^{t+1}_{a\to b}$ that a is ready to send to b ($V^a(x^t_{b\to a}) > V^a(x^{t+1}_{a\to b})$), then a will accept this offer. Otherwise, a will send its counter-offer $x^{t+1}_{a\to b}$ to b.

For an agent to generate the counter-offer, Faratin introduces three independent sets of tactics:

1. *Time dependent*: these tactics (see figure 2.2) reflect the behavior of the agents with regard to their deadlines. Each tactic in this set is differentiated by the shape of the concession curve which is a function depending on time. For an agent a at time t, the value of the issue j that a will propose to b is calculated as:

$$x^t_{a\to b}[j] = \begin{cases} min^a_j + \alpha^a_j(t)(max^a_j - min^a_j) & \text{if } V^a_j \text{ is decreasing} \\ min^a_j + (1 - \alpha^a_j(t))(max^a_j - min^a_j) & \text{if } V^a_j \text{ is increasing} \end{cases}$$

   where $\alpha^a_j(t)$ is a function whose value is dependant on the relation between t and t^a_{max}. Two families of $\alpha^a_j(t)$ function are proposed:

   - **polynomial**: $\alpha^a_j(t) = \kappa^a_j + (1 - \kappa^a_j)(\frac{min(t, t^a_{max})}{t^a_{max}})^{\frac{1}{\beta}}$
   - **exponential**: $\alpha^a_j(t) = e^{(1 - \frac{min(t, t^a_{max})}{t^a_{max}})^{\beta} ln\kappa^a_j}$

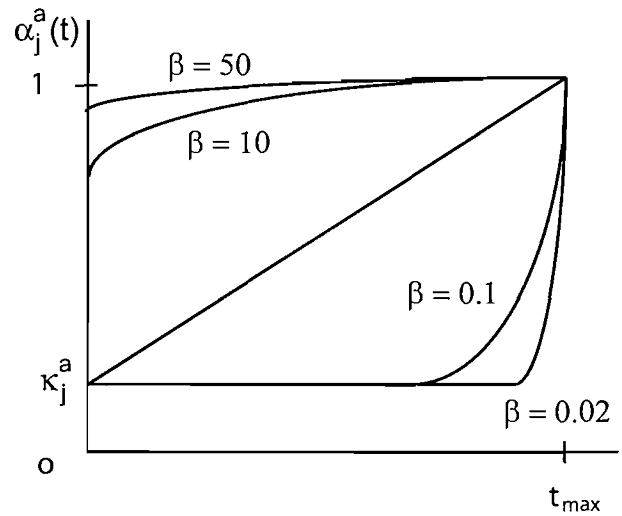


FIGURE 2.2: The time dependent tactics.

where κ_j^a is the constant that when multiplied by the size of the interval, determines the value of the issue j in the initial proposal and β is the value that determines the convexity degree of the curve. If $\beta < 1$, these tactics are related to Boulware tactics [Raiffa, 1982], which maintain a low value until the deadline approaches, whereupon the value will be conceded up to the reservation value. In contrast, if $\beta > 1$, these tactics are related to Conceder tactics [Pruitt, 1981], which quickly concede to their reservation value.

2. *Resource dependent*: these tactics reflect the behavior of the agents with regard to the resource that is available to them at time t of the negotiation. The resource could be money, the number of agents interested in the negotiation or time. These tactics use a similar formula to the *time dependent* ones but with a different function for $\alpha_j^a(t)$:

$$\alpha_j^a(t) = \kappa_j^a + (1 - \kappa_j^a)e^{-resource^a(t)}$$

where $resource^a(t)$ is the function that measures the quantity of the resource at time t for a. For example, if the resource is time then $resource^a(t) = min(0, t - t_{max}^a)$.

3. *Behavior dependent*: these tactics generate counter-offers based on the previous offers from the opponent. There are three different families of these tactics: *relative tit-for-tat*, *random absolute tit-for-tat* and *average tit-for-tat*. These tactics are designed to empirically evaluate the relative success of the agents when imitating the behaviors of the opponents.

As can be seen, this negotiation model focuses on the *process* of the negotiation, unlike the models based on game theory that focus on the *outcome* of the negotiation. Consequently, this model is applicable for designing autonomous agents that can negotiate.

The time dependent tactics introduced in the responsive mechanism are functions that have their values defined based on the relation between the negotiation time and the deadline of an agent. Different emphasizes can be placed on different time periods (e.g. Boulware tactics keep the value high up until the deadline approaches while Conceder tactics quickly concede to theirs reservation value). Moreover, the computational complexity of these tactics is polynomial. As a result, these functions are chosen as the basic strategies in our negotiation model described in chapter 3.

Against the requirements stated in section 1.2, this model is:

- *computationally tractable*: all the mechanisms proposed in this model are designed to be computationally tractable. Specifically, the responsive mechanism uses heuristic search to find the offer and this method requires very little computation.
- *incomplete information*: this model requires no specific information about the opponent and makes no special assumptions about the agents' capabilities. Thus, it is applicable in the incomplete information setting.
- *partial information*: this model does not exploit the information about the opponent even when it is available.
- *negotiation deadline*: this model satisfies this requirement because the participating agents all have deadlines.
- *concurrent negotiations*: this model is not designed for and, thus, is incapable of handling multiple concurrent negotiations.
- *efficient outcomes*: since the responsive mechanism uses heuristic techniques to search for offers, the agreement reached is not guaranteed to be optimal.
- *commitments*: this model is not designed for and, thus, is incapable of handling commitments amongst the participating agents.

Although this model is well developed and evaluated, there are still some points for further improvement. Firstly, this model is naturally designed for handling bilateral negotiation and, thus, is incapable of handling multiple concurrent negotiations. However, the mechanisms and tactics presented are effective and have polynomial complexity, thus they can be used in each bilateral negotiation context in our model. Moreover, no commitment handling mechanism is implemented in this model (since it is focused on the standard bilateral case, the final agreement once reached is binding on both agents). Finally, the agent in this model can perform without any particular information about the opponent. However, in many of the typical negotiation scenarios, partial information about the opponent is available and can be exploited to lead to more beneficial agreements. This model lacks the ability to exploit such information.

2.2.2 Luo's Negotiation Model

[Luo *et al.*, 2003] introduce a model that focuses on bilateral multi-issue negotiation. The agents that participate in this model are the buyer (b) and the seller (s). The approach that this model adopts is based on the Prioritized Fuzzy Constraint Satisfaction

Problems (PFCSP) [Dubois *et al.*, 1994; Dubois and Prade, 1999]. This technique helps the agents to decide whether to accept an offer proposed from the opponent and, if not, what counter-offer to send back. This technique is chosen for a number of reasons:

- Typically, the preferences of b can only be vaguely stated (e.g. when b goes looking for a car, he may not have a clear idea about the specific car that he wants. Instead, b may only have vague preferences about the car such as it must be cheap, economical and safe). Fuzzy constraints are well suited to describe these types of preferences.
- When the offers are exchanged between the agents, it is not typical for an offer to be completely acceptable or completely unacceptable with regard to one agent's constraints. Rather, an offer is likely to satisfy an agent's constraints to some degree. The PFCSP framework is suitable for capturing this partial satisfaction because fuzzy constraints can be satisfied or violated partially.
- For each single attribute of the negotiation subject, fuzzy constraints allow b to express its preferences over different values. Similarly, fuzzy constraints allow b to express preferences over different combinations of the attributes of the negotiation subject.
- Fuzzy constraints can be used to model b's preferences on trade-offs between different attributes of the negotiation subject.
- Not all the constraints have the same level of importance. For example, one might be more important than the others. By introducing priority into fuzzy constraints, the different levels of importance can be captured.

The negotiation protocol used in this model is that of alternating offers (see section 2.1.1) with an extension to have the capability of offering rewards. The agents in this model exchange proposals found in their search spaces, until either the proposal is accepted by both of the agents or the search space of one agent becomes empty. Specifically, b uses PFCSP to represents its preferences and s has a finite set of available products, where each product has certain associated restrictions (the buyer must satisfy these restrictions if it wants to get hold of this product) and rewards (the buyer will have some extra benefit if it decides to take this product). Each agent has an individual set of possible primitive actions that are available during negotiation.

Specifically, at its turn, b submits a set of constraints to s and asks s to do one of the following:

- *find*: find a product that satisfies the set of submitted constraints.
- *refind*: find an alternative product that satisfies the submitted set of constraints. This is sent when b is not happy with the previous product offered by s.
- *deal*: terminate the negotiation with an agreement. This is sent when b is happy with the previous product offered by s.
- *fail*: terminate the negotiation without an agreement. This is sent when s cannot offer any product that satisfies the submitted set of constraints and b is unable to relax any of its constraints.

With regard to s, when s receives the set of constraints and the requested action from b, it tries to do as b requests. The basic assumption is that s will try to obtain the best deal that is possible. If b is not satisfied with the previous offer of s, s tries to find a trade-off with the same profit to send to b. Only when s is unable to find this trade-off does it make some concessions (find an alternative product with less profit or offer some reward). Now, s always wants to achieve an agreement since it can only obtain a profit if an agreement is made. Therefore, when b does not accept its offer (even with a reward) and an alternative product could not be found, s will ask b to relax its constraints.

Against the requirements stated in section 1.2, this model is:

- *computationally tractable*: this model is CSP based and, thus, uses CSP techniques to explore the search space to find the optimal solution. Although most of the search algorithms guarantee to find optimal solutions, they do not take the computational requirement into account [Kowalczyk and Bui, 2001]. On the other hand, search algorithms that are computationally tractable cannot guarantee the optimal result [Yokoo and Ishida, 1999].
- *incomplete information*: this model requires no specific information about the opponent and makes no special assumptions about the agents' capabilities. Thus, it is applicable in the incomplete information setting.
- *partial information*: this model does not exploit the information about the opponent even when it is available.
- *negotiation deadline*: the negotiation in this model terminates only if the search space of one agent becomes empty. This is not guaranteed to happen and so this requirement is not satisfied.

- *concurrent negotiations*: this model is not designed for and, thus, is incapable of handling multiple concurrent negotiations.
- *efficient outcomes*: this model is guaranteed to find a Pareto optimal solution when it is available.
- *commitments*: this model is not designed for and, thus, is incapable of handling commitments amongst the participating agents.

A strong point about this model is that by using fuzzy constraints, human preferences and desires can be represented naturally. Furthermore, the fact that each proposal more or less satisfies the requirements of the agents gives the model the flexibility that is not available in others models (such as those based on game theory).

However, the most noticeable disadvantage of the model is the computational requirement of the search algorithm as previously mentioned. In order to cope with this problem, this model assumes that the space of possible agreements is limited. If the size of this space is increased, the computational complexity of the search algorithm is exponentially increased. This assumption reduces the applicability of the model in practical negotiation situations. Furthermore, even though the model is able to handle multiple issues, the seller agent is not able to vary the values of the issues attached to a proposal. Thus, the seller, when faced with a refusal from the buyer, has to find a completely new offer to send to the buyer instead of modifying its current offer. This limitation greatly reduces the potential search space, which, in turn, results in less beneficial agreements.

2.2.3 Rahwan's Negotiation Model

[Rahwan *et al.*, 2002] introduce a negotiation model that deals with multiple concurrent bilateral negotiations. Each individual negotiation thread uses a Constraint Satisfaction Problem (CSP) technique to search for possible agreements. Specifically, there is an agent, called the buyer b, that wants to negotiate with a number of different agents, called the sellers, in order to find the best possible deal for a product. For each seller s, b creates a single bilateral negotiation agent, called the *sub-negotiator* c, to negotiate with s. Each *sub-negotiator* shares the same preferences with b but can have different possible negotiation strategies. After all the *sub-negotiators* are created, b acts as a central agent that will coordinate the actions of *sub-negotiators*.

In a specific negotiation between c and s, both agents use constraints to represent their preferences. Each agent iteratively searches for a prospective solution from its solution space using constraint based reasoning (constraint propagation) techniques. These techniques involve the reduction of the search space into one that contains feasible solutions that satisfy the other agent's proposed constraints. A specific utility value is then selected and, finally, the constraint propagation technique is used to determine the values of the attributes of the product. The selection of the utility value is specified by a negotiation strategy. In this model, the strategy imposed is a simple one that has its value either staying the same (trade-off) or being reduced by a constant amount (concession). The trade-off option will be selected only if there is more than one instance of the set of attribute values for the same utility value.

The coordination of the *sub-negotiators* is done via the means of the *coordination strategy*. Thus, during negotiations, each *sub-negotiator* reports its status to b. After receiving all the information, b assesses the situation and issues instructions to the *sub-negotiators* accordingly. There are three *coordination strategies* available:

- *desperate strategy*: b uses this strategy if it wants to close a deal as soon as possible. As soon as a *sub-negotiator* reaches a deal with a particular seller, b closes all the negotiations with the other sellers. If there are multiple deals reached, the one with the highest utility value will be chosen.
- *patient strategy*: b uses this strategy if b has no time constraints. All the negotiations will be carried out until they all finish. At that point, the seller that offers the deal with the highest utility value will be chosen.
- *optimized patient strategy*: this strategy is similar to the *patient strategy*. However, if there is a deal made from a negotiation, the constraint set of all the other negotiations will be updated to avoid unnecessary deals which are not as good as the one already found. After all the negotiations finish, the seller that offers the deal with the highest utility value will be chosen.

Against the requirements stated in section 1.2, this model is:

- *computationally tractable*: similar to Luo et al's model (see section 2.2.2), this model does not satisfy this requirement. It still has to face the trade-off between the computational tractability and the optimality of the final solution.
- *incomplete information*: this model requires no specific information about the opponent and makes no special assumptions about the agents' capability. Thus, it is applicable in the incomplete information setting.

- *partial information*: this model does not exploit the information about the opponent even when it is available.
- *negotiation deadline*: similar to Luo et al's model, this model does not satisfy this requirement.
- *concurrent negotiations*: this model is capable of handling multiple concurrent negotiations.
- *efficient outcomes*: this model is not guaranteed to find an optimal solution.
- *commitments*: this model is not designed for and, thus, is incapable of handling commitments amongst the participating agents.

One of the good points of this model is the capability of handling multiple concurrent negotiations using sub-negotiators. However, the model only proposes a conceptual approach to solving the concurrent negotiation problem without analyzing it theoretically or practically. This lack of evaluation eventually reduces its practicability. Also, because this model is based on CSP, it has the same disadvantages as Luo et al's model (i.e computationally intractable and unrealistic assumptions about the finite space of possible agreements). Consequently, the practicability of this model is reduced. Furthermore, the buyer agent in this model does not change its behavior throughout the negotiation. We believe, by modifying its behavior according to the characteristics of the seller it is negotiating with, the buyer will achieve better deals (in terms of the utility value of the final agreement; see section 3.5.4 for more details). Finally, the coordination mechanism of this model does not take into account the potential of partial information about the opponents. In many typical negotiation scenarios, partial information about the opponents is available and can be exploited.

2.3 Auction Protocols

Another very important area of negotiation is *auctions* (see section 1.2). Auctions are mainly used for modeling *one-to-many* (single-sided) or *many-to-many* (double-sided) negotiations and have proven to be one of the most popular and effective ways in trading goods over the Internet [Bapna *et al.*, 2001]. In this section, we only consider the single-sided auctions because they model the one-to-many negotiation case which is the most relevant to our research area.

Here, an auction is defined as a bidding mechanism, which consists of a set of auction rules that determine how to select the winner and how much the winner has to pay [Wolfstetter, 2002]. Typically, each auction consists of an *auctioneer* that attempts to sell a good for the highest possible price and a number of *bidders* that attempt to buy the good with the lowest possible price. The *auctioneer* sets the rules of the auction that the *bidders* must comply with. Each *bidder* uses a different strategy that follows the rules of the auction whilst trying to maximize its individual utility.

In an auction, the value of the good on offer is one of the most important criteria that differentiates different auctions settings. In *private value* auctions, the agent's preferences determine the value of the good (e.g. the auction of a cake that the winner will eat), whereas in *public/common value* auctions an agent's value of the good is determined by the values of the other agents (e.g. the auction of treasury bills). *Correlated value* auctions combine the aspect of both *private* and *public value* auctions, where the value of the good for an agent is partly determined by its own preferences and partly on the others' values (e.g. the auction of a contract within a project) [Sandholm, 1999a]. Here, *private value* auctions are the most relevant mechanism with respect to our negotiation domain since our negotiators operate in incomplete information settings and so do not have access to other agents' preferences.

A number of auction protocols have been developed, each of which has different properties under the three auction settings presented above. However, the following four protocols are the most widely used [Klemperer, 1999]:

- *English auction*: This is a *first-price*, *open-cry* and *ascending* auction. The auction starts when the auctioneer announces the *reservation price* for the good and allows the bidders to publicly raise their bids. The auction ends when no bidder is willing to raise its bid anymore and the winner is the *highest* bidder. The best strategy for each bidder is to make small increments to the current bid until the bid reaches the valuation price of the bidder.

- *Dutch auction*: This is an *open-cry* and *descending* auction. The auction starts when the auctioneer announces an artificially high *reservation price* for the good and awaits for a bid from any bidder. If a bidder makes a bid at that price, the auction ends and this bidder becomes the winner. If, however, no one bids for the good at that price then the auctioneer successively lowers the price until a bid is forthcoming.

- *First-price sealed-bid auction*: This is a *one-shot* auction, in which each bidder submits a bid for the good. The bids are sealed and unknown to the other bidders.

The auction ends when all the bids are collected and the winner is the highest bidder.

- *Vickrey auction*: This is a *one-shot*, *second-price* and *sealed-bid* auction, which operates almost exactly as a *first-price sealed-bid auction*. The only difference is that the winner only has to pay the price of the second highest bid. In this auction, the strategy for each bidder is bidding their true estimation for the good.

With regarding to the service or the good being auctioned, most of these auction mechanisms allow only negotiation about the single issue of price. This simplification has helped make auctions the dominant negotiation mechanism in today's electronic market [Bapna *et al.*, 2001]. However, also because of this simplification, these traditional techniques are not suitable to solve the multi-issue negotiation problem, which is the target of our research. To address this shortcoming, a number of auction types have been introduced that deal with multiple issues and agent deadlines [David *et al.*, 2002; Vulkan and Jennings, 2000]. Basically, these auctions target the environment that consists of one buyer agent and a number of seller agents and given their extensions they are clearly relevant to the research outlined in this thesis. In these auctions, the buyer publishes its preferences about the attributes of the goods together with its scoring function. The target sellers then use this information to compose their bids and submit to the buyer. After a fixed amount of time, if no new bid is submitted, the auction is terminated. The winner is the seller that submits the bid that has the highest utility value according to the buyer's scoring function. However, although these mechanisms deal with multiple issues, they are not chosen as our approach because of the following reasons:

- These techniques limit the capability of the participating agents. As can be seen, the seller agents are only allowed to use the specification of the buyer in creating their bids. They are not allowed to state their preferences to the buyer and they do not know how the buyer truly evaluates their bids. On the other hand, the buyer agent is only allowed to make a selection from the bids submitted by the sellers. The buyer is not allowed to modify any submitted bid and send it back to the seller as a counter-proposal. Consequently, the winner's bid may not be in the best interest of either the buyer or the sellers.

  Our concurrent bilateral approach overcomes this problem by allowing the buyer and the sellers to exchange two-way information by explicitly stating their preferences via the means of proposals. At each negotiation period, any agent can put forward its proposal, representing its preference of the good or the service, to the other agent. If this proposal suits the other agent's preferences, it will be accepted. Otherwise, a counter-proposal made by the opponent will be sent back if its deadline is not reached. By negotiating in this way, if an agreement is reached, it will be suitable for both agents and, furthermore, the social welfare will be increased.

- The time it takes to come to an agreement using most auction techniques is not flexible. Normally, it is fixed or indeterminate. Using concurrent negotiations can potentially provide a solution to this problem. Traditionally, in the aforementioned auction protocols, the buyer tends to select the winner at the last period of the negotiation because it will have a greater number of choices than if it does so at an earlier period. On the other hand, by allowing the agents to exchange two-way information, once a proposal is accepted by both parties, the negotiation may stop before the actual deadline is reached. This shortening can result in more efficiently resource and time saving for the participating agents.

In short, auction techniques are simple and easy to carry out. Even though they do offer the capabilities of handling multiple attributes and deadlines, they still limit the capabilities of the participating agents (as mentioned above). As a result, these techniques are not chosen as our approach.

2.4 Commitment Protocols

As mentioned in the section 1.2, not only does the model need to provide basic negotiation capabilities, it also needs to impose certain restrictions on the agents' behaviors. On one hand, the agents are required to have responsibilities for their decisions (e.g. in a bilateral negotiation, one agent should not be allowed to freely agree on a deal and later on dishonor it since this will have a negative effect on the other agent). On the other hand, however, they should be allowed to have some freedom in their choice of actions (in a sense that they are self-interested and rational). In order to achieve such a balance, a method of handling commitments among the participating agents must be provided. To date, there are a number of different mechanisms that can do this. To this end, we discuss what is probably the foremost approach, the *Contract Net Protocol* in section 2.4.1 and then the next two most prominent mechanisms, namely *Contingency Contract*s and *Leveled Commitment Contracts* in sections 2.4.2 and 2.4.3, respectively.

2.4.1 The Contract Net Protocol

Introduced during the 1980s, the aim of the Contract Net Protocol (CNP) is to provide a solution to the task distribution phase of cooperative problem solving among agents [Smith, 1980; Davis and Smith, 1983]. It is mainly based on the contracting mechanism that businesses use to control the exchange of goods and services and provides a simple solution to the problem of finding an appropriate agent to work on a given task. Figure 2.3 gives an overview of the CNP.

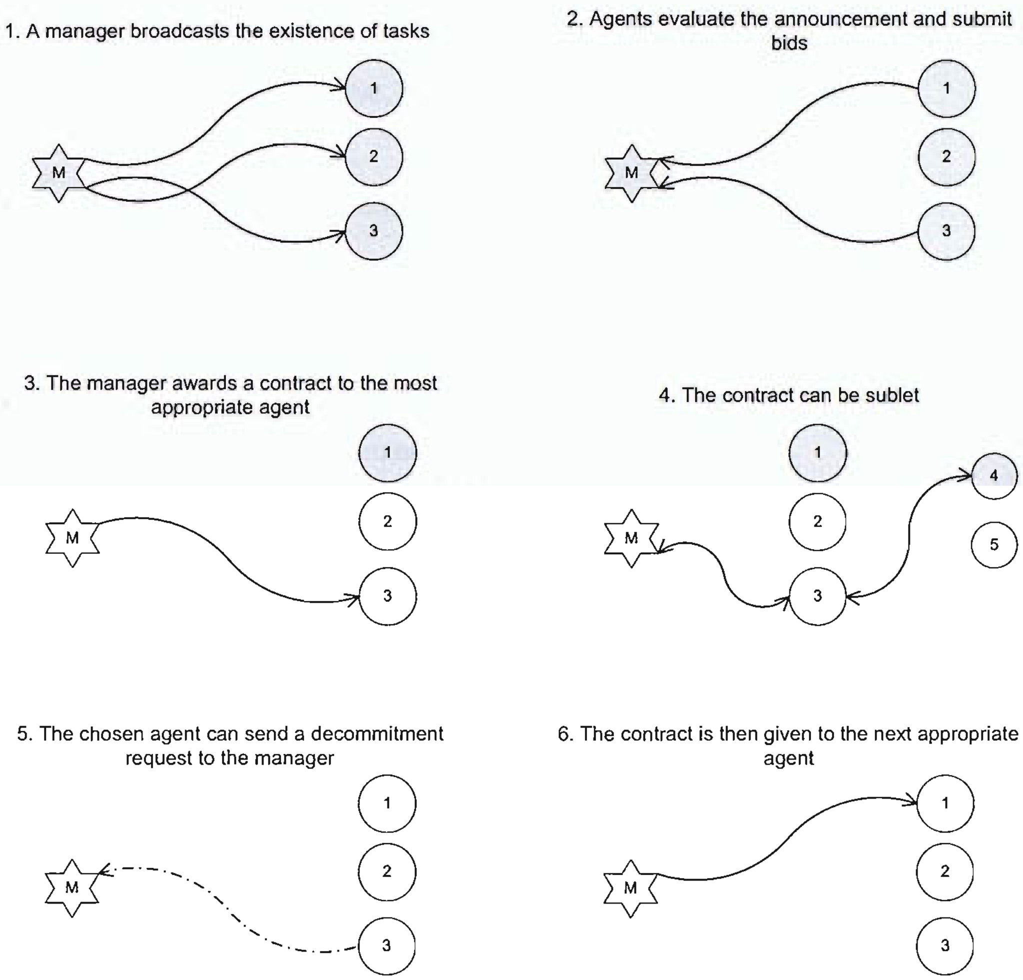


FIGURE 2.3: The overview of the contract net protocol.

Basically, when an agent has a task to be solved, it will make an announcement to other agents and become a *manager*. The other agents that have the capability to perform the task will be considered the *contractors*. From the manager's perspective, the process is as follows:

- Broadcast an announcement for the task that needs to be solved (step 1).

- Receive and evaluate bids from potential contractors (step 2).
- Award a contract to the most appropriate contractor (step 3).
- Wait for the selected contractor to finish the task. Reallocate the task to another contractor if the original contractor terminates the contract early (steps 5 and 6). The manager can also terminate the contract if the task no longer needs to be finished.
- Synthesize the final results from the allocated contractor.

From the contractor's perspective, the process is:

- Receive and evaluate the task announcement (step 2).
- Respond to that announcement with either a bid (if it is capable of handling the task) or a decline (if it cannot perform the task – step 2).
- Perform the task if the bid is accepted (step 3), can sublet the task to another contractor and become a manager (step 4).
- Terminate the contract early if required to (for any appropriate reason – step 5).
- Report the results.

The roles of manager or contractor do not need to be specified in advance. Any agent can be a manager simply by broadcasting a task announcement and any agent can be a contractor by answering a specific announcement. Thus, further task decomposition is possible; a contractor for a task can become a manager and sublet the task to another contractor. These manager-contractor links form a hierarchy of nodes, referred to as the *contract net*.

This protocol offers a basic commitment handling capability in the sense of a graceful performance degradation [Huhns and Stephens, 1999]. When a contract is given to the contractor, it is not considered binding on that particular agent. If, for some reason, the contractor could not finish the task, it does not mind losing its efforts and the manager could simply reallocate the task to another contractor. In that situation, the contractor simply sends a termination message to the manager, announcing it decommitment from the contract. Similarly, if the manager no longer needs the task to be done, it will simply send a termination message to the contractor and the contract will be terminated.

Speaking more generally, the contract net protocol provides a mechanism for symmetrically coordinated behavior [Faratin, 2001]. It is a distributed coordination architecture that is applicable in cooperative agents contexts (it can only work under the assumptions that the agents are cooperative) and has been integrated in a range of applications, including the allocation of computational jobs among processors in a network [Malone *et al.*, 1988], distributed meeting scheduling [Sen and Durfee, 1994; Sen and Durfee, 1998] and in cooperative coordination protocols [Decker and Lesser, 1995].

Against the requirements stated in section 1.2, this commitment model is:

- *computationally tractable and flexible*: the design of this commitment model is simple and straightforward. Thus, when being applied in the agent contexts, its computational complexity is tractable.
- *efficient*: it is not applicable to be used in our context so this criterion cannot be asserted.

The CNP provides a mechanism for handling commitments among agents. It is flexible in that the participating agents are not tied to their commitments, they are allowed to back down if they want to. To decommit, an agent just sends an appropriate message to the other agent and the contract will be amended accordingly (either being terminated or reallocated). Even when the contract has been carried out partially, the agents do not mind losing their efforts without any compensation [Sandholm and Lesser, 2001]. This is possible because the CNP only considers cooperative environments in which the agents are not self-interested. However, this assumption is highly problematic for our work because we are concerned with self-interested and rational agents. Thus, each agent has its specific goals and objectives and it is not willing to lose its efforts for nothing. Thus, this particular mechanism is not used in this work.

2.4.2 The Contingency Contracts Protocol

The contingency contracts protocol has been introduced to solve the commitment problem in non-cooperative environments [Raiffa, 1982]. This protocol is based on a game theory approach that focuses on modeling probabilistically known future events among self-interested agents. The contract is obligated upon future events; that is, an agent is allowed to break its commitment if any specified contingency condition happens. For this, an event verification mechanism is required in order to monitor the contingency events and later, enforce the validity of the contract.

In more detail, assume that two agents b and s negotiate about a service c. There are a number of future events that could have some effects on c (namely $e_1, e_2, \ldots$). Using the notion of contingency contracts, the final contract (which is agreed upon by both agents) will be represented as $(x_0, x_{e_1}, x_{e_2}, \ldots)$. Here x_0 represents the price that b agrees to pay s, $x_{e_i} \in \{true, false\}$ represents the statement that if e_i happens and x_{e_i} is *true*, any agent is allowed to back down from this contract. Under this notion, a non contingency contract is represented as $(x_0$, false, false, $\ldots)$.

To illustrate this mechanism, assume that b and s negotiate about buying a news subscription service (only an issue of price will be considered). There are only two future events, namely e_1 = [s stops selling the service] and e_2 = [s increases the price of its service]. Now under the notion of contingency contract, (£100, true, false) means that b will pay s £100 for using s's service. If s stops selling the service, any agent can back down. However, none of them is allowed to back down even when s increases the price for its subscription service.

Against the requirements stated in section 1.2, this commitment model is:

- *computationally tractable and flexible*: this model is only computationally tractable if the number of future events is small (when this number gets larger, it is cumbersome to include all the potential values for all the events). Furthermore, it is not flexible in the sense that if an unexpected future event happens that has not been prescribed, an agent will not be allowed to back down from its commitment even though that event might have a severe negative effect on the agent's outcome.
- *efficient*: for the buyer to make use of the situation, all the future events must be determined (otherwise, it will not be able to renege). This is not applicable in our case since not all the future events are known to the agents in advance. Thus, this criterion is not satisfied.

Thus, although this mechanism is applicable for self-interested agents in non-cooperative environments, there are a number of associated problems that means this approach is not suitable as a basis for our work.

The first, and the biggest, problem is that all the contingency conditions must be enumerated beforehand. This is inappropriate in many real world contexts since target environments are invariably dynamic. This means that not all the future events are known in advance and, therefore, it is impossible for the agents to make use of this commitment protocol efficiently.

The second problem is associated with the number of contingency conditions that need to be considered in the contract. If this number increases, this protocol can become cumbersome in monitoring all the future events. Furthermore, the state of the contract can be affected not only by these future events alone, but also by combinations of these events [Sandholm and Lesser, 1993; Rosenschein and Zlotkin, 1994]. For example, instead of allowing the agents to decommit based on individual events (e_1, e_2), it might be a combination of a number of them (e_1 and e_2). This leads to a potential combinatorial explosion of possible future states, each of which may be associated with a different contingency. Consequently, this may result in a combinatorial explosion in representing the contract.

2.4.3 The Leveled Commitment Contracts Protocol

The most recent approach to handling commitment among autonomous agents is the work by Sandholm and Lesser on the leveled commitment contracting protocol (LCC) [Sandholm and Lesser, 2001]. This protocol is based on the original Contract Net Protocol (see section 2.4.1) but differs in the sense that agents can drop their existing contracts only by paying a pre-agreed penalty fee to their opponents (rather than paying no fee). The protocol is built upon the intuition that agents should be able to decommit unilaterally from a contract for whatever reason they deem appropriate, as long as they pay some penalty fee. This means no explicit conditioning on future events is needed, as well as no event verification mechanism (as per contingency contracts). The reason why it is called "leveled commitment" is that different decommitment penalties determine correspondingly different levels of commitment. The larger the penalty, the lower the probability that an agent will decommit [Sandholm, 1999b]. Similar to some of the game theory based negotiation models (see section 2.1), the agents need to have information about their opponents (their actual and alternative commitment options) in order to calculate the Nash equilibrium decommitment thresholds.

Typically, an agent will decommit if it needs to do so. However, there is one interesting observation about this protocol. Once a contract is made, it can happen that a rational agent may decide not to decommit since it believes there is a chance that the other agent will decommit. If this is the case, the former agent could back out of its commitment without paying the penalty whilst collecting a penalty fee from the other agent. Consequently, some contracts are kept, even though they are inefficient for both agents. Nevertheless, it can be shown that leveled commitment contracts can enable deals in scenarios where both agents will not be benefit from full commitment contract [Andersson and Sandholm, 1999]. Furthermore, there are situations in which leveled commitment contracts will give higher expected payoffs to both agents than any full commitment contracts and for any full commitment contract, there is always a corresponding leveled commitment contract that will give higher expected utility than or, at least, equal value for both agents [Sandholm and Lesser, 2001].

Against the requirements stated in section 1.2, this commitment model is:

- *computationally tractable and flexible*: this model satisfies both constraints since the design is simple and straightforward. No special computational requirement is needed to compute the penalty value.
- *efficient*: this commitment handling mechanism satisfies this criteria since it only provides a method to handle commitments, but it does not actually affect the outcome of the negotiation game in which the agents are involved.

Of the three commitment mechanisms considered in this section, LCC is the most advanced since it is both computational feasible and able to provide an acceptable solution to the target commitment problem. Moreover, it is simpler than the contingency contracts in the sense that no future events need to be considered in advance and, thus, there is no need for a verification mechanism. However, there are a number of key problems that still need to be addressed [Nguyen and Jennings, 2004b]:

- The original LCC only covers a two person game. Since our target environment consists of more than two agents, we need to extend the protocol to accommodate this situation.
- The original LCC only reasons about decommitment. However, this is not sufficient to make the model applicable. Specifically, deciding when to commit is an equally important issue and we need to reason about this as well.
- LCC requires the agents to have information about the actual and alternative options of their opponents in order to be able to calculate the Nash equilibrium decommitment thresholds [Sandholm, 1999b]. This assumption is unrealistic in our target environments (recall, our agents need to be able to operate with *incomplete information* about the environment and the opponent) and, thus, needs to be relaxed.

- LCC typically assumes a fixed penalty for decommitting (pre-determined before the negotiation happens), regardless of the stage of the process at which the commitment is broken. Again, we believe that this assumption is unrealistic in many scenarios where the time an agent drops its commitment may have different effects to the other agent's outcome. For example, the earlier an agent drops its commitment, the easier for the other agent to find a suitable replacement.

Against this background, we decided to adopt the LCC protocol, however we need to extend it so that it is suitable for our target environment. This extended commitment model is an important contribution of this thesis and is defined in chapter 4.

2.5 Learning Techniques

Next, we look at the aspect of exploiting different sources of information in bargaining in order to gain better outcomes. Naturally, the agents in our target environment do not have access to every aspect of their environment. In particular, they do not have access to complete information about their opponents. Hence, they need to be able to act under uncertainty [Russell and Norvig, 2003]. However, in many situations, it is possible that they may have some partial information about their opponents (see section 1.2 for more detail).

Given this situation, most heuristic-based negotiation models can operate without complete information about their environment (as discussed in section 2.2). If, however, some partial information exists, it would improve the performance of the agents if they could exploit this (e.g. by adopting an appropriate negotiation strategy). Such exploitation can lead to optimal results (see section 2.1.2)[5] or can improve the performance of the model [Zeng and Sycara, 1998][6]. Against this background, this section looks at how partial information can be represented using probability theory and processed using various learning techniques to improve negotiation performance.

We start with a brief recap of *probability theory* in section 2.5.1 (the partial information that we are interested in is typically represented as a probability distribution), followed by two different approaches to machine learning (*reinforcement learning* techniques in section 2.5.2 and *statistical learning* techniques in section 2.5.3). A brief introduction

[5]The agent in this negotiation model makes use of the information about the other agent's reservation value, as well as their negotiation deadlines, to find a Pareto optimal solution.

[6]Information about the other agent's preferences helps improve the efficiency of the Bayesian learning algorithm employed in this model.

to *weighted learning* techniques is given in section 2.5.4 to conclude our discussion on learning techniques.

2.5.1 Probability Theory

Probability theory is chosen to represent partial information in our negotiation model because it is a common and well understood means of summarizing the uncertainty about the knowledge an agent possesses [Russell and Norvig, 2003].

In more detail, an agent's belief is represented by a *proposition*. Probability theory uses a language to state propositions. The basic element of the language is the *random variable* (the name of the variable is denoted with a capital letter, such as X, and the value of the variable is denoted with a lower case letter, such as x). Each variable has a *domain* that consists of the possible values it can take. There are three different types of domains that define the types of the variable:

- *Boolean random variable*: the domain of each variable consists of only two possible values: *true* or *false*. For example: the variable *Tough* (representing whether an agent is typically tough in negotiating) might have the domain of $\langle true, false \rangle$.
- *Discrete random variable*: the domain of each variable consists of a finite number of possible values. For example: the variable *Strategy* (representing the chosen negotiation strategy of an agent) might have the domain of $\langle tough, conceder, linear \rangle$ (see section 2.2.1 for more detail).
- *Continuous random variable*: each variable can take a real number as a possible value. The domain of each variable can be either the entire set of real numbers or some subset. For example, the variable *Price* (representing the price of a holiday package, see section 1.2) might have the domain of $[0, 1000]$.

The *unconditional* or *prior* probability of a proposition a, $P(a)$, is the degree of belief assigned to it in the absence of any other information. For example, if the unconditional probability that an agent is a tough negotiator is 0.4, then:

$$P(Tough = true) = 0.4 \text{ or } P(Tough) = 0.4$$

For random variables, such as *Strategy*, $P(Strategy)$ denotes the probabilities of all the possible values of this variable.

$$P(Strategy) = \langle 0.2, 0.3, 0.5 \rangle \Leftrightarrow \begin{cases} P(Strategy = tough) = 0.2 \\ P(Strategy = conceder) = 0.3 \\ P(Strategy = linear) = 0.5 \end{cases}$$

The statement, $P(Strategy)$, defines a prior *probability distribution* for the discrete random variable *Strategy*. For continuous variables, it is impossible to list all the distribution probabilities. Instead, the probability that a random variable X takes on a value x is defined as a parameterized function of x. For example, let the random variable X denote the price of a bike, then the statement $P(X = x) = U[100, 200](x)$ expresses the belief that X is distributed uniformly between £100 and £200.

Now, probability distributions for continuous variables are called *probability density functions*. These density functions differ in meaning from discrete distributions. For example, $P(X = 150) = U[100, 200](150) = 0.01$. This does not mean that there is a 1% chance that the price will be exactly £150. Instead, the technical meaning is that the probability that the price is in a small region around £150 is equal to 0.01 divided by the width of the region:

$$\lim_{dx \to 0} P(150 \leq X \leq 150 + dx)dx = 0.01$$

Unconditional probability is not applicable if the agent has obtained some evidence that is concerned with the previous unconditional random variable (e.g. variable a might not have a particular value if variable b has not been set to have another particular value). In such cases, *conditional* or *posterior* probabilities must be used. In this case, $P(a|b)$ denotes the probability of a given b. For example, $P(tough|consistant) = 0.7$ indicates that if an agent is observed to be consistently keeping its offer's value close to its initial value and no other information is available, then the probability that this agent is a tough negotiator is 0.7. The formula to define conditional probability from unconditional probability is given as:

$$P(a|b) = \frac{P(a \wedge b)}{P(b)}$$

where $P(b) > 0$ and $P(a \wedge b)$ denotes the probability of both propositions a and b. This equation can also be written as:

$$P(a \wedge b) = P(a|b)P(b) \tag{2.2}$$

which is called the *product rule*. This is also true for the other way around:

$$P(a \wedge b) = P(b|a)P(a) \tag{2.3}$$

The *probability distribution* for continuous random variables can be expressed in a similar way to the *probability distribution* for discrete random variables:

$$P(X,Y) = P(X|Y)P(Y)$$

where $P(X|Y)$ give the values of $P(X = x_i|Y = y_i)$ for each possible i, j.

From equations 2.2 and 2.3, we have:

$$P(b|a)P(a) = P(a|b)P(b)$$

Dividing by $P(a)$ (assuming $P(a) > 0$) we have:

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)} \tag{2.4}$$

This equation is known as the *Bayesian rule* or *Bayesian theorem.* The more general version of this rule, with a background evidence e, is expressed as follows (assuming $P(X, e) > 0$):

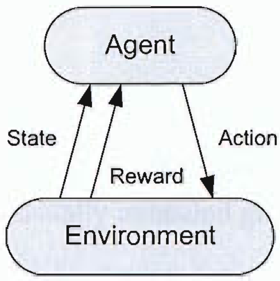
$$P(Y|X,e) = \frac{P(X|Y,e)P(Y,e)}{P(X,e)} \tag{2.5}$$

This equation represents the general *Bayesian rule.*

Next, we look at different learning techniques and their applications in the negotiation domain. As will be seen, each has the own advantages and disadvantages.

2.5.2 Reinforcement Learning Techniques

Reinforcement learning is a technique in which the agents rely upon their past experience to improve their behaviors [Kaelbling *et al.*, 1996; Sutton and Barto, 1998]. A reinforcement learning system must provide a scalar evaluation (e.g a reward) for each output an agent has chosen. The idea behind is that if an action produces favorable results, that tendency should be strengthened (reinforced) and, correspondingly, weakened if it produces unfavorable results [Mitchel, 1997]. Figure 2.4 shows the general procedures that an agent must follow using reinforcement learning.



[1] The agent senses the environment
[2] The agent makes a decision about the next action to execute by analysing its current circumstances (state).
[3] The agent executes the selected action
[4] The agent obtains a reinforcement reward for the action it performed.
[5] Go to [1]

FIGURE 2.4: Agent's general procedure in RL (taken from [Mitchel, 1997]).

RL tasks are naturally divided into two types: sequential and non-sequential. In non-sequential tasks, an agent needs to learn to map from states to actions that maximizes the expected intermediate payoffs from executing different actions. Any action selected by the agent will have no effect on its future states and, thus, its future payoffs. On the other hand, in sequential tasks, the agent needs to carefully evaluate its actions on the basis of their long-term consequences [Sandholm and Crites, 1995]. Among the reinforcement learning techniques, Q-learning is the most widely employed in multi-agent learning and has been applied in a variety of multi agent systems [Claus and Boutilier, 1998; Sen and Weiss, 1999; Excelente-Toledo, 2003]. In short, Q-learning works by estimating the values of state-action pairs and by selecting the action that maximize this estimation.

In more detail, let $Q(a, s)$ denote the value of doing action a in state s. This value is directly related to the agent's utility value as $U(s) = max_a Q(a, s)$. Now assume that in state s action a is selected and an immediate payoff τ is received and the current state moves to s'. In this case, the value for $Q(a, s)$ is updated as follows:

$$Q(a, s) \leftarrow Q(a, s) + \alpha(\tau + \gamma max_{a'} Q(a', s') - Q(a, s))$$

where α is the learning rate and $0 \leq \gamma \leq 1$ is the discount factor (see [Russell and Norvig, 2003] for more detail on these parameters of a Q-learning model).

In a stable and Markovian environment[7], this algorithm is guaranteed to converge to the correct Q-values [Mitchel, 1997]. Even though Q-learning does not specify which

[7] An environment in which the transition from the current state only depends on the current state and the action taken in it, without any influence from the history that led to the current state.

action should be taken at each step, typically, in practice, a method is usually provided that both ensures sufficient exploration and selects actions that have higher value estimates. One such example is the Boltzmann distribution [Hastings, 1997], where the probability of selecting action a_i from state s is:

$$p(a_i) = \frac{e^{Q(a_i,s)/t}}{\sum_a e^{Q(a,s)/t}}$$

where t is a computational temperature parameter that controls the amount of exploration, usually annealed gradually overtime.

Since it does not require a model for either learning or action selection, this method has been applied into a number of multi-agent systems and it has proven to be useful in specialized situations [Sandholm and Crites, 1995; Sridharan and Tesauro, 2000; Arai *et al.*, 2000]. However, there are a number of problems with reinforcement learning. First, in our target negotiation model, we are unable to give a value for each action of our agent. Since we target incomplete information scenarios, we do not know how the other agents evaluate the action of our agent and, thus, such reinforcement values cannot be obtained. Furthermore, our target environment is not stable and Markovian; the actions of the participating agents might depend on the whole history of behaviors in the encounter, not just on the behavior in the current state. Thus, it is not guaranteed to give the optimal result. Consequently, this learning technique is not adopted in our model.

2.5.3 Statistical Learning Techniques

Here, we look into statistical learning techniques and how they can be applied to different negotiation situations. In general, these techniques use *data* and *hypotheses* to make *predictions*. The target domain is represented by a number of different random variables. *Data* is the instantiation of some (or all) of these random variables and the hypotheses are the probabilistic theories of how the domain works [Russell and Norvig, 2003]. The general rule is the predictions are then made using the posterior probabilities of the hypotheses. To illustrate this, consider the Bayesian learning method (the most powerful statistical method) as applied in the Bazaar negotiation model [Zeng and Sycara, 1998]:

Here, the negotiation is a game of 2 players, following Rubinstein's model of alternating offers (section 2.1.1) that terminates with either *Accept* (denotes the final agreement reached) or *Quit* (implies that no agreement is found). A negotiation is then modeled by a 10-tuple $\langle N, M, \triangle, A, H, Q, \Omega, P, C, G\rangle$, where:

- N is the set of players,
- M is the set of issues covered in the negotiation (i.e. price, quantity ...),
- $\triangle$ is the set of vectors of each and every dimension of an agreement in the negotiation,
- A is the set of all the possible actions that can be taken by every player: $A \equiv \triangle \cup \{Accept, Quit\}$. For each player $i \in N$, A_i is the set of possible agreements: $\forall i \in N : A_i \subset A$,
- H is the set of sequences (finite or infinite) that satisfies:
  - The elements of each sequence are defined over A,
  - The empty sequence Φ is a member of H,
  - If $(a^k)_{k=1,\dots,K} \in H$ and $L < K$ then $(a^k)_{k=1,\dots,L} \in H$,
  - If $(a^k)_{k=1,\dots,K} \in H$ and $a^K \in \{Accept, Quit\}$ then $a^k \notin \{Accept, Quit\}$ $\forall k \in \{1, \dots, K-1\}$,
  - Z is the set of terminal history. $Z = \{h\} : h = (a^k)_{k=1,\dots,K} \in H$ and $a^K \in \{Accept, Quit\}$.
- Q is the function that associates each non-terminal history ($h \in$ H\Z) to a member of N.
- Ω is the set of relevant information entities. It represents the players' knowledge and beliefs about the following aspects of the negotiation:
  - The parameters of the environment, which can change over time (i.e. the overall product supply, demand and interest rate).
  - Beliefs about other players, including:
    1. Beliefs about the factual aspects of other agents, such as their payoff functions, the resources that they have, etc.
    2. Beliefs about the decision making process of other agents, such as their reservation prices, etc.
    3. Beliefs about the meta-level issues of other agents, such as their risk-taking attitudes, etc.

- P is the probability distribution defined over Ω: $P = \{P_{h,i}\}\ \forall i \in N, h \in \{H\backslash Z\}$. This distribution is a concise representation of each player's knowledge at each stage of the negotiation,

- C is the set of costs of executing an action a for agent i (communication or time related). $C = \{C_{i,h,a}\}\ \forall i \in N, h \in \{H\backslash Z\}, a \in A_i$,

- For each terminal history $h \in Z$, player $i \in N$, a *preference* relation $\succeq_i$ on h and $P_{h,i}(x), x \in Q$. $\succeq_i$ in turn results in an evaluation function $E_X^{(h,i)}[G_i(X, h)]$ that will be used in selecting which action to be executed.

- For each player i, a negotiation strategy is a sequence of actions $(a_i^k, k = 1, \ldots, L)$ where $a_i^k \in A_i \backslash \{Accept, Quit\}, \forall k = 1, \ldots, K-1$ and $a_i^K \in \{Accept, Quit\}$.

- Before the negotiation starts, each player i has a certain amount of knowledge about Ω (including knowledge about the environment as well as other players) denoted as $P_{\Omega,i}$,

- Suppose player i has been interacting with another player j for k times (i has sent exactly k proposals to j and has received either k or $k+1$ proposals from j, depending on whom has initiated the negotiation) and neither *Accept* or *Quit* has been proposed. Assume that the following information is available to each player i:

  1. All the actions that have been taken by all the agents up to date. Formally, each and every history h that is a sequence of k actions is known to i. Let $H_{i,k}$ denote the set of these histories,
  2. The set of subjective probability distributions over Ω: $P_{H_{i,k-1},i} \equiv \{P_{h,i} \mid h \in H_{i,k-1}\}$ is known to i.

Given the concepts, in order to reply to the most recent action taken by other participant, i takes the following steps:

1. Update its subjective evaluation about the environment and other players using the Bayesian rule (equation 2.5). Thus, given prior distribution $P_{H_{i,k-1},i}$ and the newly incoming information $H_{i,k}$, generate the posterior distribution $P_{H_{i,k},i}$.

2. For $h \in H_{i,k}$, select the best action from A_i according to the following recursive evaluation criteria:

$$V_{i,k,h} = \begin{cases} E_X^{(h,i)}[G_i(X,h)] \ \text{if} \ h \in Z \\ max_{a_i \in A_i}\{-C_{i,a_i,h} + \int_X [V_{i,k+1,(h,a_i)} \times P_{h,i}(X)]dX\} \ \text{otherwise} \end{cases}$$

To illustrate the Bazaar negotiation framework, consider the following example. The negotiation is considered only from the viewpoint of the buyer and the relevant information set Ω consists of only one item: its belief about the supplier's reservation price $RP_{supplier}$. An agent's reservation price is the maximum (minimum if it is the supplier) price that it consider acceptable. As shown in figure 2.5, if $RP_{supplier} \leq RP_{buyer}$, a *zone of agreement* exists, in which any point within it can become an agreement. The negotiation only finishes successfully if this zone exists and, if it does, the negotiation will consist of a series of concessions from both players. Eventually, a proposal within this zone will be reached and agreed by both players.

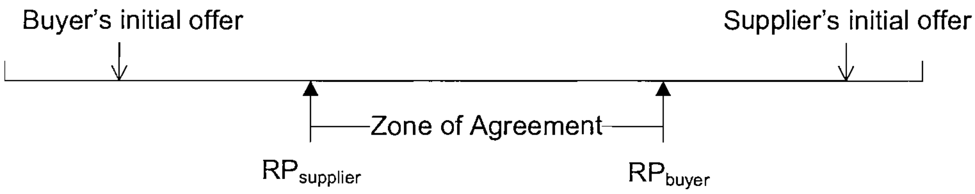


FIGURE 2.5: An example of reservation prices and the zone of agreement.

The main point of using Bayesian learning in this case is to predict the possibilities of potential values for $RP_{supplier}$. As the buyer gains more accurate information about this value, it will be able to make more advantageous offers for itself [Zeng and Sycara, 1998]. In particular, partial belief about $RP_{supplier}$ can be represented by a set of hypotheses $H_i, i = 1, \ldots, n$ where H_1 can be $RP_{supplier} = £100$ and H_2 can be $RP_{supplier} = £90$ for example. A priori knowledge of the buyer is the probabilistic distribution over H_i (i.e. $P(H_1) = 0.5, \ldots$). Later, when a proposal from the suppliers arrives ($Offer_{supplier}$), its value together with the domain knowledge[8] will be used to update the posterior subjective evaluation over H_i.

Given the encoded domain knowledge (in the form of conditional statements and the signal, e), the buyer can update its belief about $RP_{supplier}$ using the standard Bayesian updating rule (extended from equation 2.5):

$$P(H_i \mid e) = \frac{P(H_i)P(e \mid H_i)}{\Sigma_{k=1}^{n} P(e \mid H_k)P(H_k)} \tag{2.6}$$

[8]The set of observations, such as "usually in business, people will offer a price that is 17% higher than the reservation value". This is represented by a set of conditional statements, for example $P(e_1 \mid H_1) = 0.3$ where e_1 represents "$Offer_{supplier} = 117$".

For the purpose of illustration, suppose that the buyer knows that $RP_{supplier}$ is either £100 or £90. This means the buyer has only two hypotheses: H_1: $RP_{supplier} = £100$ and H_2: $RP_{supplier} = £90$. Prior to the negotiation, the buyer does not have any other additional information; its a priori knowledge is summarized as $P(H_1) = 0.5$ and $P(H_2) = 0.5$. Additionally, assume that the buyer is aware of the observation "suppliers will typically offer a price which is above their reservation price by 17%". Part of this observation is encoded as $P(e_1 \mid H_1) = 0.3$ and $P(e_1 \mid H_2) = 0.05$, where e_1 denotes the event that the supplier demands £117 for the good under the negotiation. Now, if the buyer receives the offer of £117 from the supplier, its posterior estimation of $RP_{supplier}$ can be calculated (using equation 2.6) as follows:

$$P(H_1 \mid e_1) = \frac{P(H_1)P(e_1|H_1)}{P(H_1)P(e_1|H_1)+P(H_2)P(e_1|H_2)} = 85.7\%$$

$$P(H_2 \mid e_1) = \frac{P(H_2)P(e_1|H_2)}{P(H_2)P(e_1|H_1)+P(H_2)P(e_1|H_2)} = 14.3\%$$

Now assume that the negotiation strategy that the buyer uses is a simple "p*ropose a price which is equal to the estimated* $RP_{supplier}$" [Zeng and Sycara, 1998]. Prior to the negotiation process, based on its belief (H_1 and H_2), the buyer will offer £95. After receiving the proposal of £117 from the supplier, the buyer will offer £98.57 as the counter offer. Since this value is calculated based on a more accurate estimation of the supplier's utility structure, it might result in a potentially more beneficial final outcome for the buyer. Furthermore, it is also possible for both players to come to agreement more efficiently. In fact, it is shown via empirical evaluation that the greater the zone of agreement, the better the learning agents seize the opportunity [Zeng and Sycara, 1998].

As can be seen, Bazaar provides a negotiation model that uses Bayesian learning to improve its operational efficiency. However, the main problem of this particular negotiation model is that its complexity dramatically increases as the set of negotiation issues and/or the set of subjective probability distributions increase (it only works well with a single issue and one probability distribution). Moreover, it requires a complete parametric model in order to be able to operate [Bui *et al.*, 1996; Heckerman, 1999]. Nevertheless, the benefit of using Bayesian learning in a multi agent negotiation model is that it can provide a learning framework that does not require a large sample in order to be able to operate. For this reason, this particular learning technique is adopted as a part of our learning framework (discussed in section 3.4.3 and chapter 5).

2.5.4 Weighted Learning Techniques

The other learning techniques that we want to address in this section is the family of techniques called *weighted learning* [Atkeson *et al.*, 1997]. In particular, we are interested in the most two prominent techniques, namely *kernel density estimation* [Wand and Jones, 1995] and *function smoothing techniques* [Moore *et al.*, 1995]. Similar to other learning methods, these techniques aim to improve the agent's performance based on the past experiences and the values of the offers and counter-offers received during the negotiation process. Unlike Bayesian learning, these techniques provide a simple way of finding structure in data sets without the imposition of a parametric model. Basically, any information gained prior to the negotiation is processed offline and it can then then be augmented by online learning that reflects new information emerging from the ongoing encounter.

Weighted learning is usually selected for its lowcomputational complexity and its effectiveness in learning [Wand and Jones, 1995; Moore *et al.*, 1995]. Now, complexity is a very important constraint in our model since our agents are bounded in their computational power (see section 1.2). Moreover, with weighted learning, most of the learning can be done offline and the online part has the complexity of n log n for the KDE [Wand and Jones, 1995] and n^2 for the function smoothing techniques (see section 5.1). More importantly, when learning about one aspect of the opponent (i.e. their negotiation strategies), these techniques do not require complete information (in Bayesian learning, a priori distribution must be given in advance). Nevertheless, the more information that is available, the better the performance of our model (see section 5.3 for more details).

2.6 Summary

Having presented existing negotiation, commitment and learning models, this section discusses the advantages, as well as disadvantages of such models. First, table 2.3 summarizes our discussions about existing negotiation models[9]. As can be seen, there are a number of existing bilateral negotiation models that each adopt a different approach to solve the negotiation problem. Consequently, the performance of these models vary in different negotiation settings. There are also trade-offs between different approaches. For example, most game theory based models guarantee to find an optimal solution but

[9] In this table, $\surd$ denotes the fact that the corresponding requirement is fully satisfied, * denotes the fact that the corresponding requirement is only partially satisfied, *unknown* denotes the fact that there is not enough information to assess the corresponding requirement, and $\times$ indicates that the corresponding requirement is not satisfied.

are not computationally tractable, whereas most heuristic-based models are computationally tractable but cannot guarantee optimality.

	Kraus	Faratin	Fatima	Luo	Rahwan
Number of agents	2	2	2	2	n
Number of issues	1 & n	n	1	n	n
Approaches (Game-theory, CSP, Heuristic)	G	H	G & H	CSP	CSP
Protocols (Interactive)	I	I	I	I	I
Computationally tractable	√	√	√	×	×
Incomplete information	*	√	*	√	√
Partial information	√	×	√	×	×
Negotiation deadline	√	√	√	×	×
Concurrent negotiation	×	×	×	×	√
Efficient outcome	√	×	√	√	×
Commitment handling	×	×	×	×	×

TABLE 2.3: The comparison matrix of existing negotiation models.

In addition, there is very little work which specifically addresses the problem of multiple concurrent encounters (only Rahwan et al's model). Now, this is a particular problem in situations where there are a number of providers that offer the same service and only one of them needs to be chosen. Particularly, one of the inherent drawbacks of not being able to handle this situation is that the agent has to a priori identify a single partner to interact with. While this is acceptable if there is only one provider of the desired service, it is inefficient in an uncertain setting if there are multiple providers of the service and each has different characteristics. In this multiple provider case, there are two alternatives: (1) negotiate sequentially with all the providers or (2) negotiate concurrently with them. The former case has the disadvantage that it may result in lengthy negotiation encounters, but has the advantage that it is comparatively easy to use the outcome of one negotiation to dictate behavior in subsequent ones. For example, if an agent reaches an agreement of price p for service S in negotiation i, then in all subsequent negotiations, p can be viewed as its reservation price since it can always claim this agreement. The latter case has the advantage of taking less time, but the disadvantage that coordinating behaviors among the negotiation threads is more difficult[10].

Against this background, we concentrate on the concurrent case and develop a coordinated bidding model in which the various negotiation threads mutually influence one another. By mutually influencing, we mean that the progress and agreement in one negotiation thread is used to alter the behavior of the agent in another thread for the same service. For example, having obtained a good deal in one thread, the agent may adopt a tougher stance in its other threads, to see if it can get an even better deal than the one it can fall back on. Based on these observations, we introduce a concurrent negotiation

[10]Such coordination is necessary to ensure the agent does not end up with multiple agreements for the service when only one is required.

model that is capable of handling multiple negotiations simultaneously. The details of this work are expressed in chapter 3.

With regard to commitment handling capabilities, in traditional bilateral negotiation (full commitment contracts), a contract, once made, is considered binding to both parties (i.e. neither agent can drop its commitment no matter how future events affect its status) [Kraus, 1993; Andersson and Sandholm, 1999]. However, there are many situations in which one agent will gain better utility values if it can drop its current commitment (i.e. it is offered a proposal that has higher utility value than the current agreement it is holding). By being tied to its current commitment, the chance of an agent having a better deal is lost. In our concurrent negotiation context, since the buyer only seeks to buy the service from a single provider, if it is limited to only a single agreement, it will have no chance of getting a high value deal. Thus, full commitment contracts are undesirable in our target contexts.

On the other hand, if the agents are allowed to freely back down from their commitments, there is no guarantee that an agreement will be honored in the future. Therefore, the result of a negotiation is uncertain and could result in a waste of time and resource for all the participants. This will, in turn, reduce the desirability of engaging in the bargaining encounter of the agents. Consequently, a method to control and enforce commitments among the agents is desirable.

Amongst the available mechanisms to handle commitment, the Contract Net Protocol is targeted at cooperative environments and, thus, is inapplicable. The contingency contracts protocol is able to handle non-cooperative scenarios and self-interested agents. However, it is only useful if the number of contingency conditions is small and all the future events are known to the participating agents beforehand. Again, these assumptions are somewhat limited and unrealistic. Thus, we choose not to adopt this approach in our negotiation model. On the other hand, the leveled commitment contracts protocol provides an efficient and computation feasible mechanism to handle commitment among the agents. It does force the agents to have responsibility for their behaviors, whilst allowing them to have the freedom to back down from their commitments if they have the incentive to do so. For this reason, this approach is selected as the base for the commitment handler in our model. However, we need to extended the original LCC protocol to accommodate multiple agents situations as well as other important improvements (see chapter 4 for more details).

With regard to learning capabilities, we would like to retain the practicability and low computational requirements of our model (as per section 1.2). Thus, our target learning environment must be based only on the data gathered from the interactions between our buyer and the participating seller agents. Also we do not want to introduce unrealistic assumptions about the agents in our scenarios (related, for example, to assumptions about knowledge about the agents' preferences or deadlines). Furthermore, we do not want to employ additional structures to govern the learning process since this is likely to require additional computational and resources for our agent. Given all of this, the learning domain that we target can be considered *unsupervised* in that the causal relation between the action of our agent and its outcome cannot be determined[11].

There are a number of unsupervised learning techniques that could potentially be employed in this scenario; each of which has its own advantages and disadvantages. The most common model, and the one that is often used in multi-agent research, is that of reinforcement learning (see section 2.5.2). Such techniques are popular because they do not require a model to be set up and are also able to provide good results in specialized negotiation situations [Kaelbling *et al.*, 1996]. Thus the learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them. Nevertheless, their main disadvantage is the fact that for every action an agent selects, there must be a feedback given and this is not always feasible in our negotiation contexts[12]. Furthermore, in order to generate correct feedback values for the model to work, most of these techniques employ assumptions similar to those of game-theory (see section 2.1). Although such assumptions provide the agent with appropriate feedback values for their actions, converging to a good learning performance, the practicability of such learning models is limited [Kaelbling *et al.*, 1996]. Consequently, we do not employ this kind of technique in our learning model. The same broad criticisms can be leveled at many of the other forms of unsupervised learning including analytical learning or instance-based learning [Alpaydin, 2004].

On the other hand, the weighted learning approach (see section 2.5.4), is an unsupervised learning technique that does not require a parametric model (unlike Bayesian learning, it can operate without the explicit probability representation of the learning data). It also does not require a feedback for each action step (since its learning structure is based only on the interactions amongst the participating entities). As can be seen,

[11] *Supervised* learning techniques (such as neural networks, statistical machine learning) are controlled using an external mediator whose role is to adjust the parameters so that the causal relation between the input and output parameters is justified (*instructive*). In contrast, *unsupervised* learning does not have that requirement [Michie *et al.*, 1994].

[12] In our model, the agents operate in an incomplete information scenario. The only information that they get from the other agents is their proposals. Consequently, an appropriate feedback is unable to be given to each of the agent's action. One can argue that a randomized feedback could be used instead of a proper value, however, we believe that by doing this way the performance of the learning model will be greatly reduced.

such advantages suit our target learning domain and, thus, it is selected as the base for our learning model, described in chapter 5.

Chapter 3

The Concurrent Bilateral Negotiation Model

This chapter introduces a new model for concurrent bilateral negotiations, denoted eCN (e-commerce Concurrent Negotiation). Its aim is to address the shortcomings of the current models described in chapter 2 and to implement the basic negotiation requirements discussed in section 1.2[1]. The sections start with a description of the internal reasoning of the buyer agent in section 3.1 and the negotiation protocol in section 3.2. The list of symbols that are used throughout this thesis is then detailed in section 3.3. Next, we go on to the design of the concurrent negotiation model in section 3.4, followed by the experimental results in section 3.5 and the practical application of the model in section 3.6. Finally, section 3.7 summarizes.

3.1 The Buyer's Internal Reasoning Process

We start the section with a brief discussion of the model. The negotiation environment that the model targets consists of one agent, called *the buyer*, seeking a service $\overline{s}$ and with n agents (called *the sellers*) that are capable of providing $\overline{s}$. Each agent has its own negotiation deadline (t_{max} for the buyer and $t_{k_{max}}$ for each seller k), after which the negotiation cannot continue. At each negotiation time period, the interest of each agent, both in buying and selling $\overline{s}$, is represented by a proposal ϕ. ϕ is composed of m issues, where each issue j is an attribute of $\overline{s}$ (e.g. price, quantity, etc.).

[1]Excluding the requirements related to commitments and learning which will be dealt with in chapters 4 and 5, respectively

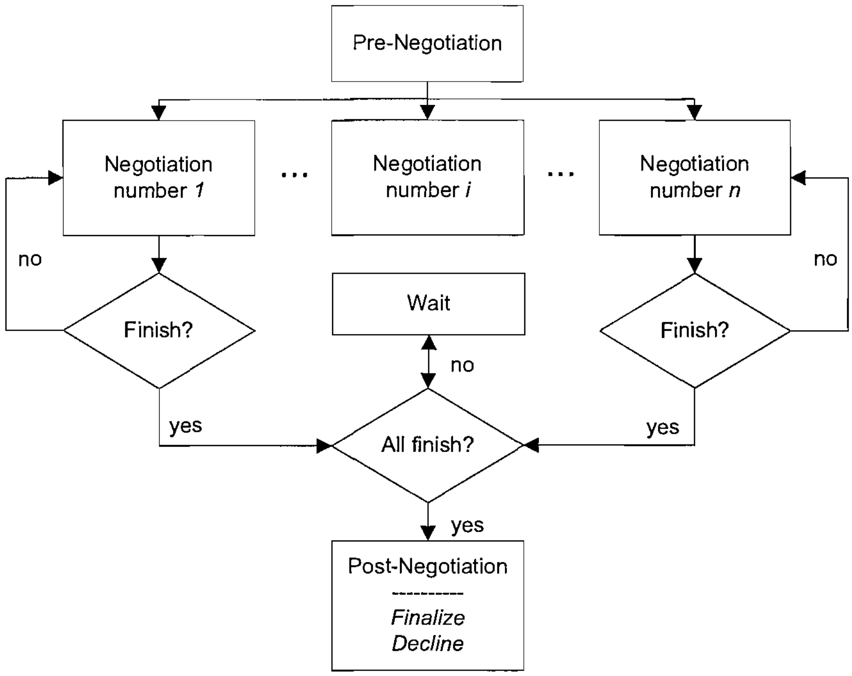


FIGURE 3.1: The buyer's internal reasoning process.

In particular, the buyer agent's internal reasoning process, or how it engages in and finalizes the negotiations, is expressed in figure 3.1. The process starts with a pre-negotiation phase, in which the buyer prepares the strategy for each of its threads[2], based on its belief about the current market situation and the information about the other agents (see section 3.4.3). After this phase, all the negotiations are carried out concurrently (where each negotiation follows the protocol described in section 3.2). The status of each negotiation is reported back to the buyer and this may in turn affect the other ongoing negotiations (see section 3.4.3). After all the negotiations have finished, the buyer goes into a post-negotiation phase in which it finalizes a deal with the agent that has provided the highest value agreement and declines all deals with the other agents. This finalization phase is necessary in order to deal with the problem of concurrent encounters. As it currently stands, if the buyer accepts an offer from a seller then this is viewed as binding on the seller (for a specified period of time, which is assumed to be no longer than t_{max}). However, it is not binding on the buyer. Thus, the buyer may accept offers from multiple sellers in any one negotiation episode. However, when it has completed all the negotiations, the buyer will *finalize* one of the accepted deals with one of the sellers and *decline* the others, thus freeing them from their commitments to the proposal. This two phase process is necessary so that the buyer can use accepted deals as a base line for its

[2]Each thread handles a negotiation with a particular seller (see section 3.4.2 for more detail).

subsequent concurrent negotiations.[3]

3.2 The Negotiation Protocol

As discussed in section 1.1, the agents participating in the negotiation need to communicate and interact with each other. An interaction protocol is therefore needed to control the flow of communication among the agents, so that chaotic situations are avoided. Once specified, the agents must follow the protocol strictly in order to make the result of the negotiation valid. In the scope of this research, we will not consider the situation where agents do not follow the protocol.

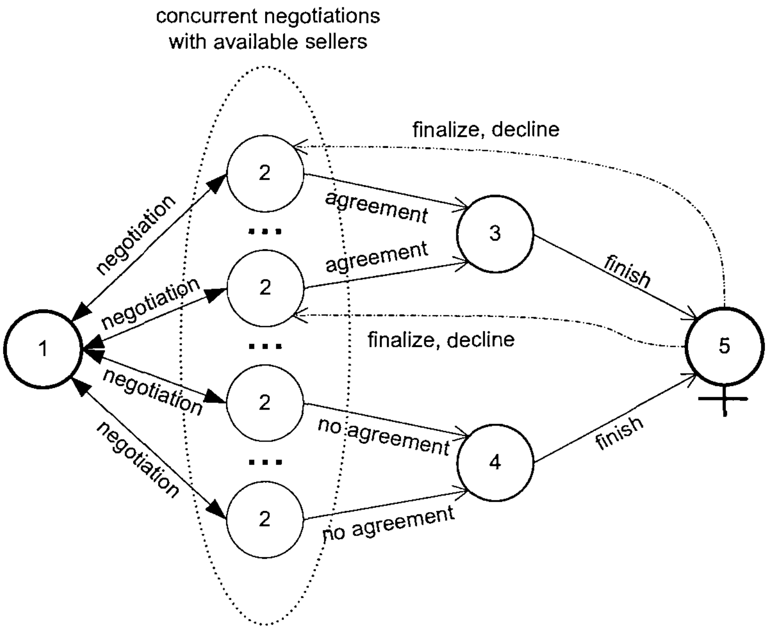


FIGURE 3.2: The execution states of the model.

In more detail, figure 3.2 shows five execution states of the negotiations. First, all the negotiations are started simultaneously (system state transits from starting state 1 to negotiating state 2). During the life time of the bargaining process, any negotiation that terminates with an agreement will move its state from negotiating state 2 to agreement state 3, whereas those that terminate without an agreement will move their states to failure state 4. After all the negotiation terminates, the system state will move from state 3 and 4 to the terminal state 5, at which point the buyer agent finalizes the deal that has the highest utility value and declines all the other accepted agreements (represented by the dotted lines).

[3]This is obviously biased in favor of the buyer. In chapter 4, we relax this constraint so that sellers can also renege on deals.

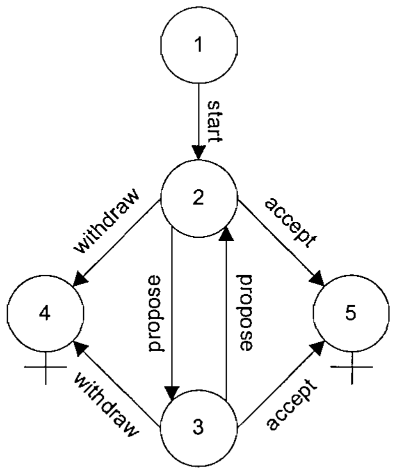


FIGURE 3.3: The negotiation protocol.

In each thread, the negotiation with the corresponding seller agent will be carried out via the negotiation protocol detailed in figure 3.3. This protocol is based on Rubinstein's model of alternating offers (section 2.1.1). Specifically, there are five states in this protocol in which state 1 is the initial state and states 4 and 5 are the termination states. First, the buyer's thread starts the negotiation (state transits from 1 to 2) by proposing an initial offer to the opponent (state transits from 2 to 3). At this point, the opponent has three options: (1) accept the offer which terminates the negotiation (state transits to 5) and causes the seller to wait for the buyer to finalize the deal (see section 3.4.1); (2) withdraw from the negotiation (state transits to 4) which signals the termination of the negotiation; or (3) propose a counter-offer to the buyer (state transits to 2) which causes the negotiation to move to the next period. Typically, the agents iterate between states 2 and 3, exchanging counter-offers until either the offer is accepted by the opponent or one of them reaches its negotiation deadline and withdraws. The primitives that the agents are allowed to use in the negotiation protocol are specified in table 3.1.

3.3 The List of Symbols

To start the process of formalizing our model, table 3.2 lists the symbols that are used in developing the model. These symbols are listed together with a brief description of their meaning. The detailed explanations of the terms, as well as the working mechanism, are expressed in the subsequent sections. In what follows, all symbols will have the meaning given here unless explicitly stated to the contrary.

Actions	*Content*	*Context*
start(b, s)	none	the buyer b sends this message to start the negotiation with the seller s.
propose(i, y, ϕ)	the contract	the agent i sends a contract ϕ to the agent y.
accept(i, y, ϕ)	the contract	the agent i accepts the contract ϕ proposed by the agent y.
withdraw(i)	none	the agent i withdraws from the negotiation.
finalize(k)	none	the buyer finalizes the deal with the seller k.
decline(k)	none	the buyer declines the previous agreement made with the seller k.

TABLE 3.1: The interaction primitives.

3.4 The Concurrent Negotiation Model

This section describes the negotiation model that complies with the protocol of section 3.2. This description starts with the basic definitions and concepts of the model (see figure 3.4). Next, the concept of a *negotiation thread* is described, followed by a discussion of the *coordinator*.

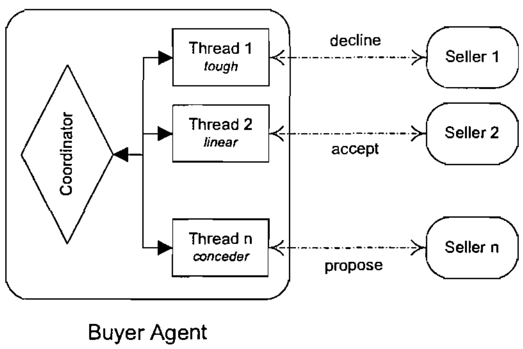


FIGURE 3.4: The model's components.

3.4.1 Definitions and Concepts

This section details the concepts of a negotiation *contract*, how it is structured and evaluated, and a *negotiation strategy*.

Symbols	*Description*
n	the number of seller agents participating in one encounter
i, y	a general agent, can be the buyer agent or any seller agent
b	the buyer agent
k	a generic seller agent
$s_k, k \in [1, n]$	a specific seller agent
A	the set of all participant agents
$\overline{s}$	the service under negotiation
ϕ	a general proposal, a contract
m	the number of issues in one proposal
j	a generic issue
D_j	the quantitative domain for the issue j
I	the set of negotiation issues
$\mathbf{A}^s$, $\mathbf{A}^c$, $\mathbf{A}^n$	the set of *generic*, *conceder* and *non conceder* seller agents (indicating different types of seller, according to our buyer agent's classification), respectively
S	the set of all the available negotiation strategies
$\mathbf{S}_c$, $\mathbf{S}_l$, $\mathbf{S}_t$	the set of *conceder*, *linear* and *tough* negotiation strategies (indicating different types of negotiation behaviors that our buyer agent may engage in response to the sellers' offers), respectively
t_{max}	the negotiation deadline of the buyer agent
$t_{i_{max}}$	the negotiation deadline of the seller i
U^i	the utility function of agent i
$U^i(\phi)$	the utility value of contract ϕ according to agent i's utility function
β	the parameter to decide which class a strategy will be (see section 3.4.1.3)
δ	the parameter to decide which class an evaluation function will be (see section 3.4.1.2)
α	the parameter to decide which class a seller will be (see section 3.4.3)
γ	the buyer's reservation value for all the negotiation threads

TABLE 3.2: The list of symbols (a **bold** symbol represents a set).

3.4.1.1 Contracts

As defined in section 1.1, a *service*, the object that the buyer and the seller are bargaining over, represents the capability of an agent in performing tasks. Here, a *contract* represents one agent's interest in a service to the other agent at a specific point in the negotiation encounter.

Each contract ϕ can have a single or multiple *issues*. The set of all the negotiation issues is called $\mathbf{I}$: $\mathbf{I} = \{issue_1, \ldots, issue_m\}$. The number of issues, m, in one contract is finite and is agreed upon by the buyer and all the sellers in the pre-negotiation phase and it is not changed throughout the negotiation encounters.

By means of an illustration, figure 3.5 represents a sample contract taken from the holiday scenario in section 1.2. This contract consists of 5 issues, namely *package price, direct flight, hotel rating, meals included* and *extra activities.* Each issue comprises a single value and its domain can be either finite (fixed number of choices) or infinite (a real number). At this time, for reasons of simplicity, we choose to use a quantitative representation for all negotiation issues (thus the domain of each issue can only contain numerical values)[4]. As can be seen, the domain for the issue *package price* is infinite, whereas the domains of the other issues are finite.

Issues	*Values*	*Domains*
package price	895.0	0.0 ... 1000.0
direct flight	0	0, 1
hotel rating	2	1, 2, 3, 4, 5
meals included	1	0, 1, 2
extra activities	0	0, 1

FIGURE 3.5: The sample contract with each issue's domain.

Formally, for agent $i \in \mathbf{A}$, each issue $j \in \mathbf{I}$ is a tuple $\{x^i_j, D^i_j, s^i_j, w^i_j\}$ where:

- D^i_j: is the value domain $\{x^i_{j_{min}}, x^i_{j_{max}}\}$ for j where $x^i_{j_{min}}$ and $x^i_{j_{max}}$ are the minimum and maximum acceptable values that x^i_j can have,
- x^i_j: is the current quantitative value of the issue $x^i_j \in D^i_j$,
- w^i_j: is the *weight*, or how i values j with regards to $\mathbf{I}$. The weights are normalized: $\sum_{j=1}^{m} w^i_j = 1$,
- s^i_j: is the step value of the issue:

[4]For qualitative issues, we choose to have their values represented by a corresponding quantitative representation. For each issue, this is done via a transformation function that converts each qualitative value into a quantitative numerical value within a predefined range (e.g. the issue *direct flight* and *extra activities*). Refer to [Faratin, 2001] for more details.

- for the issues with finite domains, the value of x^i_j can only be one of a number of finite options in D^i_j, but not all. In these cases, s^i_j specifies how x^i_j can be changed within D^i_j: x^i_j must either be of the form $x^i_j = x^i_{j_{min}} + k * s^i_j$ or $x^i_j = x^i_{j_{max}} - k * s^i_j$ where $k \in \mathbb{N}$. One example is the *hotel rating* issue: $D^i_3 = [1, 5]$ but x^i_3 can only take one value from the set $\{1, 2, 3, 4, 5\}$. In this case, s^i_3 is set to 1, allowing x^i_3 to have the desired values.
- for the issues with infinite domains, the value of x^i_j can be any value in D^i_j. In these cases, s^i_j is insignificant and its value is assumed to be 0 (e.g. x^i_1 can be any value in [0, 1000]; thus s^i_1 is set to 0).

Issues	*Values*	*Domains*	x	x_{min}	x_{max}	s	w
package price	895.0	0.0 ... 1000.0	895.0	0.0	1000.0	0.0	0.6
direct flight	0	0, 1	0.0	0.0	1.0	1.0	0.1
hotel rating	2	1, 2, 3, 4, 5	2.0	1.0	5.0	1.0	0.1
meals included	1	0, 1, 2	1.0	0.0	2.0	1.0	0.1
extra activities	0	0, 1	0.0	0.0	1.0	1.0	0.1

FIGURE 3.6: The internal representation of the sample contract.

The tuple $\{x^i_{j_{min}}, x^i_{j_{max}}, w^i_j\}$ represents agent i's preference over the issue j. To illustrate this quantitative issue representation, figure 3.6 shows the sample contract with all the issues fully expanded. As can be seen from this figure, the agent values the issue *package price* most (w_1 = 0.6) and considers the rest to have the same importance ($w_j = 0.1; j \in [2, 5]$).

3.4.1.2 Contract Evaluation

An agent differentiates different contracts by assigning them a score or a utility value [Binmore, 1992]. The higher the value, the better the contract is from the agent's point of view. In more detail, for each agent $i \in \mathbf{A}$ and issue $j \in \mathbf{I}$, i has a scoring function $U^i_j : D^i_j \rightarrow [0, 1]$, which represents i's preferences of j's value (x^i_j) with regards to its domain (D^i_j). The higher U^i_j is, the more i prefers j. Generally, we can always assume i's preference for increasing values of x^i_j are ascending (i.e. the higher the value of x^i_j, the higher the value of U^i_j). If this is not the case (e.g. for the issue *package price*, here the smaller the price the more the buyer prefers), we can simply reverse the value of the U^i_j function (i.e. $U'^i_j = 1 - U^i_j$).

In general, i's scoring function about x^i_j, U^i_j, can be of three types:

1. as the value of x^i_j goes from $x^i_{j_{min}}$ to $x^i_{j_{max}}$ linearly, the value of U^i_j increases linearly.

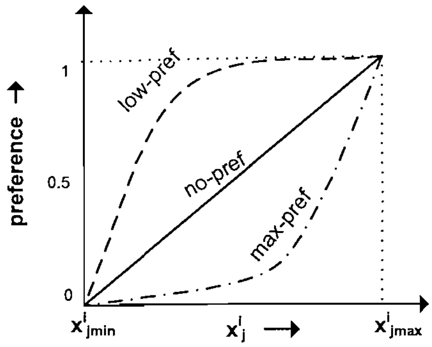


FIGURE 3.7: Three families of U^i_j functions.

2. as the value of x^i_j goes from $x^i_{j_{min}}$ to $x^i_{j_{max}}$ linearly, the value of U^i_j first increases slowly until x^i_j approaches $x^i_{j_{max}}$, at that time the value of U^i_j increases dramatically toward 1.

3. as the value of x^i_j goes from $x^i_{j_{min}}$ to $x^i_{j_{max}}$ linearly, the value of U^i_j first increases dramatically until x^i_j approaches $x^i_{j_{max}}$, at that time the value of U^i_j increases slowly toward 1.

We choose to have a function of U^i_j with one parameter δ to represent these types (see equation 3.1). Specifically, the instances or the function corresponding to the types are named: *no-pref* ($\delta = 1.0$), *max-pref* ($\delta < 1.0$) and *min-pref* ($\delta > 1.0$) (see figure 3.7).

$$U^i_j = \left(\frac{x^i_j - x^i_{j_{min}}}{x^i_{j_{max}} - x^i_{j_{min}}}\right)^{\frac{1.0}{\delta}} \tag{3.1}$$

Given $U^i_j, \forall j \in \mathbf{I}$, the formula for U^i, the *utility function* of agent i for the contract ϕ, is given in equation 3.2. This utility function is capable of consolidating all the individual preferences of the issues into a single value. Because the weights are normalized: $\sum_{j=1}^{m} w^i_j = 1$, the value of U^i is always in the range $[0, 1]$.

$$U^i(\phi) = \sum_{j=1}^{m} U^i_j(x^i_j) * w^i_j \tag{3.2}$$

To illustrate the utility function U^i, consider the *sigma* agent of section 1.2 with the sample contract ϕ in section 3.4.1.1. Assume *sigma* has no strong preference with regard to the individual issues' values and therefore chooses the *no-pref* for U^i_j functions ($\delta = 1.0$). Here, ϕ has 5 issues: *package price*, *direct flight*, *hotel rating*, *meals included* and *extra activities*. The individual U_j^i functions and the value for U^i are calculated as (from equation 3.1):

- $U_1^i = 1 - (\frac{0.895-0.0}{1.0-0.0})^{1.0} = 0.105$
- $U_2^i = (\frac{0.0-0.0}{1.0-0.0})^{1.0} = 0.0$
- $U_3^i = (\frac{0.25-0.0}{1.0-0.0})^{1.0} = 0.25$
- $U_4^i = (\frac{0.5-0.0}{1.0-0.0})^{1.0} = 0.5$
- $U_5^i = (\frac{0.0-0.0}{1.0-0.0})^{1.0} = 0.0$

Then by equation 3.2, the overall utility for the contract is:

$$U^i(\phi) = 0.105 * 0.6 + 0.0 * 0.1 + 0.25 * 0.1 + 0.5 * 0.1 + 0.0 * 0.1 = 0.138$$

3.4.1.3 Negotiation Strategies

This subsection looks at the behaviors of the buyer's threads during negotiation. These behaviors are specified by the negotiation strategies. Specifically, a negotiation strategy is the sequence of decisions that an individual thread will make during negotiation. These decisions could be either deciding the initial offer to send to the seller, selecting an offer to propose, accepting the offer proposed by the opponent or withdrawing from the negotiation (as per figure 3.8).

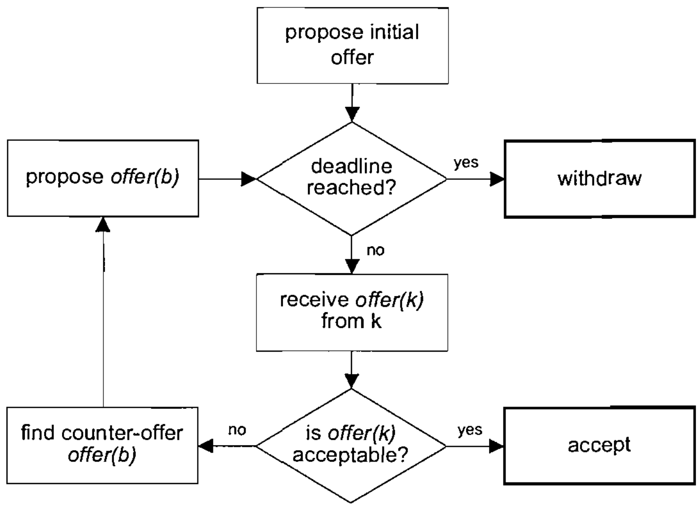


FIGURE 3.8: A general strategy structure.

In more detail, in order to decide whether to accept an offer ϕ^k (*offer(k)*) from a seller k, the values of all the issues of ϕ^k are compared against *the values domain for the set of negotiation issues* (see section 3.4.2). If all the issues' values are within this domain, the utility value $U(\phi^k)$ is calculated as described in section 3.4.1.2 and compared against *the reservation value* γ (see section 3.4.2). ϕ^k is then accepted if and only if $U(\phi^k) > \gamma$.

In order to generate an offer $\phi^t_{b\to k}$ (*offer(b)*) to propose to the seller, the offer's utility value U is calculated. Based on this value, a set of offers, Θ, that all have the utility value U, are generated. From this set, an offer ϕ is picked as the chosen offer to send to the seller. The specific way that this set is generated is not prescribed here but see section 3.5.2 for the specific exemplar view in the empirical evaluation.

With regard to the capability of finding the utility value of $\phi^t_{b\to k}$ to propose to the seller, we consider the set of negotiation strategies S to be composed of the class of time dependent strategies advocated by Faratin for bilateral negotiations in uncertain environments with time constraints (see section 2.2.1). These strategies are selected both for their ability to represent different relations between the negotiation time and the negotiation deadline, as well as their polynomial complexity. Specifically, these strategies fall into three broad categories, namely: *conceder* (S_c), *linear* (S_l) and *tough* (S_t) where $S = S_c \cup S_l \cup S_t$ (see figure 3.9). All of the strategies start with the same initial value that is generated in relation to the deadline and the reservation value. The *conceder* strategy quickly lowers its value until it reaches its reservation value. The *linear* strategy drops to its reservation value in a steady fashion. Finally, the *tough* strategy keeps its value until the deadline approaches and then it rapidly drops to its reservation value.

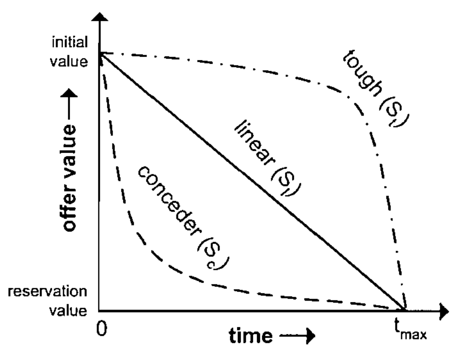


FIGURE 3.9: Strategy classes.

In more detail, these three categories of strategy can be expressed by one function: (see equation 3.3). At time $t \in [1, t_{max}]$, the utility value of the proposal that could be sent ($\phi^t_{b\to k}$) is calculated as:

$$U^b(\phi^t_{b\to k}) = \delta^b + (1-\delta^b) * (\frac{t}{t_{max}})^{\frac{1.00}{\beta}} \tag{3.3}$$

where δ^b is the initial proposal utility value (depending on the chosen strategy), $t \in [1, t_{max}]$ is the specific time period at which the counter-offer needs to be found and β is the parameter that decides the shape of the function (see section 3.5.2 for specific values of β).

3.4.2 The Negotiation Threads

An individual negotiation thread is responsible for dealing with an individual seller agent on behalf of the buyer. Each such thread inherits its preferences from the buyer agent and has its negotiation strategy specified by the coordinator (see section 3.4.3). Specifically, the preferences that each thread inherits from the buyer agent are:

- *the negotiation deadline*: All the threads have t_{max} as their negotiation deadline,
- *the values domain for the set of negotiation issues*: This is the set $\mathbf{D} = \{D_1, D_2, \ldots, D_m\}$ where $D_j = \{x^b_{min_j}, x^b_{max_j}\}; j \in \mathbf{I}$ (see section 3.4.1.2). Any proposal from a seller k is considered *valid* if and only if $\forall j \in \mathbf{I} : x^k_j \in D_j$,
- *the reservation value* γ: The thread only accepts a *valid* proposal $\phi^t_{k\to b}$ from the seller k at time t if $U^b(\phi^t_{k\to b}) > \gamma$.

The values domain is the criteria each thread uses to decide whether an offer from a seller satisfies its basic requirements. Each issue is considered to be *valid* if its value lies in the domain. If all the issues are valid, the offer is considered *valid.* If, however, one issue is invalid then the offer is rejected. The value of γ represents the highest utility value of an agreement that the buyer has already accepted from a seller (it is initially set to 0, meaning that the buyer does not yet have any agreement). The buyer can always revert back to this agreement even if all other negotiations fail. Thus, even when an offer from another seller is *valid*, it may not be accepted by the thread. Specifically, an *acceptable* offer is only accepted by the thread if its utility value γ' is greater than γ. If this is the case, the thread will notify the coordinator about this new and better agreement. Then γ' will replace γ as the new reservation value for all other ongoing threads.

A more detailed view of the structure of each thread is presented in figure 3.10. As can be seen, each thread is composed of three subcomponents, namely *communication*

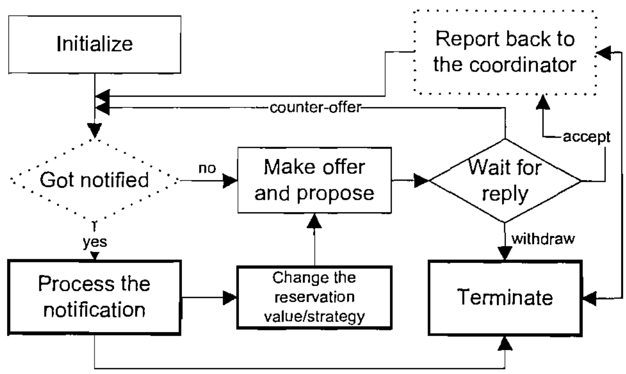


FIGURE 3.10: A single negotiation thread.

(represented by the dotted lines), *process* (represented by the bold lines) and *strategy* (represented by the normal lines). The *communication* subcomponent is responsible for communicating with the coordinator (as shown in section 3.1). Before each round, it checks for incoming messages from the coordinator and if there are any, it passes them to the *process* subcomponent. After each round, it reports the status of the thread (including the proposals and the deal's value if an agreement is reached) back to the coordinator. The *process* subcomponent processes messages from the *communication* subcomponent. This can either be changing the reservation value γ or changing the strategy (see section 3.4.3 for more details). The *strategy* subcomponent is responsible for making offers/counter-offers, as well as deciding whether or not to accept the offer made by the seller agent. It uses the reservation value as the basis for deciding whether to accept the seller's offer; in this case any offer with a value greater than this reservation value is accepted, otherwise a counter-proposal is made (unless the deadline has passed in which case a withdraw is sent).

3.4.3 The Coordinator

The coordinator is the most important component of the buyer agent (see figure 3.4). It is responsible for synchronizing all of the negotiation threads' behaviors and choosing an appropriate negotiation strategy for each thread. This section details its design.

The first task of the coordinator is to select the initial negotiation strategies for the threads. It is important to select the right negotiation strategies at the start because choosing inappropriate negotiation strategies can lead to poor outcomes.

Prior to the encounter, the coordinator assumes some information about the classification classes of the available sellers that the threads will negotiate with (e.g. the toughness of the sellers) in order to help it select the strategies (see section 2.5 for more detail). This information is given as a probabilistic distribution over the classification classes. If no information is available, all the classes are assumed to have the same probabilistic distribution ($\frac{1}{\text{number of classes}}$).

In addition, there are two matrices that aid the coordinator's decision: the *percentage of success matrix (PS)* and the *pay off matrix (π)*. The former measures the chance of having an agreement as the outcome of the negotiation when the buyer applies a strategy to negotiate with a specific type of the seller (e.g. when applying a tough strategy with a tough seller, the chance of having an agreement is 15%). The latter measures the average utility value of the agreement reached in similar situations (e.g. when applying a tough strategy with a tough seller, the average utility value of the agreement, once reached, is 0.7). The values of these matrices are updated after all the negotiation threads have finished.

Given this information, the coordinator calculates the probability of the first seller belonging to a specific class and, based on this, the expected utility of applying the corresponding strategy to handle this particular seller. It then selects the strategy $\lambda \in \mathbf{S}$ that provides the highest expected utility and applies λ to the thread that will handle the negotiation with this agent. The expected utility function, EU(λ), that the coordinator uses is defined as:

$$EU(\lambda) = \sum_{a}^{\mathbf{A}} PS(\lambda, a) \cdot \pi(\lambda, a) \cdot P(a) \tag{3.4}$$

where $P(a)$ is the probability that the seller agent belongs to specific class a. After finishing with the first seller, the coordinator uses Bayes rule (see section 2.5.3) to update the probabilistic distribution it has and continues with the second seller. This process is repeated until the coordinator finishes allocating the strategies to all the negotiation threads.

To illustrate this idea, consider the following example. Assume that the coordinator has the following data prior to negotiation:

- there are 100 participating sellers: $n = 100; A^s = \{s_1, s_2, \ldots, s_{100}\}$.
- there are three available negotiation strategies that it can select for a given thread: $\mathbf{S} = \{S_c, S_l, S_t\}$.

- there are two types of sellers: $\mathbf{A} = \{A^c, A^n\}$
- the classification information about the sellers is: the probability that the first seller is a conceder, $P(s_1 \in A^c)$ or $P(c)$, is 0.45 and the probability that the first seller is a non-conceder, $P(s_1 \in A^n)$ or $P(n)$, is 0.55.
- the values of the matrix *PS* are given in table 3.3

PS	A^c	A^n
S_c	0.35	0.75
S_l	0.25	0.28
S_t	0.6	0.15

TABLE 3.3: The sample matrix PS.

- the values of the matrix π are given in table 3.4

π	A^c	A^n
S_c	0.5	0.4
S_l	0.35	0.4
S_t	0.7	0.65

TABLE 3.4: The sample matrix π.

Based on this information, the values for EU functions are calculated, using equation 3.4, as follows:

- EU(S_c) = 0.35 * 0.5 * 0.45 + 0.75 * 0.4 * 0.55 = 0.2438
- EU(S_l) = 0.25 * 0.35 * 0.45 + 0.28 * 0.4 * 0.55 = 0.1010
- EU(S_t) = 0.6 * 0.7 * 0.45 + 0.15 * 0.65 * 0.55 = 0.2426

As can be seen, because strategy S_c gives the highest expected utility value, it is chosen as the strategy for the first thread. In other words, the highest expected utility is achieved when considering seller s_1 as a non-conceder. The probability distribution $P(n)$ is then calculated as (from equation 2.5):

$$P(n|A^s \setminus s_1) = \frac{P(A^s \setminus s_1|n)P(n)}{P(A^s \setminus s_1)} = \frac{(n-1)\cdot P(n)}{n} = \frac{99 \cdot 0.55}{100} = 0.5445$$

Since there are only two types of seller: conceder and non-conceder, we have $P(c) = 1 - P(n)$.

$$P(c|A^s \setminus s_1) = 1 - P(n|A^s \setminus s_1) = 0.4555$$

Here, $P(c|A^s \setminus s_1)$ is understood as the probability that the second seller is a conceder. Similarly, $P(n|A^s \setminus s_1)$ is understood as the probability that the second seller is a non-conceder. Again, the values for EU functions for the second seller are calculated, using equation 3.4, as follows:

- EU(S_c) = 0.35 * 0.5 * 0.4555 + 0.75 * 0.4 * 0.5445 = 0.2431
- EU(S_l) = 0.25 * 0.35 * 0.4555 + 0.28 * 0.4 * 0.5445 = 0.1008
- EU(S_t) = 0.6 * 0.7 * 0.4555 + 0.15 * 0.65 * 0.5445 = 0.2444

As can be seen, because strategy S_t gives the highest expected utility value, it will be chosen as the strategy for the second thread. The coordinator keeps applying these steps again and again, until it finishes allocating the strategies to all the threads.

The second task of the coordinator is to coordinate the threads, specifically, to monitor the reservation value γ (see section 3.4.2). During the encounter, whenever the coordinator receives the status that a negotiation thread has accepted an offer from a seller with the utility value γ', it replaces the value of γ with γ' and sends a request to change the reservation value to all other ongoing negotiation threads (as detailed in section 3.4.2).

The final task of the coordinator is to classify the sellers during negotiation and change the negotiation strategies for the threads. The coordinator does this to make sure the threads use appropriate strategies when bargaining with the sellers. For example: if seller s_1 is willing to concede to reach an agreement, it is possible for b to exploit s_1 by keeping b's offer consistently low; thus, if an agreement can be made, it will be of higher value for b than if b does not follow this strategy. On the other hand, if seller s_2 consistently keeps its offer value high after a number of negotiation periods, b may need to negotiate in a more concessionary fashion in order to come up with a deal.

From the buyer's viewpoint, the sellers are characterized based on the utility value of their proposals. Because the seller's evaluation functions or strategies consist of private information, the value of the proposals that are sent to the buyer are the only information the buyer can use to characterize the sellers. In more detail, the sellers are classified into two categories, namely *conceder* and *non-conceder*. At time t: $2 < t \leq t_{max}$, called the *analysis time*, the coordinator tries to determine if a given seller is a *conceder* or a *non-conceder*. In particular, assume O^k_τ is the utility value of the offer that seller agent k made at time $\tau : 1 \leq \tau \leq t$ according to the buyer agent's preferences. Then seller k is considered a *conceder* if $\forall \tau \in [3, t] : \frac{O^k_\tau - O^k_{\tau-1}}{O^k_{\tau-1} - O^k_{\tau-2}} > \alpha$ where α is the threshold value set on concessionary behavior. Similarly, seller k is considered a *non-conceder* if $\forall \tau \in [3, t] : \frac{O^k_\tau - O^k_{\tau-1}}{O^k_{\tau-1} - O^k_{\tau-2}} < \alpha$. If the agent falls into neither category, it is judged as not classified.

Let the set of *conceder* and *non-conceder* agents be represented by A^c and A^n, respectively. Now, given the set of strategies $\mathrm{S} = \{\mathrm{S}_c, \mathrm{S}_l, \mathrm{S}_t\}$ and the set of agents $\mathrm{A} = \{b, \mathrm{A}^c, \mathrm{A}^n\}$, the coordinator changes the strategy for each negotiation thread based on the type of the agent it is negotiating with, in order to try to obtain better outcomes. Agents belonging to the set A^c are willing to concede in order to end up with agreements. Therefore, if the agent negotiates toughly with some of them (keeping its offer consistent), it may obtain a deal that has a higher value than if it continues negotiating in its present manner. However, if the agent negotiates in this way with all the agents, it may not obtain any deals at all. Therefore, for reasons of balance, the agent will negotiate in a tough manner with a subset of the agents ($P^c_t\%$) in A^c. For the remainder of the agents in A^c, the strategy remains unchanged. Similarly, if the agent believes a particular agent is in the set A^n then in order to make sure it obtains a deal with some of them, it makes some of its own threads more conciliatory. Thus, for the agents belonging to the set A^n, a percentage of them ($P^n_c\%$) will have their behavior made conciliatory, while the remainder have their strategies unchanged. There is no change to agents whose behavior cannot be classified.

Having defined the model, the next step is to see how it performs in different contexts so that its relative advantages and disadvantages can be ascertained. This analysis will be empirical in nature and is reported in the next section.

3.5 Evaluation

This section evaluates our concurrent model (eCN) previously described in this chapter in a range of different environments and assesses its performance in terms of the utility value of the final agreement and the number of agreements achieved. In this work, *empirical evaluation* is used as the method of measurement for a number of reasons. Firstly, because this model is heuristic-based, its behavior cannot be theoretically predicted (see section 2.2). Secondly, there are a number of internal variables which control the behavior of our model as well as external variables which define the environment in which our model is being used (see section 3.5.2). These variables are interrelated and need to be considered in a broad range of situations. Empirical techniques allow us to manipulate these variables, conduct the experiments and analyze the results. Thus, they are suitable for our evaluation purpose. In more detail, the evaluation technique we use is called *exploratory studies* [Cohen, 1995]. With this method, *general hypotheses* are formed to express the intuitions about the causal factors within our model. The *experiments* are then conducted and generate the results that either support these hypotheses or go against them.

The structure of this evaluation section is as follows: the experiment sets (e.g. the set of both internal and external variables being tested) are discussed in section 3.5.1. The design principles (e.g. how to conduct the experiments and how the results are gathered from these tests) are described in section 3.5.2. The assumptions about the seller agents are detailed in section 3.5.3. Finally, the *hypotheses* are presented and evaluated in sections 3.5.4, 3.5.5 and 3.5.6.

3.5.1 Experimental Setup

There are three sets of experiments in this evaluation. The first set is designed to compare the performance of our concurrent model with a sequential negotiation model, which has the same setup but where all the negotiations happen sequentially (the details are expressed later in this section). This sequential model is chosen as a control in order to emulate the traditional way of doing negotiations when there are multiple providers. By evaluating the results of the two models, the advantages, as well as the disadvantages, of negotiating concurrently can be shown. In more detail, this set of experiments deals with the requirements associated with deadlines and computational tractability (see section 1.2).

There are a number of internal variables that control the behavior of our model. In each negotiation environment in which our model is evaluated, these variables have different effects. Therefore, the second set of experiments is designed to find the most influential variables and to find out how to set their values in order to gain the best overall performance in each environment. With this experiment set, the requirements of *incomplete information* and *computationally tractable* (see section 1.2) are evaluated.

The final set of experiments is designed to compare the performance of our model against the only other concurrent model in the literature: Rahwan et al's (see section 2.2.3). We experiment with both models in the same environment and plot the results achieved. Here, the requirements of *mutually influence* and *efficient negotiation outcomes* (see section 1.2) are evaluated.

3.5.2 Design Principles

As previously mentioned, the main purpose of this chapter is to evaluate the negotiation model. Since the evaluation method is empirical, the results of our model running in different environments are gathered and compared with other (standard) results to highlight the cases where our model performs either well or poorly. From these cases, hypotheses can be stated about our model's specific characteristics.

In our evaluation, the variables or parameters that have their values set by the experimenter are called the *independent variables* (see table 3.5), while those that have their values gathered after the experiments are called the *dependent variables* (see table 3.6) [Cohen, 1995]. In either case, these variables must either be of type *categorical* (each variable belong to a specific category), *ordinal* (the variables can be ranked but the distance between two variables is meaningless) or *interval* (the variables can be ranked and the distance between two variables is meaningful) [Cohen, 1995].

Variables	*Descriptions*	*values*
n	the number of seller agents	[1,30]
m	the number of negotiation issues	[1,8]
t_{max}	the negotiation deadlines of the agents	[5,30]
$x^i_{j_{min}}$	minimum acceptable value for an issue	[0, 20]
$x^i_{j_{max}}$	maximum acceptable value for an issue	[30, 50]
w^i_j	the weight of an issue	$\frac{1}{m}$
δ	the parameter for the buyer's evaluation function (see section 3.4.1.2)	1
β	the parameter of the negotiation strategy (see section 3.4.1.3)	[0.01,0.2], [0.95,1.05], [10,20]

TABLE 3.5: The independent variables.

The environments in which our model are run would ideally cover all the possible environments that could realistically happen. However, according to [Friedman, 1984], every theory must involve some simplification, as none can include all the possible features of reality. This is also the case for our evaluation. Although our model is not theoretically bounded (e.g. the number of participating sellers has no upper limit, nor does the number of issues or the negotiation deadlines), we choose to ignore extreme situations that we believe are unlikely to happen in our target environment. Even though one can argue that these extreme situations could happen, we believe it is unlikely to be the case.

Variables	*Descriptions*
number(T)	the total amount of time required to complete the negotiation
number(P)	the total number of proposals exchanged during negotiation
PI	the performance improvement of the concurrent model
U	the utility value of the final agreement
N	the number of successful negotiations

TABLE 3.6: The dependent variables.

In our experiments, each environment is characterized by the independent variables, including the number of seller agent participants, n, the number of negotiation issues, m, the deadlines for the agents and the preferences of the agents about the set of negotiation issues. Since there are an infinite number of possible environments, selecting a finite subset of these is necessary to assess the performance of our model. To this end, n is simplified to be in the range of [1, 30] and m in the range of [1, 8]. Recall from section 3.4.1.1 that agent i's preference for issue j is represented by the tuple $\{x^i_{j_{min}}, x^i_{j_{max}}, w^i_j\}$. The tuple $[x^i_{j_{min}}, x^i_{j_{max}}]$ is an interval independent variable whose scale is infinite. To simplify this problem, we randomly set the value for $x^i_{j_{min}}$ in the interval of [0, 20] and $x^i_{j_{max}}$ in the interval of [30, 50]. The values for w^i_j are simply set to $\frac{1}{m}$, meaning that the weight of all issues have the same value. The negotiation deadline for each agent is an ordinal independent variable, whose value is randomly chosen, ranging from 5 (very short deadline) to 30 (long deadline).

The second simplification we use in our evaluation is our strategic parameters. In evaluating individual issues (see section 3.4.1.2), the buyer uses the *no-pref* function with the value of δ set to 1.0. The three categories of negotiation strategies (see section 3.4.1.3) have the following ordinal scale for the value of β: [0.01,0.2] for *boulware*, [0.95,1.05] for *linear* and [10,20] for *conceder*. These ordinal scales are chosen to represent the characteristic shape of each category.

In order to highlight the benefit of negotiating concurrently, we decide to test our model against a traditional negotiation model, in which all the negotiations happen sequentially. In this model, the number of participants, together with their preferences, are exactly the same as in our model. The only difference between two models is the way the buyer handles the negotiations with the sellers. In the sequential model, the buyer starts with the first seller. After this negotiation is completed, the buyer negotiates with the second seller and carries on this way, until it finishes negotiating with all of them[5]. If the buyer reaches an agreement of value p in negotiation i, then in all subsequent negotiations, p will be its new reservation value.

To illustrate the operation of the sequential negotiation model, we use the sample scenario described in section 1.2. Assume that in the concurrent model, *sigma* uses 10 threads to concurrently negotiate with the sellers $\{s_1, s_2, \ldots, s_{10}\}$ and each thread k uses strategy $strategy_k \in$ S. The sequential model is then constructed to have 10 negotiations processing sequentially. First, *sigma* negotiates with s_1 using strategy $strategy_1$ until it finishes with s_1, then it carries on with s_2 using strategy $strategy_2$, and so on until it finishes with s_{10}. Prior to each negotiation k, if γ is the utility value of the highest value agreement reached so far, γ will be used as the reservation value for negotiation k.

During the evaluation, the performance of the concurrent model is assessed by comparing the utility value of the final agreement with the one from the sequential model. Assume U_c and U_s are the utility values of the final agreements from the concurrent and sequential model, respectively; then the relative *performance improvement* of the concurrent model over the sequential one, or *PI*, is calculated as $\frac{(U_c - U_s)}{U_s}$. This equation indicates the difference in percentage terms of the two agreements' values; if *PI* is positive, the concurrent model outperforms the sequential one and vice versa if *PI* is negative.

The results are gathered from a series of experiments in different environment settings. Each experiment consists of a number of negotiation episodes (either 1000 or 2000), each such episode is a complete run of our eCN model in a single negotiation environment. The results are averaged and put through a regression test to ensure that all differences are significant at the 99% confidence level.

PS	A^c	A^n
S_c	0.25	0.75
S_l	0.25	0.25
S_t	0.75	0.25

TABLE 3.7: The initial matrix PS.

[5]We overcome the sellers' ordering problem of sequential negotiations (see section 1.2) by running the sequential model n times and averaging the utility value achieved. In each run, the negotiation order of the sellers is randomly re-generated.

Prior to each experiment, the initial values for the matrix *PS* and π are given in tables 3.7 and 3.8, respectively. These values represent the initial assumption about the strategies' distributions and will be updated after each negotiation episode.

π	A^c	A^n
S_c	0.5	0.5
S_l	0.5	0.5
S_t	0.5	0.5

TABLE 3.8: The initial matrix π.

Also, prior to each negotiation episode, there is a 50% chance that the probability distribution of the sellers (classified according to their selected negotiation strategy, see section 3.5.3) is given to the buyer agent. If it is not the case, no probability distribution information is given to the buyer.

3.5.3 The Seller Agents

In our evaluation, each seller is characterized by 3 independent variables, namely (i) *the values domain for the set of negotiation issues*, (ii) *the negotiation strategy* and (iii) *the negotiation deadline*. As can be seen, there are an infinite number of possible sellers, similar to the negotiation environments. To simplify this problem, we select the value for each variable as follows:

- *the values domain for the set of negotiation issues*: This is the seller's preferences for the service (similar to the ones of the buyer's thread described in section 3.4.2). These domains are randomly generated so that each domain does intersect with the corresponding domain of the buyer's preference. For example, if the buyer's value domain for an issue j is $[a, b]$ then the corresponding value domain for the seller will be generated as $[c, d]$ that satisfies $a \leq c \leq b \leq d$.
- *the negotiation strategy*: Each seller is assigned a random strategy selected from a predefined list of alternatives. This set is composed of time-dependant and behavior-dependant tactics (as per section 2.2.1) with randomized values for each tactic's attribute. The negotiation strategies of the sellers are fixed and do not change throughout the negotiation episode[6].

[6]Future work will consider the situations where the seller agents also adapt their strategies to the behaviors of the buyer agent.

- *the negotiation deadline*: The deadline for each seller is generated in a similar fashion to the one of the buyer (ranging from 5 to 30).

The values for these variables are generated prior to each negotiation episode and kept unchanged throughout that period.

We now turn to the specific hypotheses.

3.5.4 Comparing Concurrent and Sequential Negotiations

Here, we are interested in the performance of the eCN model and its sequential counterpart. Thus, both models are experimented with under the same evaluation setup and the results are recorded to be compared. The actual results are detailed follows.

HYPOTHESIS 1. The time to complete the negotiation will be less for the concurrent model than for the sequential one [7].

EVALUATION. Figure 3.11 shows the percentage of time saved by performing the negotiation concurrently as compared with sequentially. As can be seen, this saving is proportional to the number of participating agents. The reason for this saving is that by negotiating concurrently, the time consumed for all the threads is not more than the largest deadline of the sellers and the buyer. Each agent is only allowed to continue the negotiation until its deadline at which point it must stop. Thus, for each negotiation thread with a seller k, the longest period it is allowed to continue is $min(t_{max}, t_{k_{max}})$. As a result, our model's negotiation will stop at the latest period $t = min(t_{max}, max(t_{1_{max}}, t_{2_{max}} \ldots t_{n_{max}}))$. On the other hand, when negotiating sequentially, the buyer needs to wait for the first negotiation to finish, then it can start the second negotiation and so on. In general, each negotiation could last until the deadline $min(t_{max}, t_{k_{max}})$. Thus, the overall negotiation could finish at $t = min(t_{max}, t_{1_{max}}) + min(t_{max}, t_{2_{max}}) + \ldots + min(t_{max}, t_{n_{max}})$ in the worst case.

HYPOTHESIS 2. The number of proposals that will be made in the concurrent model is less than the number in the sequential one.

EVALUATION. Figure 3.12 shows the number of proposals saved (as a percentage) by performing the negotiation concurrently compared with sequentially. As can be seen,

[7]Here, we assume that each time unit is one negotiation round instead of the actual execution time. Nonetheless, the model is able to handle a large number of sellers (1000) in real-time using multi-threaded java implementation in a typical development PC (see hypothesis 11).

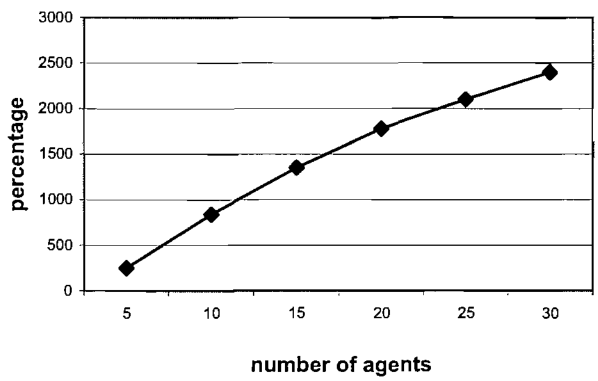


FIGURE 3.11: Percentage of time saved.

the average number of proposals made in the concurrent model is less than the number in the sequential model and this difference increases in proportion to the number of participating agents. From our experiments, we observe that the total number of agreements reached in the concurrent model is more than in the sequential model (greater than 50% more). This means, in the concurrent model, more negotiations terminate before their deadlines have elapsed. Thus, the total amount of time it takes to complete all the negotiations in the concurrent model is less than in the sequential model. As the number of proposals made in each negotiation is relative to the time it takes to complete the negotiation, the number of proposals made in the concurrent model is less.

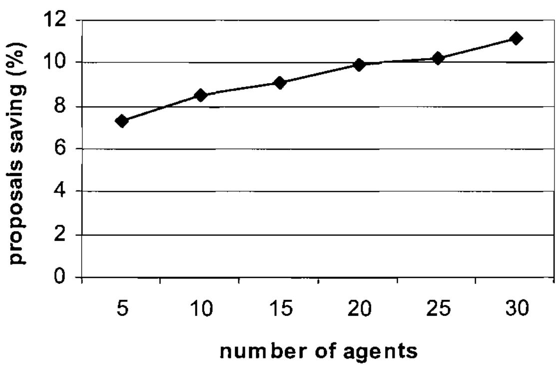


FIGURE 3.12: Percentage saving in the number of proposals sent.

HYPOTHESIS 3. To realize the benefits of concurrent negotiations, the buyer agent's deadline cannot be too short.

EVALUATION. Figure 3.13 shows the performance of the concurrent model with different values of the buyer's deadline. This shows the longer the buyer's deadline, the better the performance improvement. This occurs because if the buyer's deadline is short (less than 15 units in this case), the time when one negotiation thread reaches an agreement is necessarily close to the deadline. Thus, it has little effect on the other negotiation threads. On the other hand, if the buyer's deadline is longer, once a negotiation thread finishes with an agreement, it can be used to influence the other threads. Hence, this would give better deals. However, when the buyer deadline is too long, the change in the subsequent performance improvement is less. This is because by the time an agreement has been reached in one thread that is close to the buyer's deadline, a number of other threads may have already terminated due to their deadlines having been reached. This results in the newly reached agreement having little effect on our model. As a result, the changes in performance improvement of our model are only marginal.

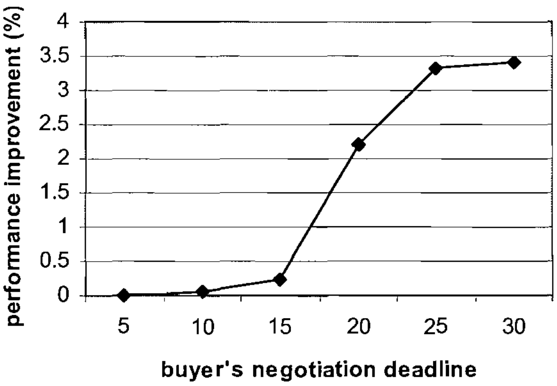


FIGURE 3.13: Performance versus deadline.

HYPOTHESIS 4. The final agreements reached by the concurrent model will have, on average, higher or equal utility for the buyer than those of the sequential model (assuming the deadline is not too short).

EVALUATION. When a thread negotiates with a seller, it tries to find an offer that is acceptable to that particular seller. This is achieved by exploring the space of agreements in some way. In the sequential model, each thread starts with a reservation value γ_s, which can either be the initial preference value (if this is the first encounter or no agreement has yet been reached) or the value of the previous best agreement reached in an earlier negotiation. Moreover, this value remains unchanged until the particular negotiation thread finishes. On the other hand, in the concurrent model, each thread starts with a reservation value γ_c ($\gamma_c = \gamma_s$ at $t = 0$), which is the initial preference value of the agent. This value may then be changed during the course of the negotiation as a result of an agreement obtained in another thread. This, in turn, narrows the space of agreements for the buyer to only those that have a higher utility value than the current reservation value. Hence, if the buyer reaches an agreement, the utility value of this agreement will be greater than the one it already has. Assuming all the threads have sufficient deadlines, whenever an agreement is reached, the search space of all the concurrent negotiation threads will be reduced simultaneously. Thus, on average, the buyer strives to reach a higher utility value for a greater proportion of the negotiation time with more sellers than it does in the sequential model. This means the performance of the concurrent model is often better than the sequential one.

However, in some cases, by narrowing the space of agreements, no intersection with the seller's space of agreements may be found in the concurrent model. Therefore, the agents will not be able to reach an agreement and so the utility value of the final agreement is reduced. In these cases, the overall performance of the concurrent model will be less than its sequential counterpart. Our experimental results indicate that in all environmental settings, on average, the results of the concurrent model are better than the sequential, ranging marginally from 1.5% to 2.5% depending on the number of participating sellers (see figure 3.14).

Moreover, it can also be seen that the performance of the concurrent model decreases slightly as the number of sellers increases. This is because in the sequential model, the more sellers that participate, the higher the chance the buyer will have in finding an agreement. In some cases, this results in obtaining a higher value for the final agreement. On the other hand, the number of participating sellers does not have that strong an impact on the performance of the concurrent model. As a result, the differences between the value of the agreements reached by the concurrent and the sequential model in these cases are reduced. Thus, the overall performance improvement of the concurrent model decreases.

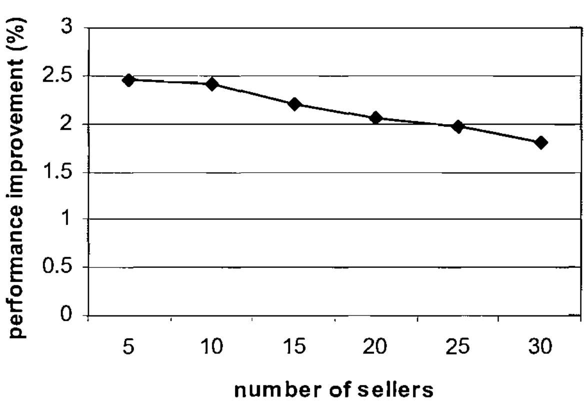


FIGURE 3.14: The improvement rate.

As can be seen, by negotiating concurrently, the time elapsed to complete each negotiation episode is significantly reduced compared to negotiating sequentially. Furthermore, the buyer agent is also able to obtain a higher number of successful negotiations and a better utility value for the final agreement once reached. Next, we aim to find the most influential variables and to find out how to set their values in order to gain best overall performance in each environment.

3.5.5 Tuning the Performance of the Concurrent Model

This section is concerned with the internal performance of our model or, alternatively, how different parameters affect our model's performance in a single environment. As described in section 3.4, eCN is composed of many parameters. These parameters are designed so that the eCN is flexible in different environments. The values of these variables must be set appropriately, otherwise, our model may not perform well. Given this, it is important to find the right combinations of the variables in each environment. To do this, we run eCN with different values for a particular parameter in a single environment and report the results in the following hypotheses.

HYPOTHESIS 5. Changing the strategy in response to the agent's assessment of the ongoing negotiation is equal or better than not doing so.

HYPOTHESIS 6. Assuming that it is beneficial to change the strategy during the negotiation, the analysis time should be moderately early (to have time to have some effect), but not too early (so that it is reasonably accurate).

EVALUATION. To evaluate these hypotheses, we varied the *analysis time* (see section 3.4.3) relative to the buyer's deadline (see figure 3.15). As can be seen, the best performance improvement is obtained when the sellers are analyzed about a third of the way into the negotiation period. This is sufficiently near the beginning to be able to have an effect on the rest of the negotiations, but sufficiently far into the encounter to make a reasonable approximation about the type of the negotiation opponent. With respect to hypothesis 5, the outcome of analysis time equal to 100% is equivalent to an agent that does not change its strategy during the encounter. As can be seen, this leads to poor outcomes and so changing (at any time) is not worse, and in most cases, is beneficial.

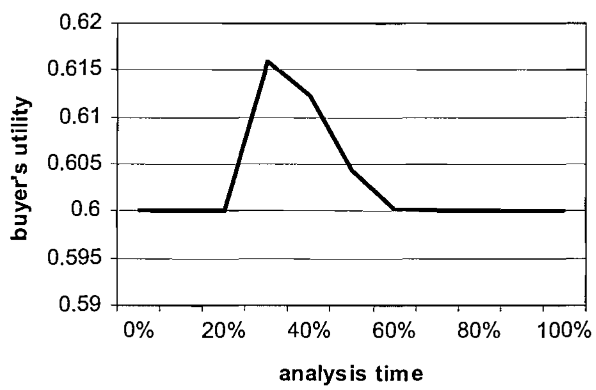


FIGURE 3.15: Performance versus analysis time.

HYPOTHESIS 7. When dealing with sellers in A^c, the tougher the buyer negotiates the better the overall outcome it will obtain.

EVALUATION. To evaluate this hypothesis, we varied P_t^c (see section 3.4.3) through all possible values. To this end, figure 3.16 shows the more tough the agent is, the better the outcome it obtains. This is because when dealing with a *conceding* seller, if the buyer keeps its offer consistent, as the deadline approaches, the seller will quickly lower its proposal value close to its reservation value (if it has a deadline shorter than that of the buyer thread[8]). Thus, if an agreement is reached at this point, its utility value for the buyer will be higher than that obtained if the buyer adopts any other strategy.

On the other hand, the value of P_c^n (see section 3.4.3) does not have a strong impact on the performance of our model. This is because if a seller is not willing to concede in order to find an agreement (e.g. it keeps its proposal value close to its maximum), it does not matter how the buyer behaves and no agreement will be reached.

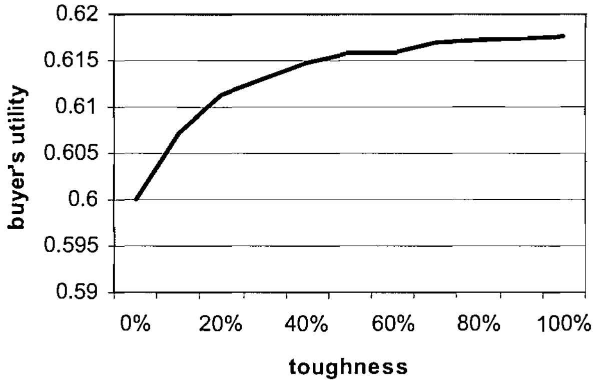


FIGURE 3.16: Performance versus degree of response toughness.

HYPOTHESIS 8. When analyzing the seller, different values for the conceder threshold have different effects on the outcome of eCN.

EVALUATION. To evaluate this hypothesis, we varied α (see section 3.4.3) through all possible values (values that are greater than 1.5 or less than 0.1 do not have an impact on the sellers' classification process). Figure 3.17 shows the results.

As can be seen, as the value of α goes up, the utility of the final agreement for the buyer decreases. This is because the higher the conceder threshold is, the fewer the number of conceder sellers there will be. Now, since the buyer can press the more conciliatory sellers for high value deals (see hypotheses 7), if that number is decreasing, so will the utility of the final agreement. On the other hand, it can also be noticed that the buyer will gain the highest number of agreements if the threshold is set to 0.5. Which means, in general, the value of α should be set to 0.5 in order to give the buyer the highest possible number of final agreements together with an acceptable utility value.

[8]If the seller has a longer deadline then no agreement will be reached.

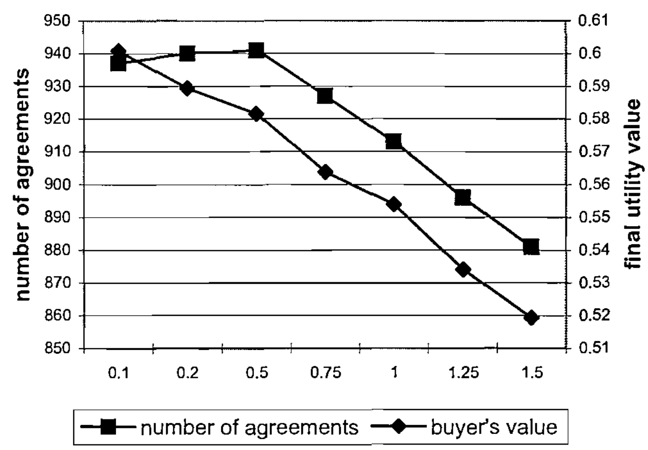


FIGURE 3.17: Performance versus different levels of conceder threshold.

Next we compare the performance of our model with the only other concurrent negotiation in the literature (section 2.2.3). In particular, we run both models in the same environment and measure the differences, in terms of both number of successful negotiations (N) together with the average utility value of the final agreement (A).

3.5.6 Performance Against Rahwan's Model

Having evaluated our model against the sequential negotiation case and having obtained the best set of parameters that it can use, we now consider its performance against a more realistic benchmark. Specifically, the controls we used in this set of experiments are the optimal solution ***(optimal)***, *desperate **(D)***, *patient **(P)*** and *optimized patient **(OP)***. In more detail, the optimal mechanism operates in a perfect information situation in which the agents know the preferences and strategies of other agents. Given this, the buyer agent is able to find the Pareto optimal agreement for each thread if such an agreement exists. If no such agreement exists, the utility of that thread is considered to be 0. The individual agreement that maximizes the buyer agent's utility is then selected as the optimal solution. The other three controls are based on the theoretical work of Rahwan et al. (see section 2.2.3), which is the only other extant model that deals explicitly with concurrent encounters. Basically, ***D*** terminates all the negotiations whenever an agreement is found in any one thread, ***P*** waits until all the negotiations finish and then selects the highest value agreement as the final answer, and ***OP*** extends ***P*** in that whenever an agreement is found, its value is broadcast to all other ongoing threads so that they will not accept a lower value agreement.

HYPOTHESIS 9. The concurrent model will achieve more and higher utility agreements than the controls.

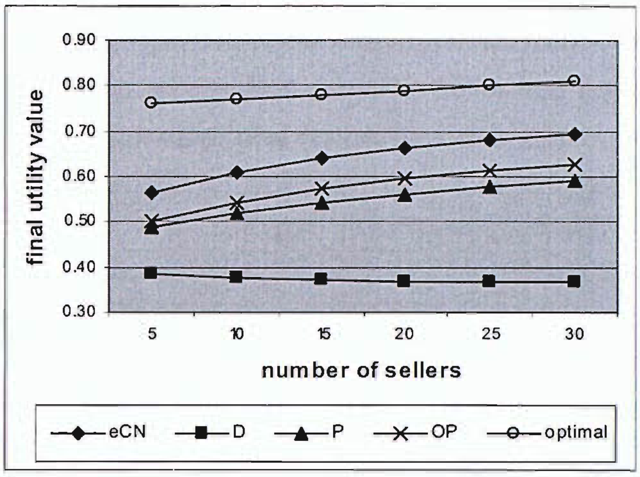


FIGURE 3.18: Final utility value for varying numbers of sellers.

EVALUATION. To evaluate this hypothesis, we average the utilities achieved with varying numbers of seller agents and varying deadlines. The results of eCN, the controls and the optimal are displayed in figure 3.18. As can be seen, our model is between 6-8% better than the closest control and between 11-21% lower than the optimal. Amongst the controls, ***D*** has the worst utility (since it terminates whenever an agreement is reached), ***P*** has a better utility (since it waits until all the negotiations finish and selects the best deal), and ***OP*** provides the best utility (since it is an improved version of ***P***). Consequently, from now on, we will only focus on ***OP*** as the main point of comparison.

Fundamentally, the eCN model differs from the others in the way that the buyer agent behaves both prior to and during the negotiation process. Unlike the controls, in which the strategy employed by the buyer stays constant throughout the negotiation episode, each negotiation thread in our model will change its strategy if it believes there is a benefit in so doing. Recall from section 3.4.3, our agent selects its initial strategies based on its beliefs about the opponents that it is likely to encounter[9]. Specifically, as the sellers have different objectives, they are likely to behave differently. Some desperately want to sell their services, while others will only agree to a deal if it will benefit them more than what they already have. Since we do not know the exact characteristic of each seller, our initial strategy selection is not guaranteed to be accurate. However, we overcome this problem by reclassifying the sellers during the negotiation process (based on their actual behaviors rather than the general market beliefs). Based on this

[9]Naturally, these beliefs may not necessarily be true. Therefore we examine the effect of the accuracy of this initial selection on our model's performance in hypothesis 12.

classification, some of the threads change their strategies. In some cases, this flexible behavior of the buyer agent helps it finds high value agreements that would not have been found otherwise. Consequently, this increases the buyer's utility and leads to an improvement in our model's performance. The improvement in utility is particularly marked when the buyer can recognize a conceder seller and can negotiate in a very tough manner to obtain a high value deal (as per hypothesis 5).

No of sellers	**5**	**10**	**15**	**20**	**25**	**30**
eCN	1418	1615	1710	1762	1802	1830
OP	1389	1593	1690	1744	1776	1804
optimal	1686	1827	1887	1908	1928	1946

TABLE 3.9: Number of successful negotiations.

In terms of agreements made, as can be seen from table 3.9, our model produces more agreements than the others. This improvement can also be explained by the adaptive nature of our strategy selection. Compared to the controls, the number of times our strategy selection leads to a conflict is lower than the number of times it leads to an agreement. This, in turn, leads to a modest increase in the number of successful negotiations.

HYPOTHESIS 10. The larger the number of participants, the closer the utility produced by the concurrent model is to the optimal.

EVALUATION. Here, we measure the differences in the results obtained by the eCN model with the optimal as the number of participants increases. The results with respect to utility and number of agreements are displayed in table 3.10.

No of sellers	**5**	**10**	**15**	**20**	**25**	**30**
U(*eCN*)	0.56	0.61	0.64	0.66	0.68	0.70
U(*eCN*) $t_{b_{max}} \geq 15$	0.62	0.67	0.70	0.72	0.75	0.76
U(***optimal***)	0.76	0.77	0.78	0.79	0.80	0.81
N(*eCN*)	1418	1615	1710	1762	1802	1830
N(*eCN*) $t_{b_{max}} \geq 15$	1499	1672	1753	1797	1830	1860
N(***optimal***)	1686	1827	1887	1908	1928	1946

TABLE 3.10: Buyer's performance with varying numbers of sellers.

As can be seen, the greater the number of participating sellers, the closer our result is to the optimal. Specifically, the gap between the results decreases from 26% to 13% as the number of sellers increases from 5 to 30. This is explained by the fact that the buyer only finalizes the deal with the seller that provides the highest value deal. Thus, as the number of sellers increases, so does the number of agreements reached by the threads. Since these agreements are used to influence other ongoing threads, the utility value of the final agreement will be improved. This is also the situation for the number of successful negotiations. With 5 sellers, the agent only succeeds in 84% of the encounters. However, this rate increases to nearly 94% when there are 30 participating sellers. Furthermore, if we only consider cases where the buyer has a sufficient deadline (larger than 15 units in this case), our results come even closer to the optimal (the gap decreases from 18% to 6%, whereas the success rate is increased from 89% to 96%). Again this is mainly due to the accuracy of our classification process (see hypothesis 3).

HYPOTHESIS 11. As the number of participating sellers increases, the concurrent model will produce broadly similar results to those of hypotheses 9 and 10.

EVALUATION. To evaluate this hypothesis, we increase the maximum number of participating sellers from 30 to 1000 and record the outcome of eCN. To this end, figure 3.19 and 3.20 show the corresponding number of agreements made and the average final utility value, respectively.

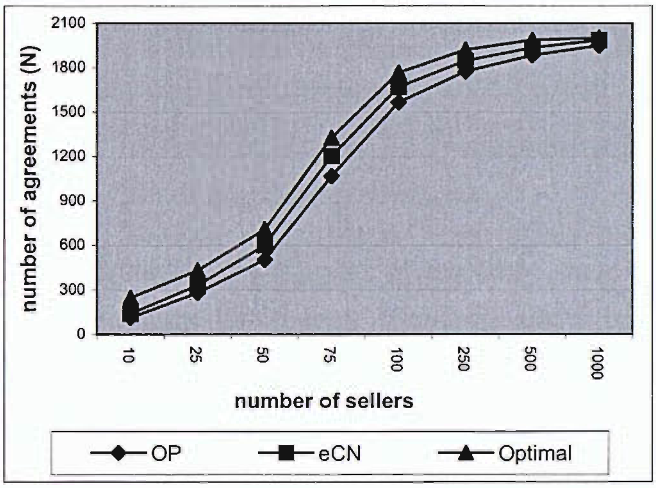


FIGURE 3.19: Number of agreements vs number of sellers.

As can be seen, the predictions of hypotheses 9 and 10 are correctwhen the number of seller agents are scaled up to 1000. Moreover, the gap between our model's results and those of the optimal is reduced as the number of sellers increases. This is the case for both the number of successful negotiations, as well as the average utility value of the final agreement reached. This occurs because as the number of sellers increases, the buyer agent will have more choices in securing a good value deal. Although the performance improvement is somewhat marginal, it is still able to outperform the controls. This indicates that our model is likely to be most effective when there are small to medium numbers of agents.

HYPOTHESIS 12. The more accurate the agent's information about the probability distribution of agent types, the better the performance of the concurrent model.

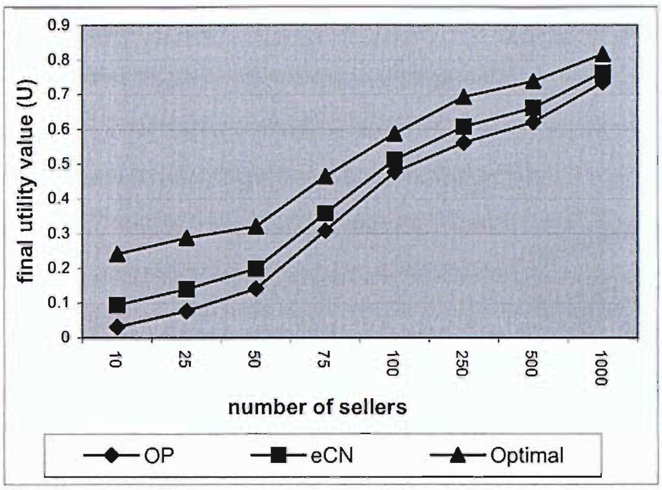


FIGURE 3.20: Final utility value vs number of sellers.

EVALUATION. To ensure our model can perform robustly in unpredictable environments, this set of experiments evaluates its reliance on the accuracy of information an agent holds about the marketplace. Specifically, we consider the degree to which the probability distribution P (defined in section 3.4.3) matches reality and what impact this has on the initial selection of negotiation strategies.

The initial selection of strategies is only part of the story since the buyer agent can reclassify the sellers and change its strategy. Nevertheless, it can be observed from figure 3.21[10] that the accuracy of this information does indeed have an effect on the result of the process, albeit by a small figure (1-2%). In our experiments, about 9% of the agreements were reached in threads before the sellers' classification occurs and some of these initial agreements become the final solutions at the end of the encounter process. Thus, the aforementioned small improvement was made by improving these early agreements.

3.6 Practical Applications

One of the key motivations behind the development of this model was that it should be applicable in practical contexts. To this end, eCN has been used in two real-world applications.

[10]Here, the *unknown* plot corresponds to the case where P has equal values throughout, *50%* to the case where half of the values in P are correct, and *100%* is where P reflects the actual strategies of the sellers. ***OP*** does not use P in its decision making and ***optimal*** also operates with the correct values for the sellers' strategies.

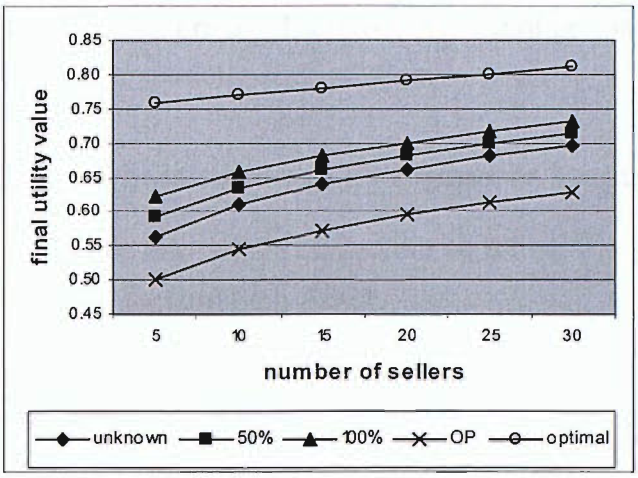


FIGURE 3.21: The accuracy of the belief versus the performances.

First, this basic concurrent negotiation model has been successfully implemented in a commercial project (iService) developed by BT [Thompson *et al.*, 2004]. This project focuses on automatically handling web services among customers. Here, a customer can request for a number of services and iService will try to provide and govern these services (see figure 3.22). In this case, our negotiation model is responsible for finding an alternative replacement for a service provider if the selected one ceases functioning. Specifically, when an agent that provides a particular web service to the client stops functioning (for whatever reason), eCN was called to search for an alternative provider from the pool of potential candidates.

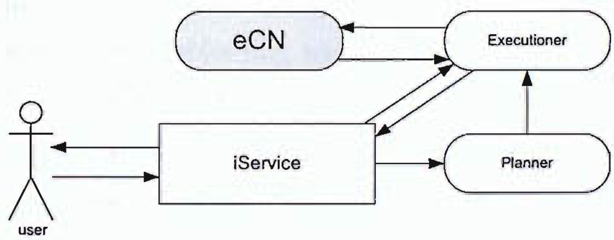


FIGURE 3.22: The iService framework.

In more detail, iService operates in a multi-agent environment that is composed of a number of different service providers. Now each such provider can provide a range of services. In this context, end users can submit their requests by telling iService which services they want and the constraints that need to be satisfied (e.g. time, quantity and so on). Based on this information, iService tries to provide the users with solutions to their requests, in terms of composing the offerings from a number of service providers and, later, executing the composite solution. For example, user A might request to have

ADSL installed in his office by Friday. Then from the list of providers that is available to A, iService might propose the following solution:

1. A BT phone line will be installed by a BT engineer on Tuesday.
2. A service check on the line will be carried out by BT on Wednesday to ensure its quality is appropriate for installing ADSL.
3. The ADSL service will be installed on Thursday by AOL.

Now, assume that when this composite solution is carried out, the first two steps complete successfully, but the third one fails because AOL is suddenly unable to provide the desired ADSL service. As it is expensive and inappropriate to replan the composite solution, an alternative solution for the third step must be found. In particular, an alternative service provider must be found that has the similar capabilities (ADSL provider) as the failed service provider (AOL in this case) and that satisfies the constraints created by both the user and the partially executed solution (i.e. it has to start and finish on the same day). In such cases, there are likely to be a number of alternative potential providers in the marketplace, each of which can provide the same ADSL service but with varying installation prices and degree of availability for the same day service. In order to find the most appropriate provider, eCN is then invoked to simultaneously negotiate with each of them. By negotiating in this way, iService is able to find the most suitable alternative service provider in the shortest time possible.

Second, our model has also been used in the CONOISE project [Norman *et al.*, 2003]. CONOISE is concerned with establishing and maintaining virtual organizations (VOs) and has been applied to both a personalizing media packaging scenario [Norman *et al.*, 2004] and a Grid scenario [Shao *et al.*, 2004]. In the context of CONOISE, eCN is responsible for the dynamic operation phase of the VO, when it needs to extend (by adding a new member) or modify its structure (by replacing an old member).

Specifically, CONOISE involves a number of agents working together to achieve the objectives of the virtual organization. Once the VO has been formed, a number of agents are recruited to provide an agreed set if services under an agreed service level agreement contract. Now, if nothing changes, then this grouping should be able to fulfill its commitments. However, in such complex and dynamic systems this is rarely the case. In particular, there are two situations that need particular attention:

1. The user submits a new (additional) requirement that cannot be provided by the current VO. Thus, a new agent needs to be added. Moreover, since the current

VO is already in operation, it is not sensible to completely reform the VO just to incorporate this new requirement.

2. One agent in the current VO ceases to provide a required service. This will typically disrupt the operating phase of the current VO in that a particular service may no longer be available or there may be insufficient capacity of the desired service. Thus, in either case, an alternative agent needs to be found and this needs to be done in a timely fashion.

Now in both of these situations, negotiation provides a feasible solution for managing the restructuring of the VO with minimal interruption. Again it makes sense to conduct these negotiations concurrently and so eCN is employed to this end.

3.7 Summary

This chapter has outlined the design of our negotiation model for managing concurrent bilateral encounters. The model itself is composed of two main components: the coordinator and the negotiation threads. Each negotiation thread is responsible for bargaining with a specific agent and all of the threads are controlled by the coordinator. The coordinator attempts to classify the agents it is negotiating with before and during negotiation and then attempts to apply appropriate strategies according to each specific agent type. Each component was designed so that it is computationally tractable and it is flexible when applied into different environments.

The eCN model developed in this section was evaluated by conducting a series of experiments and producing empirical results. Our aim in doing this was to assess the effectiveness of this model in a range of different scenarios. Specifically, we showed that the performance of eCN is better, in terms of the overall utility value of the agreement, than that achieved from a sequential model. We also showed that the amount of time and the number of proposals required to complete the negotiation is less in eCN.

Moreover, eCN gives better results than the only other concurrent model in the literature, both in terms of the number of agreements reached and the average utility value of the final agreement. This result remains even when the number of participating sellers is scaled up. In order to obtain such results, it was necessary for the agent to change its negotiation strategy in response to the seller's behavior in a particular negotiation. In particular, when dealing with conceder sellers, the tougher the agent negotiates, the better the overall outcome it gains. The model has also been applied in two large-scale projects in the areas of web service procurement and virtual organization management.

In the next phase of our research, we extend our concurrent model with the capability of handling commitments among the participating agents. This helps increase the desirability for the sellers to participate in the negotiation encounter and also makes the eCN model more robust and flexible. The details of this are given in the next chapter.

Chapter 4

Flexible Commitments in Concurrent Negotiations

This chapter presents an extension to the basic negotiation model described in chapter 3. In particular, it aims to integrate a commitment handling capability into the negotiation model so that the agents will have more flexible and realistic behaviors with respect to the way in which they handle the contracts they make. In particular, we remove the unrealistic assumption of the previous chapter that seller agents cannot renege on their commitments. Specifically, the underlying motivation is to achieve the desired commitment capabilities that were discussed in section 1.2.

Thus far, our model has operated under the assumption that all the agreements made during the negotiation encounter are binding on the sellers but not on the buyer (see section 3.1). This assumption was considered necessary because it allows the buyer to accept multiple agreements and, later, select the one that maximize its profit. It was also considered because it enables us to focus on the the buyer's reasoning component when there is greater certainty. However, this assumption is clearly biased towards the buyer and, thus, may make the seller agents less inclined to join the negotiation. Furthermore, it is not applicable in most realistic settings since rarely is the buyer given such a privileged position. Thus, we introduce a commitment handler to relax this constraint by allowing any agent to unilaterally renege on its commitment[1].

To this end, this chapter is structured in the following way. First, the commitment protocol is introduced in section 4.1. The buyer's internal commitment strategies are

[1]Here we do not look into the problem of how these commitments will be enforced since it is not the main focus of this research. Instead, we assume that there will be an external institution that with enforce these restrictions on the participating agents [Dellarocas, 2000].

then discussed in section 4.2. Then, section 4.3 details the results of applying the protocol and the buyer's commitment strategies in various negotiation settings, focusing on the impact of handling commitments to the concurrent model. Finally, section 4.4 concludes.

4.1 The Commitment Protocol

In this research, the model we used is based on the leveled commitment contracts (see section 2.4.3). This is chosen because of its ability to allow the agents to unilaterally decommit from their previously acknowledged agreements for any reason they deem appropriate (the importance of being able to do this is argued for in section 1.2). However, we cannot just take this model as is because there are a number of associated problems that need to be rectified before we can apply it in our concurrent context (again, see section 2.4.3). Therefore, we have modified Sandholm and Lesser's commitment protocol and combined it with our negotiation protocol (from section 3.2). The details of our extended protocol are given in sections 4.1.1 (where we discuss our modified negotiation protocol) and 4.1.2 (where we discuss our commitment protocol), respectively.

However, before we detail the protocols, we define the basic terms that are used to describe the commitment model (see table 4.1). Specifically, this table gives a brief description of their meaning and the detailed explanations follow in the subsequent sections.

4.1.1 The Extended Negotiation Protocol

The negotiation protocol described in section 3.2 was designed to handle the basic negotiation scenarios where the agreements can be arbitrarily made by the participating agents[2]. Specifically, it gave the buyer agent the advantage of committing to a number of agreements simultaneously, without worrying about the future consequences of over committing itself (recall, the buyer is allowed to select the single highest value agreement at the end of the encounter and decline all the other previously accepted agreements without any penalty to itself). This is clearly inappropriate for the reasons noted above. Thus, the negotiation protocol should be modified so that in any one round, there is a maximum of one agreement that can be made from any agent. This is to ensure that

[2] If an offer is considered acceptable by both agents, it will be selected as an agreement for both agents. However, this agreement is binding on the seller only, giving our buyer the right to renege at the end of the negotiation.

Symbols	*Description*
$U(\phi, t)$	the utility value of contract ϕ at time t according to the buyer agent's utility function
$U^i(\phi, t)$	the utility value of contract ϕ at time t according to agent i's utility function
ρ	the penalty fee that one agent needs to pay another agent if it decides to break a previously made commitment (see section 4.1.2)
ρ_0	the initial penalty value (see section 4.1.2)
ρ_{max}	the final penalty value (see section 4.1.2)
μ	the degree of acceptance for an offer, for the buyer to consider whether or not to accept an offer from a seller (see section 4.1.2)
τ	the threshold value for μ (see section 4.1.2)
ω	the maximum number of concurrent agreements that the buyer will take at any one time (see section 4.1.2)
$\boldsymbol{\Omega}$	the set of currently holding commitments of the buyer (see section 4.1.2)

TABLE 4.1: The list of commitment symbols (a **bold** symbol represents a set).

in that particular round, the buyer will not end up with more than one agreement and then, later, have to pay an unnecessary decommitment penalty.

In our previous negotiation protocol, an agreement is made in one step when an agent accepts the offer made by the other agent (by sending an agreement message). However, when the buyer agent sends out a number of counter-offers to different sellers, there is the possibility that more than one of them will return an agreement. Thus, in order to ensure that a maximum of one agreement can be made at any one round, a more elaborate procedure for accepting an agreement is needed. In this work, we use a two step approach for reasons of simplicity and efficiency. Now, an agreement is created only if an acknowledgement message is received after the agreement is sent. Thus, there is now an option for any agent to send a withdraw message after the agreement has been received to indicate that no agreement is made. In either case (acknowledge or withdraw), that particular negotiation will be terminated.

Specifically, in each thread, the negotiation with the corresponding seller agent will be carried out via the negotiation protocol detailed in figure 4.1. Compared to our negotiation protocol previously described in section 3.2, there are now six states in this protocol in which state 1 is the initial state and states 4 and 6 are the termination states. Specifically, the buyer's thread starts the negotiation (state transits from 1 to 2) by proposing an initial offer to the opponent (state transits from 2 to 3). At this point, the opponent has three options:

1. accept the offer (state transits to 5), the buyer is notified and either accepts the offer (state transits to 6) or withdraws (state transits to 4). If the buyer also accepts the offer, the seller will wait for the buyer to finalize the deal (see section 3.4.1);
2. withdraw from the negotiation (state transits to 4) which signals the termination of the negotiation;
3. propose a counter-offer to the buyer (state transits to 2) which causes the negotiation to move to the next period.

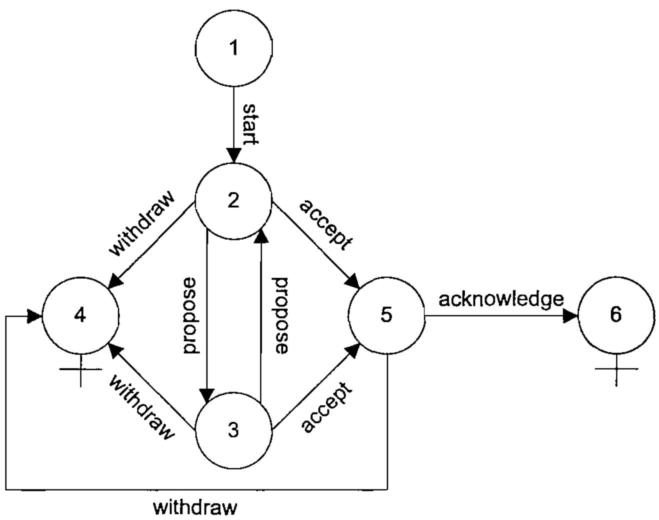


FIGURE 4.1: The extended negotiation protocol.

4.1.2 Extended Leveled Commitment Contracts

Once the buyer agent and a seller decide to reach an agreement in a particular negotiation (after an acknowledgement is received), that agreement is considered as an intermediate commitment contract. It is considered binding on both agents until either the execution time (the buyer's deadline) is reached or one agent decides to drop out. Here, an intermediate commitment contract is defined as:

Definition 4.1. An intermediate commitment contract C between the buyer agent b and a seller k at time t_k is a tuple $C(\phi, t_k) = (\phi, t_k, \rho_0, \rho_{max})$, where

- ϕ is the contract that has been agreed by both parties b and k (see section 3.4.1.1).
- t_k is the time when the acknowledgement between b and k is received.
- ρ_0 is the initial penalty (the fee to pay if the deal is broken at contract time, t_k).
- ρ_{max} is the final penalty (the fee if the deal is broken at execution time, t_{max}).

Here, ρ_0 and ρ_{max} are two values that have been agreed by the buyer agent and all the participating sellers before the encounter begins. Unlike the original leveled commitment contract, here an agent will pay different amounts of penalty fee, depending on the time when it decides to break its commitment. The motivation behind this is that if an agent decides to break a commitment early, it will be easier for the other agent to find an alternative. Thus, the decommitting agent does not have to pay as much penalty as if it decides to break at a later time. For this reason, ρ_0 is typically set at a lower value than that of ρ_{max}.

Previously, the seller agent was not allowed to renege from its contract, only the buyer agent could (simply by sending a decline message to the appropriate seller agent). Now, all the agents are treated equally. Thus, if one decides to decommit, for whatever reason, it has to pay a fee to the other agent. To this end, if an agent i decides to decommit at time $t < t_{max}$ from a commitment contract $C(\phi, t_k)$, the penalty fee it has to pay is calculated as follows:

$$\rho(t) = U^i(\phi, t) \times \left(\rho_0 + \frac{t - t_k}{t_{max} - t_k} \times (\rho_{max} - \rho_0)\right) \tag{4.1}$$

This equation is chosen to represent a linear relationship between the penalty value with the time the contract is broken. As can be seen, the closer the break time is to the execution deadline, the larger the penalty fee will be (assuming that $\rho_{max} \geq \rho_0$). If, however, the relationship between the penalty value and the broken time is not linear, any other formula can be used (the only condition is that the closer the break time is to the execution deadline, the larger the penalty fee will be).

By means of an illustration, consider the following example. Assume the buyer's deadline (t_{max}) is 10, the initial penalty (ρ_0) is 5% and the final penalty (ρ_{max}) is 10%. Now let a deal with an expected utility value 0.58 be made at time 6. If, at time 7, the buyer wants to decommit, by (4.1), it has to pay:

$$\begin{aligned} \rho(t) &= 0.58 \times \left(0.05 + \frac{7-6}{10-6} \times (0.10 - 0.05)\right) \\ &= 0.58 \times 0.0625 \\ &= 0.03625 \end{aligned}$$

whereas if it decommits at time 9 then it will face a larger penalty:

$$\begin{aligned} \rho(t) &= 0.58 \times \left(0.05 + \frac{9-6}{10-6} \times (0.10 - 0.05)\right) \\ &= 0.58 \times 0.0875 \\ &= 0.05075 \end{aligned}$$

4.2 The Buyer's Commitment Strategies

Since the buyer agent now has to pay a fee every time it breaks a contract, it cannot simply just agree on all deals and, later, select the highest value one as the final agreement (as it did in chapter 3). Thus, when presented with a potential agreement from a specific seller, the buyer has to decide whether it should take this deal or reject it. By following the extended negotiation protocol (as per section 4.1.1), the buyer can eliminate the chance of committing to unnecessary intermediate agreements in any one round by simply selecting the acceptable offer that has the highest value in that particular round as the contract to commit to.

Furthermore, the buyer agent also has to deal with the situation in which it receives what is an acceptable offer but at an early time in the negotiation, which has a comparatively low utility value. In such situations, it has to decide whether it should take this offer or wait for a better one. Since that offer is acceptable to the buyer, it means that there exists an intersection of all the negotiation issues for both agents. Thus, there are potentially more than one possible agreements from this negotiation. However, since the buyer does not have any information about the seller's reservation values or its negotiation strategy (see section 1.2), it is not able to know whether that offer has the highest utility value in that perspective (since there might be other potential offers with higher utility values from that particular negotiation). If it decides to take the offer, it may lose the chance of obtaining a higher value deal at a later time[3]. However, if the buyer decides to wait for the next offer, there is also the chance that no further deal will be forthcoming because the next offer will fall outside of the agreement zone or the seller might have reached its negotiation deadline (the buyer also has no information about this). To capture this decision problem, when presented with a contract ϕ that has utility value of $U(\phi, t)$ from seller k at time t, the buyer will accept ϕ as an intermediate contract if all the following conditions are satisfied:

1. If it already has another commitment $C(\phi', t_{k'})$ with another agent k' and this deal has not been broken, the utility gained by taking this new offer must be greater than that of the current deal, after having paid the decommitment fee. That means $U(\phi, t) > U(\phi', t_{k'}) + \rho(t)$[4].

2. The *degree of acceptance*[5] (μ) for ϕ must be over a predefined threshold (τ). This threshold specifies how the buyer should accept the offers, whether it is *greedy* (likely to accept any possible deal) or *patient* (only accepts deals that provide a certain expected utility value). Here, μ is calculated by comparing the utility value of ϕ with the predicted utility value of the next set of contracts from other sellers (described below), also taking into account the relation between the current time and the buyer's deadline. Specifically, the formula for calculating μ is:

$$\mu(\phi) = \frac{U(\phi, t) - \rho(t)}{max\{U_{exp}(k_i, t) \mid k_i \in A_s \setminus k\}} \times \frac{t}{t_{b_{max}}}, \tag{4.2}$$

where $\rho(t)$ is the decommitment fee that the buyer has to pay if it has already committed to a deal with another seller (if it has not, $\rho(t)$ is 0) and $U_{exp}(k_i, t)$ is the predicted utility of the next proposal from seller k_i. The value of $U_{exp}(k_i, t)$ is calculated as:

$$U_{exp}(k_i, t) = U(k_i, t) + \frac{d_U(t, t-1)}{d_U(t-1, t-2)} \times |d_U(t, t-1)|, \tag{4.3}$$

where $d_U(t_1, t_2)$ is the distance, in terms of utility value, between two offers from seller k_i at times t_1 and t_2: $d_U(t_1, t_2) = U(k_i, t_1) - U(k_i, t_2)$.

[3]This is possible since seller agents typically start from their reservation value and lower their requirements step by step until either the lower limit is reached or an offer is accepted. Thus if the buyer waits, the next offer might be of higher value than the current one.

[4]If the buyer is currently holding more than one commitment (discussed later in this section), the utility gained by taking this new offer must be greater than that of any of the currently committed deals (after paying all the associated penalties).

[5]It represents the buyer's evaluation about an offer from a particular seller with respect to the issue of commitment.

To illustrate the operation of the buyer commitment reasoning process in more detail, consider the following example. Assume there are 4 participating sellers, the buyer's deadline ($t_{b_{max}}$) is 6, the initial penalty (ρ_0) is 10%, the final penalty (ρ_{max}) is 20%, and the threshold (τ) is 0.8. Assume the buyer has committed on a deal with seller 4 at time 2 with the expected utility value of 0.21. The utility values of previous offers from all the sellers are displayed in table 4.2.

agent	t=1	t=2	t=3
k_1	0.03	0.12	0.16
k_2	0.01	0.04	0.10
k_3	0.1	0.19	0.23
k_4	0.11	0.21	-

TABLE 4.2: Utility values of the offers.

Now at time 3, the buyer has to decide whether it will accept the offer $\phi(k_3)$ from seller k_3. Since it is already committed to a deal with k_4, if it wants to take $\phi(k_3)$, it will have to pay a decommitment fee to k_4. By (4.1), the fee it has to pay is:

$$\rho(3) = 0.21 \times \left(0.1 + \frac{3-2}{6-2} \times (0.2 - 0.1)\right) = 0.026$$

As can be seen, $U(k_3, 3) < U(k_4, 2) + \rho(3)$, so the first condition is violated. Thus, the buyer will reject $\phi(k_3)$ and remain with its commitment to k_4.

agent	t=1	t=2	t=3	t=4	t=5
k_1	0.03	0.12	0.16	0.28	0.4
k_2	0.01	0.04	0.10	0.30	0.26
k_3	0.1	0.19	0.23	0.31	0.36
k_4	0.11	0.21	-	-	-

TABLE 4.3: Utility values of the offers (cont.).

At time 4, however, seller k_4 decides to renege on its current deal and pay the decommitment fee to the buyer. According to equation (4.1), it has to pay:

$$\rho(4) = 0.21 \times \left(0.1 + \frac{4-2}{6-2} \times (0.2 - 0.1)\right) = 0.0315$$

As can be seen, this decommitment from k_4 leaves the buyer with no agreement. Now, at time 5, the buyer has to decide if it should take up the offer from k_1 (table 4.3 shows the utility values of the offers from all the sellers). Since it has no intermediate agreement, the first condition is satisfied. To evaluate the second condition, the buyer first calculates the value for $U_{exp}(k_1, 5)$ and $U_{exp}(k_2, 5)$ using (4.3):

$$U_{exp}(k_2, 5) = 0.26 + \frac{-0.04}{0.2} \times 0.04 = 0.252$$

$$U_{exp}(k_3, 5) = 0.36 + \frac{0.05}{0.08} \times 0.05 = 0.391$$

The value of $\mu(\phi(k_1))$ is then calculated, using equation (4.2), as:

$$\mu(\phi(k_1)) = \frac{0.4}{0.391} \times \frac{5}{6} = 0.852$$

This time, since $\mu(\phi(k_1)) > \tau$, the buyer will commit to this deal. It keeps on bargaining in this way until its deadline is reached. If, at that time, there is an intermediate deal that has not been broken, this deal is selected as the final agreement. If, however, no such deal exists, the negotiation is considered unsuccessful and terminated without an agreement.

From the above example, it can be seen that selecting different values for τ will cause the buyer to have various degrees of acceptance for any incoming offer. This will, in turn, directly affect the performance of the model. There are various possible values for τ, however, only two of these really cause a major difference in the result (see hypothesis 16). Thus, in what follows, we propose two commitment strategies for the buyer: the first one is *greedy* where τ is set to 0 (which lets the buyer commit to any possible deal) and the second one is *patient* where τ is set to 0.5 (which will prevent the buyer from committing to early deals with lower utility values and will cause it to wait for a better opportunity).

Up until this point, we have only considered the situation where the buyer agent commits to a maximum of one intermediate contract at one time. However, it is possible for the buyer to commit to more than one contract at any one time and then, later, select the best one and decommit from the others. This represents a cautious approach and avoids the risks associated with committing to only one contract which is then revoked near the deadline, leaving the agent with insufficient time to find a replacement. The downside of this approach, however, is that if the sellers do not renege, the buyer may end up paying a significant part of its utility value as the penalty fee. Nevertheless, in some cases it may be beneficial for the buyer to consider the option of having multiple commitments during the bargaining process. To capture this, assume that the maximum number of commitments that the buyer will hold at any one time is $\omega \geq 1$ and let Ω be the set of contracts that the buyer is currently committed to: $|\Omega| \leq \omega$. Assuming that at time t, there is an offer ϕ from seller k that has $\mu(\phi(k)) > \tau$ (see equation 4.2[6]) and $U(\phi(k), t) > U(\phi', t_{k'}) + \rho(t) \quad \forall C(\phi', t_{k'}) \in \Omega$ then this offer will be accepted by the buyer. This, in turn, means the following steps will be taken:

1. If Ω is not full (i.e. $|\Omega| < \omega$), $C(\phi, t)$ will be added to Ω: $\Omega = \Omega \bigcap C(\phi, t)$
2. If Ω is full (i.e. $|\Omega| = \omega$) select $C(\phi', t_{k'}) \in \Omega$ that has the minimum value of $U(\phi', t_{k'})$ and decommit from that contract. Then, $C(\phi, t)$ will be added to Ω: $\Omega = \Omega \bigcap C(\phi, t)$

Now if a seller k that has a contract $C(\phi(k), t_k)$ decides to withdraw its commitment, that contract will be subtracted from Ω: $\Omega = \Omega \setminus C(\phi(k), t_k)$. Then at the end of the bargaining process, if there is more than one contract in Ω, the buyer simply selects the one that has the highest utility value as the final agreement and decommits from all the others.

Having defined the commitment protocol, the next step is to see how it performs in different contexts so that its relative advantages and disadvantages can be ascertained. As before, this analysis will be empirical in nature and is reported in the next section.

4.3 Evaluation

This section evaluates the extended commitment model described in this chapter in a range of different environments. Here we assess its performance in terms of the utility value of the final agreement and the number of agreements achieved (as per chapter 3). Once again, *empirical evaluation* is used as the method of measurement for the same reasons as outlined in section 3.5. Again, *general hypotheses* are formed to express the intuitions about the causal factors within the model. The *experiments* are then conducted and generate the results that either support these hypotheses or go against them. The structure of this section consists of two parts: the experimental setup is described in section 4.3.1 and the *hypotheses* are presented and evaluated in section 4.3.2.

[6]If $|\Omega| < \omega$, $\rho(t)$ is considered to be 0. If not, $\rho(t)$ is the penalty the buyer will have to pay to break from the contract $C(\phi', t_{k'}) \in \Omega$ that has the minimum value of $U(\phi', t_{k'})$.

4.3.1 Experimental Setup

This evaluation focuses on the effect of being more flexible with respect to commitments, particularly on the *commitments* requirement stated in section 1.2. For this reason, we reuse the basic experimental environment from section 3.5.2 and only add/update the commitment related settings. Specifically, the penalty fee (both initial and final) is an ordinal independent variable, whose value is randomly chosen, ranging from 5% (small) to 100% (equal to the value of the contract). Similarly, the τ threshold ranges from 0 to 1.5 with two special values: 0 (meaning the buyer is greedy and will commit to any intermediate deal that it can get hold of) and 0.5 (meaning the buyer is patient and will only engage in a deal that provides high expected utility value but it is not too strict on accepting offers from seller). Setting τ to be greater than 1.5 makes no different to the outcome of the mode (the buyer will never accept any offer from a seller). To sum up, the independent variables are given in table 4.4 and the dependent ones are listed in table 4.5, respectively.

Variables	*Descriptions*	*values*
ρ_0	the initial penalty fee	[5,100]
ρ_{max}	the final penalty fee ($\rho_{max} \geq \rho_0$)	[5,100]
τ	the μ threshold	[0,1.5]
ω	the number of concurrent commitments	[1,4]

TABLE 4.4: The independent variables.

The seller agents in this evaluation are characterized in a similar fashion to ones set up in the previous experiments (see section 3.5.2). The only difference is that now if a seller has committed to a deal, it has a chance of being made an outside offer with the utility value of 1.0 (which is the highest possible utility value). Thus, there is a probability that it will decommit. To this end, we consider three types of sellers:

- *loyal*: once a seller has committed to an intermediate deal, it will not renege from it (this is equivalent to the sellers in the experiments in chapter 3).
- *loose*: a seller always breaks a committed deal if it is presented with a better option.
- *partial*: if a seller finds a better option, it will break a committed deal with a percentage of probability. In this experiment, we set this percentage to be 50%, meaning that half of the time a seller finds a better deal, it will renege and half of the time it will stay with its current deal.

Variables	*Descriptions*
U	the utility value of the final agreement
N	the number of successful negotiations
D	the number of decommitments made by buyer

TABLE 4.5: The dependent variables.

4.3.2 Hypotheses

Here, we want to evaluate the performance of our commitment model against different seller's types, in terms of the number of successful negotiations and the average utility value of the final agreement obtained. Next, we aim to find the most influential variables and to find out how to set their values in order to gain best overall performance in each environment. We now turn to the specific hypotheses.

HYPOTHESIS 13. When dealing with loose or partial sellers, the higher the penalty fee is, the lower the number of final agreements reached by the buyer.

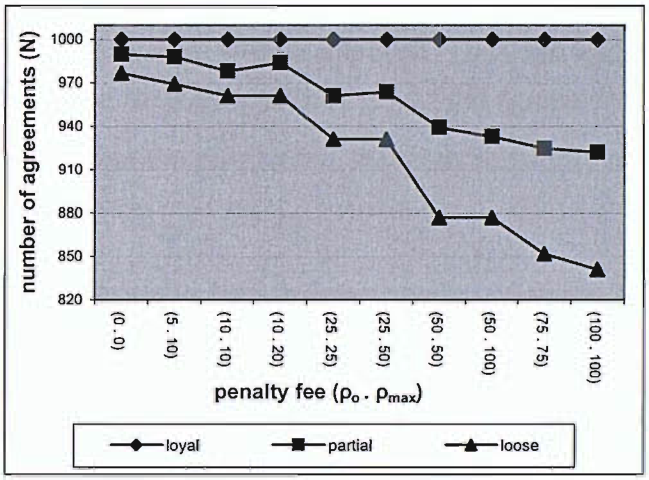


FIGURE 4.2: Number of successful negotiations for varying penalty fee.

To evaluate this hypothesis, we measure the number of final agreements achieved with varying types of seller agents (see figure 4.2). As can be seen, the number of final agreements reached by the buyer is dramatically reduced as the penalty fee is increased. Specifically, when dealing with loose sellers, around 97% of the negotiations are successful when the penalty fee is 5%. As the penalty fee increases to 100%, this success rate drops down to only 84%. Similarly, the figures when dealing with partial sellers are 98% and 92%, respectively. This decreasing trend is explained by the deliberation mechanism of the buyer. Specifically, assume that the buyer has already made a commitment with seller k and now it is presented with another offer from seller k'. If it decides to take this new offer from k', it will have to pay k a decommitment fee ρ. As the penalty fee is increased, so is ρ. Thus, in some cases, the buyer cannot afford to take this new offer and it has to stay with its commitment to k. Later on, if k decides to break its commitment, the buyer is left with no intermediate agreement. As such, there may not be enough time for the buyer to find another replacement deal and, thus, no final agreement can be reached. On the other hand, if the buyer can take the offer from k', the probability that k' will renege is less than that of k. Thus, a final agreement can be reached.

Another observation is that the more loyal the seller is, the greater the number of final agreements that the buyer makes. This difference is caused by the probability of the sellers breaking their commitments. Since a loyal seller never reneges, once it has committed, its contract is kept until either it is declined by the buyer or it is selected as the final agreement. Therefore, once an intermediate deal is reached, a final agreement is always guaranteed to exist. However, this is not the case for the other types of sellers. Once they have committed, it is not guaranteed that they will actually stay faithful with their commitments. If a seller breaks a contract, the buyer has to find a replacement. If it fails to do so, no final agreement will be achieved. Thus, the less loyal the sellers are, the fewer chances there are for the buyer to reach a final agreement.

HYPOTHESIS 14. The higher the penalty fee, the lower the utility of the final agreement gained by the buyer.

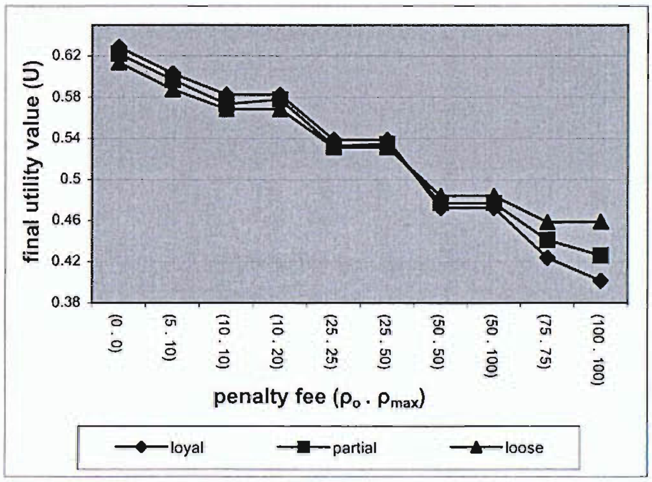


FIGURE 4.3: Final utility value for varying penalty fee.

As can be seen from figure 4.3, this trend is true for all seller types. Specifically, when dealing with loose sellers, the average utility of the final agreement for the buyer drops from 0.61 to 0.46 when the penalty fee goes from 5% to 100%. The corresponding figures for partial and loyal sellers are 0.62 to 0.43 and 0.63 to 0.40, respectively. The reason for this decrease in the final utility value is that the higher penalty fees mean more chance that the buyer will commit to an early agreement (and stay with this commitment until either its deadline is reached or the corresponding seller decides to renege). These early commitments by the buyer have two main effects. First, such agreements tend to have lower utility value for the buyer, compared to the contracts that are offered at a later stage (the buyer cannot afford to take these contracts due to high decommitment fees). Second, once that commitment is later broken, the buyer will have to find a replacement. Even if it is successful in finding one, since there is not much time for bargaining, the utility value of this newly found agreement is likely to be less than that of the previous deal. Consequently, the utility gained by the buyer is reduced.

Furthermore, with increasing penalty fee, the more loyal the seller, the lower the value of the final agreement gained by the buyer (see figure 4.3). The reason for this observation is because the buyer benefits from the decommitment fee gained when a seller reneges from a committed deal. As per our experimental setup, loose sellers decommit more often than partial sellers and loyal sellers never renege. Thus, as the penalty fee increases, the buyer will benefit more when dealing with less loyal sellers.

HYPOTHESIS 15. The buyer decommits less frequently as the penalty fee increases.

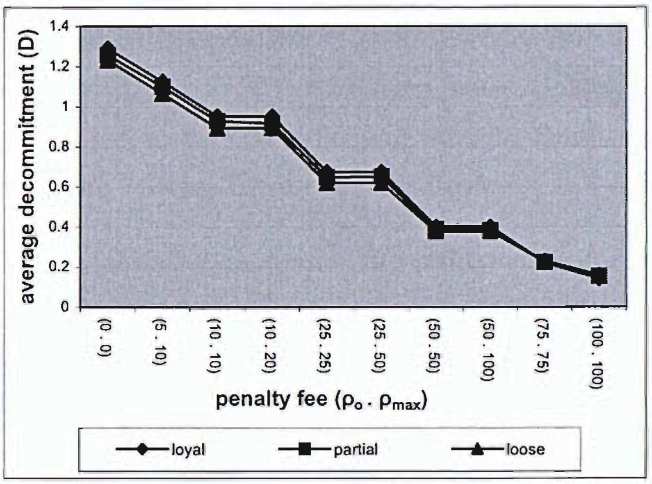


FIGURE 4.4: Number of buyer's decommitments for varying penalty fee.

Figure 4.4 shows the average number of decommitments made by the buyer for varying penalty fees and different seller types. Since the buyer's deliberation includes the decommitment fee it has to pay if it want to replace its current intermediate deal (see equation 4.1), the less it has to pay, the more favorable it will be to take up a better deal. Thus, even when a seller offers an intrinsically higher value contract than the current deal it has, the buyer may be better off sticking with its existing commitment in order to avoid paying a hefty fine. This is why the buyer almost never reneges when the penalty fee is close to 100%.

HYPOTHESIS 16. The more patient the buyer, the higher the utility for the final agreement. However, the chance of having a final agreement is reduced.

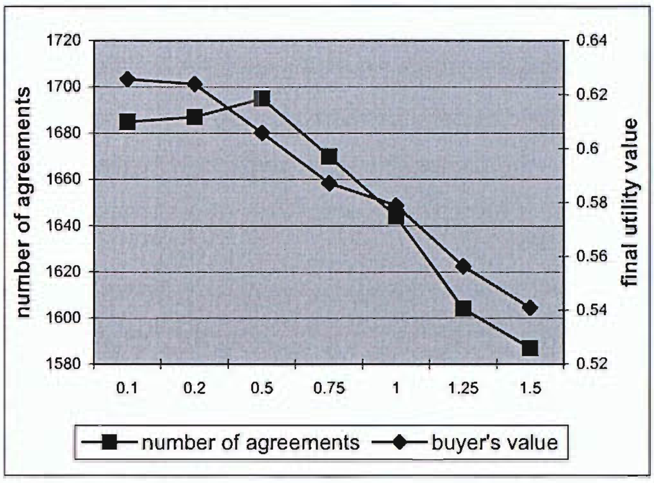


FIGURE 4.5: Performance vs degree of acceptance.

We start by looking at the performance of the model with a number of different values for the degree of acceptance (specifically $\tau \in [0.1, 1.5]$). For simplicity, we fix the penalty fee value at $(\rho_0 = 5, \rho_{max} = 10)$ and assume we are dealing with *partial* sellers. These values are chosen just to give us an idea of how the results could potentially be and a detailed analysis will follow. The results are displayed in figure 4.5.

As can be seen, as the value of τ increases, the utility value for the final agreement decreases. This is because in a particular negotiation, if the buyer tends to ignore the current offer from the seller, in favor of a higher value one at a later time, there is a possibility that a high value offer will not be forthcoming (e.g. the seller may run out of time or be at the limit of its reservation values). Thus, towards the end of the encounter it will have to settle for a lower value deal (because this is better than no deal). This, in turn, puts a downward trend on the final utility value achieved.

On the other hand, the number of final agreements reached increases as the value of τ increases up to 0.5, then it decreases. Now, since we are dealing with partial sellers, if they are presented with a better outside offer, they have the chance to renege and may leave the buyer with no agreement at hand. When the value for τ is small (less than 0.5 in this case), the buyer tends to take up any offers that are available to it at an early time. Later, when the seller that is sharing the commitment with the buyer decides to back down, there might not be enough time for the buyer to recover from this loss and thus it might end up with no agreement at the end of the encounter. However, if it is too strict on accepting intermediate deals, it also risks the chance of having obtained no deal at all. This is the situation when the value of τ increases past 0.5. From figure 4.5, it can be seen that by setting value of τ at around 0.5, the buyer will achieve the highest number of final agreements with a reasonably good final utility value.

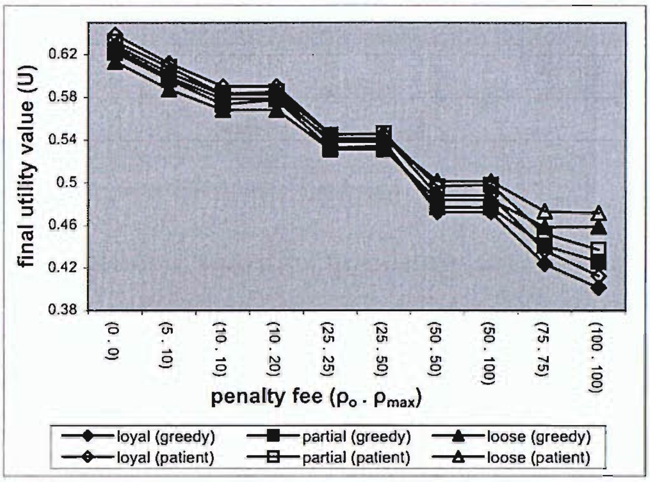


FIGURE 4.6: Final utility value for varying penalty fee.

We extend the aforementioned result by comparing the results of having two different values for τ: *greedy* ($\tau = 0$) and *patient* ($\tau = 0.5$) in the experiments with different penalty values, as well as different seller types. Recall, the greedy buyer will commit to any offer that it can take (if it is more beneficial than the one it currently has, taking into account the decommitment fee it will have to pay). In contrast, the patient buyer will only commit to an offer that has significantly greater value (compared with the one that it currently has). As it only accepts higher value contracts compared to its counterpart, the patient agents' final agreements always have higher utility value than those of the greedy agent (see figure 4.6).

However, even though it can gain better utility value than its greedy counterpart, the patient agent manages to get fewer agreements than its counterpart (see figure 4.7). This is because the patient agent only accepts a deal if the degree of acceptance (μ) of this deal is greater than a threshold (in this case, $\tau = 0.5$). Thus, not all the deals proposed by the sellers satisfy this condition. Indeed, in some cases, none of the proposed contracts satisfy this condition. This limits the chance of the buyer having an agreement at the end of the negotiation. On the other hand, the greedier the agent is, the higher the chance that an offer will be accepted. Consequently, the greedy agent will be able to reach more agreements than the patient one at the end of the bargaining process.

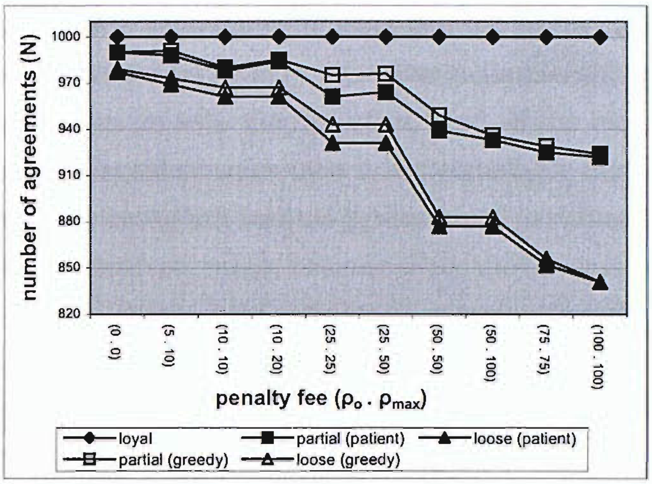


FIGURE 4.7: Number of successful negotiations for varying penalty fee.

HYPOTHESIS 17. When dealing with loyal sellers, the buyer is better off committing to a maximum of one contract at any one time. For other seller types, the buyer should commit to a maximum of two contracts.

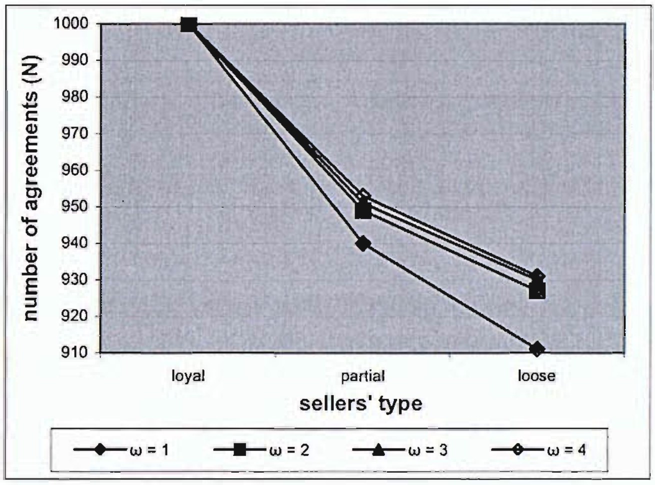


FIGURE 4.8: Number of agreements vs buyer's maximum commitments.

Figure 4.8 shows the number of agreements obtained by the buyer at the end of the encounter when it varies the number of commitments it can hold at any one time (here $\omega \in [1, 4]$). As can be seen, when holding more than one commitment, the buyer increases its chance of reaching an agreement when dealing with non-loyal sellers. In particular, when ω is increased from 1 to 2, the buyer gains 0.9% more final agreements when dealing with partial sellers and 2% more when dealing with loose sellers. This improvement can be explained simply by looking at the behaviors of the sellers. As the sellers are not loyal, when presented with an outside offer, they may renege. If this happens near the end of the negotiation process and the buyer can only commit to a single contract, it will leave the buyer very little time to find an alternative (and in some cases it will not be able to do so). On the other hand, if the buyer is holding more than one contract and an agent reneges then it has something that it can fall back on and it is less vulnerable to being left with no agreement. For values of $\omega > 2$, however, the improvement is comparatively minor because when the buyer is committing to more than one contract, the chance that all the sellers renege is significantly reduced compared to the situation when the buyer can only have one commitment at a time. This, in turn, has a very slight impact on the number of agreements achieved.

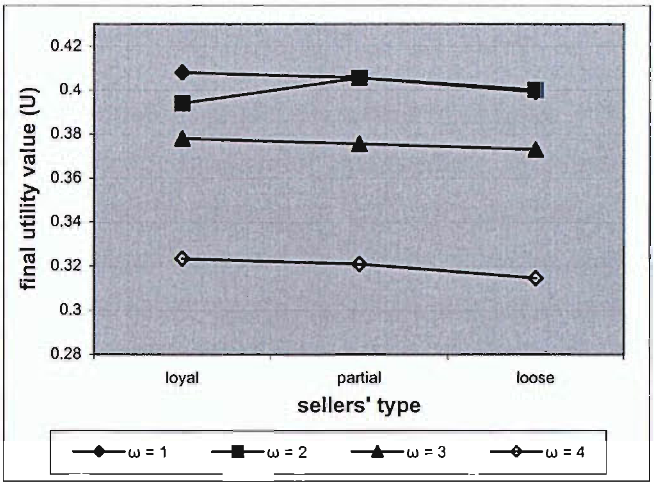


FIGURE 4.9: Final utility value vs buyer's maximum commitments.

The final utility achieved by the buyer with varying values for ω is displayed in figure 4.9. As can be seen, when $\omega > 2$, the utility gained is dramatically reduced (8% decrease when ω goes from 2 to 3 and 16% decrease when ω goes from 3 to 4). This is because if the buyer agent has more contracts at the end of the negotiation process, it will end up paying a significant penalty fee for breaking them. When $\omega = 2$, the situation is similar to that of dealing with loyal sellers. However, when negotiating with partial or loose sellers, $\omega = 2$ gives similar results and, in some cases, is better than setting ω to 1. The reason for this is because as the non-loyal seller agents can renege on their commitments and if they do so towards the end of the negotiation process, the buyer will gain additional penalty fees from those sellers and is still left with at least one intermediate contract in hand. Thus, at the end, it is still able to have the final agreement, but it does not have to pay a decommitment fee to any other seller agent.

As can be seen, when dealing with loyal sellers, the buyer does not necessarily need to have more than one commitment since it can be sure that the sellers will never renege from their deals. However, when dealing with partial or loose sellers, the situation is different. The greater the number of commitments it holds, the higher the number of final agreements it is likely to obtain. Nevertheless, the final utility value reached is decreased because it will have to pay a large amount of penalty fees to decommit from these commitments. To this end, the buyer is best setting ω to 1 when dealing with loyal sellers and ω to 2 when dealing with other types of seller to ensure that it will achieve the highest possible utility value together with an acceptable number of final agreements.

4.4 Summary

The basic concurrent negotiation model presented in chapter 3 assumes that the buyer has the right to commit to any number of intermediate agreements and then, without penalty, be able to select one of these to be the final agreement. This assumption was required to minimize the buyer's uncertainty initially; however, it gives the buyer agent a privileged position over the seller agents and thus limits the desirability of participating in the bargaining encounter from their perspective. To overcome this, the commitment protocol discussed in this chapter provides a method to relax this constraint and to treat all the agents equally. Thus in the revived protocol, if any agent decides to break off its commitment, it will have to pay a penalty fee to the other agent, regardless of its role.

In order to adapt to this change, the internal reasoning of the buyer agent has been modified. Specifically, when faced with an intermediate acceptable offer, instead of taking it promptly (as per chapter 3), it has to reason whether or not it should take it. To do this, the agent embodies a number of different commitment strategies, ranging from *greedy* (desperately seeking deals) to *patient* (waiting for a good opportunity). It can also choose to hold different numbers of concurrent commitments which gives it flexibility in different negotiation contexts. In particular, when dealing with loyal sellers, it does not necessarily need to have more than one concurrent commitment but it should hold two concurrent commitments when dealing with other types of seller in order to gain best possible outcome.

Using empirical evaluation, we have shown that our concurrent model performs well in a variety of contexts. Naturally, since this commitment protocol puts the buyer in a less favored position, the performance of our model drops compared with the basic concurrent model. However, it is now much more realistic and practically plausible. From the experiments conducted, the most notable observation is that the higher the penalty fee, the lower the utility of the final agreement gained by the buyer. Another observation we obtained is that the more patient the buyer, the higher the utility for the final agreement. However, the chance of having a final agreement is reduced.

Having made the model more realistic, the next step is to see if we can make the overall negotiation process more efficient (in terms of exploiting potential information about the sellers, if applicable). To this end, we aim to use a variety of learning techniques to improve the performance of the model by introducing a learning based negotiation strategy. This work is detailed in the next chapter.

Chapter 5

Adaptive Negotiation

The buyer agent in our basic concurrent model described in chapter 3 operates in an incomplete information environment. This means it can negotiate without knowledge of its opponents' utility functions, evaluation criteria and reservation values. The same assumption remains in our model for handling commitment discussed in chapter 4. Such an assumption is made because it means the model can be applied in almost all negotiation settings. Nevertheless, there are situations in which partial information about the seller agents is sometimes available [Fatima *et al.*, 2001; Zeng and Sycara, 1998]. In such cases, this information can be exploited to give the buyer an advantageous stance in its bargaining. To this end, this chapter investigates the application of machine learning techniques to exploit such information in our negotiation model.

In more detail, this chapter is structured in the following way. First, a new learning-based negotiation strategy for the buyer agent is introduced in section 5.1. Next, section 5.2 gives an example of how this strategy can be appliedin one particular negotiation context. As before, we evaluate it empirically and the results are detailed in section 5.3. Finally, section 5.4 concludes.

5.1 A Learning-Based Negotiation Strategy

Even though agents rarely have complete information about the opponents that they are dealing with in realistic situations, it is not uncommon to have some information about them [Fatima *et al.*, 2001; Zeng and Sycara, 1998]. Given this, in this chapter, we consider the situation where the buyer agent does indeed have some information about the sellers that it will negotiate with. In particular, we assume it has information about how the sellers generate their offers, which it then uses to help it make a better value counter-offers. We focus on this case because such information may be available from different sources (e.g. in the travel agent scenarios detailed in section 1.2, from its previous holiday arrangements, *sigma* knows that most of the travel agents start with very high price offers and then quickly lower down their prices as time goes by). However, before we detail the situation, we define the basic terms that are used to describe the learning model in table 5.1. This table gives a brief description of their meanings and the detailed explanations follow in the subsequent sections.

Symbols	*Description*
$\mathbf{F}_s$	the set of available negotiation strategy functions for the seller
f, f_s	a specific negotiation strategy function
P_f	the set of parameters that goes with a specific negotiation strategy function f
p_i	a specific parameter that belongs to P_f
Q_{p_i}	the quantitative range of p_i
$E(f)$	the different between the utility values generated by the buyer and the actual utility value from the seller f
$\mathbf{H_k}$	the set of counter-offers made by seller k to buyer b
$\mathbf{H_{b_k}}$	the set of offers made by buyer b to seller k
RP_k	the buyer's predicted lower limit of k's reservation value
Δ_t	the number of negotiation rounds after which b will start applying the learning strategy
Δ_k	buyer b's assumption about the difference between its deadline and seller k's deadline
Δ_E	the error threshold that determines if b's prediction should be used to generate a new offer
Δ_U	the utility value threshold that b uses to generate a new offer

TABLE 5.1: The list of learning symbols (a **bold** symbol represents a set).

In particular, in this chapter we are interested in the functions that the sellers use in order to generate counter-offers to respond to the buyer's offers. To this end, we assume that the buyer agent knows the set $\mathbf{F}_s$ of strategy functions that the sellers will use, but it does not know which specific function (i.e. which $f_s \in \mathbf{F}_s$) is used by a particular seller. We believe this is a reasonable assumption since most of the dominant negotiation strategies can be found in the literature and, thus, are available to the buyer. Specifically, at time t, assume seller k generates counter-offer $\phi^t_{k \to b}$ using the strategy function $f_s \in \mathbf{F}_s$ (which takes the tuple $(H_k, H_{b_k}, t, t_{k_{max}})$ as the input parameters). The utility value of $\phi^t_{k\to b}$, U^k, is calculated as follows:

$$U^k(\phi^t_{k\to b}) = f_s(H_k, H_{b_k}, t, t_{k_{max}})$$

where H_k is the set of previous counter-offers made by k to date: $H_k = \{\phi^{t'}_{k\to b} \mid t' < t\}$ and H_{b_k} is the set of offers proposed by the buyer to date $H_{b_k} = \{\phi^{t'}_{b\to k} \mid t' < t\}$ and $t_{k_{max}}$ is the negotiation deadline of k.

By means of illustration, consider a negotiation strategy function $f_s \in \mathbf{F}_s$ which implements the *time dependent* strategy described in section 3.4.1.3:

$$f_{time\_dependent}(H_k, H_{b_k}, t, t_{k_{max}}) = \delta^k + (1 - \delta^k) * (\frac{t}{t_{k_{max}}})^{\frac{1.00}{\beta}}$$

where δ^k is the initial proposal utility value (depending on the chosen strategy) and β is the parameter that decides the shape of the function. As can be seen, this particular f_s does not take into account the previous offers. Hence, H_k and H_{b_k} are not used.

Another example is the Relative Tit-for-Tat function discussed in section 2.2.1:

$$f_{tit\_for\_tat}(H_k, H_{b_k}, t, t_{k_{max}}) = min(max(\frac{\phi^{t-3}_{b\to k}}{\phi^{t-1}_{b\to k}}\phi^{t-2}_{k\to b}, \delta^k, 1))$$

where $\phi^{t-3}_{b\to k}, \phi^{t-1}_{b\to k} \in H_{b_k}$ are the previous offers from the buyer and $\phi^{t-2}_{k\to b} \in H_k$ is the previous offer made by k.

Now that the buyer knows that such functions exist, however, it does not know the accompanying parameters (such as β, δ^k and H_k). Given this information, the buyer's task is to try and determine the function that a seller is using (with the accompanying parameters) in order to try and predict the reservation values of the seller. If this can be achieved, an offer that is close to that value will be generated and proposed to that particular seller. If accepted, it will give the buyer a much higher value return than might otherwise be the case.

In more detail, figure 5.1 illustrates the underlying operation of this approach in a particular negotiation. In this figure, the thin arrows represent the buyer's offers (the longer arrow is the predicted value offer) and the thick arrows represent the seller's counter-offers. Typically, the agents will start from one end of their reservation values and go towards the other end. If an offer falls in the zone of agreement, it is likely to be accepted (there are some situations in which our buyer agents will not accept, see sections 3.4.1.2 and 4.1.2). Then, as can be seen, if the buyer can reasonably identify the seller's reservation value, it can generate an offer that is likely to be accepted by that seller and that has a higher utility value than that of either the seller's counter-offers or its typically generated offers.

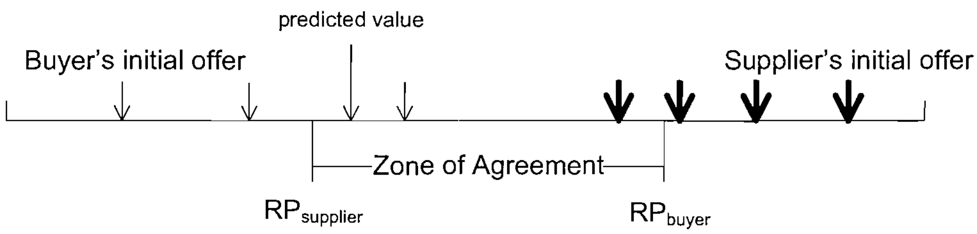


FIGURE 5.1: Buyer's prediction of the seller's reservation value.

Specifically, in order to predict seller k's reservation value, the buyer operates under the assumption that k's strategy function f_s will give the value that initially starts from a high value for k (low for b) then go down to a lower value for k (higher for b) near k's deadline. However, since the buyer does not know either the f_s that k uses, nor its deadline $t_{k_{max}}$, it attempts to find a function $f \in \mathbf{F}_s$ that gives the values that are closest to the value of the counter-offers that k has proposed to b so far.

Thus, for each function $f \in \mathbf{F}_s$, let P_f be the set of accompanying parameters that f uses (i.e. $P_{f_{time\_dependent}} = \{\delta^k, \beta\}$). Although b does not know the exact value that k uses for each parameter $p \in P_f$, it is assumed to know the quantitative range of each parameter (i.e. $\delta^k \in [0 \ldots 0.1], \beta \in [0 \ldots 2]$)[1]. Obviously, b knows the elements of the set H_k and H_{b_k}.

Now at time $t > \Delta_t$[2], a number of offers/counter-offers have been proposed by both agents. Given this, our agent will then perform a Brute Force search in the set $\mathbf{F}_s$ to find the function that best explains the seller's negotiation offers. Specifically, for each $f \in \mathbf{F}_s$, b will attempt to find a value for each parameter $p \in P_f$ so that the value generated by f will have the least error value (E) calculated by the standard error function given in equation 5.1:

$$E(f) = \sqrt{\sum_{t'=1}^{t'<t} (f(t') - U(\phi^{t'}_{k \to b}))^2} \tag{5.1}$$

[1]Again, we believe this is a reasonable assumption because for each parameter, normally, its value will be within a defined range to be effective

[2]In earlier rounds, the buyer uses a standard negotiation strategy as per section 3.4.1.3.

This equation is selected based on the standard error function that is normally used in a typical neural network [Dayhoff. 1990; Specht, 1991], which allows the buyer to calculate the average differences between the values that the targeting f function generates and the actual values of the counter-offers that the seller k has proposed.

```
function findFunction() return f, P_f
/* return a function f and its set of parameters P_f */
input
    n_F: the number of functions in F;
    F = {f_1 ... f_{n_F}}: the set of available strategy functions;
    P_{f_i} = {p_{i_1} ... p_{i_{n_f}}}: the set of parameters for function f_i ∈ F_s;
output
    f ∈ F: the selected function;
    P_f: the exact value for each member of the parameters of f;
begin
    minError := MaxInt;
    f := ∅;
    P_f := ∅;
    for i = 1 to n_F do begin
    /* go through each function f in F */
        P_{f_i} := findParameters(f_i);
        /* find the set of parameters P_f that gives the smallest error value */
        C_i := checkError(f_i, P_{f_i});
        if (C_i < minError) then begin
        /* if this error value is smaller than what we currently have then this
        combination of f and P_f is saved */
            minError := C_i;
            f := f_i;
            P_f := P_{f_i};
        end
    end
    return f, P_f;
    /* at the end, return the saved combination */
end
```

FIGURE 5.2: Buyer's algorithm to find the closest matching strategy function.

Given such information, the buyer attempts to find the specific function that seller k is using. To this end, figure 5.2 details the algorithm that our buyer uses (the details of the two functions that this algorithm uses, namely `findParameters` and `checkError` are expressed in figures 5.3 and 5.4, respectively). Basically, for each function $f \in \mathbf{F}_s$, it tries to find the set of parameters P_f that give the lowest error rate, which, in our cases, means that the differences between the calculated values and the actual values of the counter-offers received are minimized for that particular function. After that, it assumes that the function that has the lowest error rate is the matching function that k is using.

Specifically, in order to find the set of values for the parameters of a given function f, the buyer uses a *divide-and-conquer* technique (the algorithm is detailed in figure 5.3). Now, since the quantitative range for each parameter is continuous, it is impossible for the buyer to try every possible value (given it is possible to have more than one parameter for each function, the computational complexity would increase exponentially). Thus, for reasons of simplicity, we adopt a simpler approach. Basically, for each parameter $p_i \in P_f$, the range $Q_{p_i} = [min_{p_i}, max_{p_i}]$ is equally divided into s points that will cover the whole range of Q_{p_i}. Then, for each parameter, the buyer will set the value at each point to the temporary set of values for P_f and continue for the next parameters. If the combination of values gives the lowest error rate so far, it is selected as the final set of values for P_f. This process continues recursively until all combinations have been tested.

Next, figure 5.4 details the algorithm based on equation 5.1 to determine the differences between the values generated by a given strategy function f and a particular set of parameters P_f compared with the values of the actual counter-offers proposed by the seller. The algorithm starts from the time when the first counter-offer has been received and it continues up until the previous round. At each time period, it measures the differences between the calculated and the actual value and, finally, calculates the overall error value.

After using algorithm 5.2, the buyer will have the function that most closely matches what seller k has been offering so far. Now, this matching is only approximated since the buyer does not actually know the utility function of k and it therefore has to use its own function. However, it does give the buyer some indication of what the next move of k might be. From this, the next step is to predict what is the lower limit of seller k's reservation value (RP_k). To this end, we assume a seller will typically offer a value that is closer to its reservation value at the time that is close to its negotiation deadline. Thus, if we know the deadline of k, we can obtain some idea of what the reservation value of k will be. However, since this information is unavailable to the buyer, it cannot make such a prediction. Nevertheless, the buyer can attempt to estimate the negotiation deadline of k (T_k) by considering the following options:

- O1: The seller's estimated deadline is the same as the buyer's: $T_k = t_{b_{max}}$.
- O2: The seller's estimated deadline is greater than that of the buyer's: $T_k = t_{b_{max}} + \Delta_k$.

```
function findParameter(f) return P_f
/* return the best set of parameters P_f for this function */
input
    f: a strategy function ∈ F_s;
    n_f: the number of parameters of f;
    P_f = {p_1 ... p_{n_f}}: the set of parameters of f;
    Q_{p_i} = [min_{p_i}, max_{p_i}]: the quantitative domain of parameter p_i;
output
    P_f: the exact value for each member of the parameters of f;
procedure testSingleParameter(pIndex)
/* check all the possible values for each single parameter pIndex */
begin
    for i := min_{p_pIndex} to max_{p_pIndex} do begin
    /* go through each possible value of parameter pIndex */
        pIndexValue := i;
        tempSet := tempSet + pIndexValue;
        /* add this value to a temporary set of parameter tempSet */
        if (pIndex < n_f) then begin
        /* if pIndex is not the last parameter then check the next one recursively */
            testSingleParameter(pIndex + 1);
        else begin
        /* if pIndex is the last parameter then measure the error value */
            errorValue := checkError(f, tempSet);
            if (errorValue < minValue) then begin
            /* if this error value is smaller than what we currently have then
            the value of tempSet is saved */
                minValue := errorValue;
                minPf := tempSet;
            end
        end
        tempSet = tempSet − pIndexValue;
        /* remove this value from tempSet */
    end
end
begin
    minValue := MaxInt;
    minPf := ∅;
    tempSet := ∅;
    testSingleParameter(1);
    /* start checking the first parameter */
    return minPf;
end
```

FIGURE 5.3: Buyer's algorithm to find the best value set of parameters for a given function.

---

function checkError(f, P_f) **return** E
input
  f: *a strategy function* $\in \mathbf{F}_s$;
  $P_f = \{p_1 \ldots p_{n_f}\}$: *the set of parameters of f*;
output
  E: *the measured error*;
begin
  $E := 0$;
  for $i := 1$ **to** $t - 1$ **do begin**
    $E := E + \mathtt{sqr}(f(i) - U(\phi^i_{k \to b}))$;
    /* *measure the difference between the generated value and the actual one* */
  end
  $E := \mathtt{sqrt}(E)$;
  return E;
end

FIGURE 5.4: Buyer's function to measure the differences between a given function and the value of the seller's counter proposal.

- O3: The seller's estimated deadline is less than that of the buyer's: $T_k = t_{b_{max}} - \Delta_k$.

Initially, the buyer assumes that all three options have the same probabilities (i.e. $P(O1) = P(O2) = P(O3) = \frac{1}{3}$). At each negotiation round, the buyer picks the option that has the highest probability and applies it to the selected function in order to calculate the seller's reservation value. Then it selects an offer that is close to that value and proposes it to the seller. The aforementioned probability distribution will be updated accordingly (if the offer is accepted, the probability of that particular option is increased and if the offer is rejected then the probability of that particular option is decreased). Formally, at time $t > \Delta_t$, the steps that the buyer agent takes are as follows:

1. find the function $f \in \mathbf{F}_s$ and its set of parameters P_f so that the error value $E(f)$ is smallest.

2. if $E(f) > \Delta_E$ (the error threshold), process with its initial negotiation strategy (as selected by the coordinator) and continue to the next round.

3. select $O_i, i \in [1, 3]$ so that $P(O_i)$ is highest. If there is more than one option, choose randomly between them.

4. estimate the seller's deadline T_k and use it to predict the lower limit of the seller k's reservation value (RP_k) using the reverse utility function of the buyer (since

the actual utility function of k is not available to b so it has to be approximated). Specifically, for each negotiation period $t' < t$, calculate seller k's reservation value up to that time $RP_k(t')$. Finally, these values are averaged to get the final value for RP_k.

5. propose an offer with utility value of $max(RP_b, RP_k - \Delta_U)$ to k.

6. in the next round, update the probability distribution $P(O)$ according to the response of k.

To illustrate the operation of our approach, the next subsection considers a specific instance of such an adaptive strategy.

5.2 An Example Adaptive Negotiation Strategy

Assume that the buyer negotiates with one seller, k, about a single issue (a contract ϕ is composed of a single value x). The negotiation deadline for b is 10 and for k it is 15. The reservation values for b are $[x^b_{1_{min}} = 10, x^b_{1_{max}} = 90]$ and for k are $[x^k_{1_{min}} = 50, x^k_{1_{max}} = 160]$. The strategy that k uses is $f_k(t) = (\frac{t}{t_{k_{max}}})^{\frac{1.00}{0.4}}$, here the parameters P_f are $\{\delta^k = 0, \beta = 0.4\}$ and the buyer uses the strategy $f_b(t) = (\frac{t}{t_{b_{max}}})^{\frac{1.00}{0.15}}$, here the parameters P_f are $\{\delta^b = 0, \beta = 0.15\}$. Let b and k use a linear utility value function (see section 3.4.1.2):

$$U^b(\phi) = \frac{x^b_{1_{max}} - x}{x^b_{1_{max}} - x^b_{1_{min}}}$$
$$U^k(\phi) = \frac{x - x^k_{1_{min}}}{x^k_{1_{max}} - x^k_{1_{min}}}$$

Assume that b knows that the set of strategies function $\mathbf{F}_s$ is composed of 2 families of functions (taken from Faratin- section 2.2.1):

$$\mathbf{F}_s = \begin{cases} f_1(t) = \delta^k + (1 - \delta^k) * (\frac{t}{t_{k_{max}}})^{\frac{1.00}{\beta}} \\ f_2(t) = e^{(1 - \frac{min(t, t_{k_{max}})}{t_{k_{max}}})^{\beta} ln(\delta^k)} \end{cases}$$

where the range for δ^k is $[0 \dots 0.1]$ and β is $[0 \dots 2]$. The other parameters that b has are: $\Delta_k = 3, \Delta_E = 0.1, \Delta_U = 0, P(O1) = 0.5, P(O2) = 0.3, P(O3) = 0.2$. Thus, it is assumed that k's estimated deadline is 10 since $P(O1)$ has the highest probability.

Given all this, let table 5.2 show the history of the negotiation process after 4 rounds. In this table, $U^b(\phi_k)$ denotes the utility value of the seller's counter-offer calculated using the utility function of b (it is negative since the value of the offer from k is beyond the acceptable range for b). In order to use the closest matching strategy algorithm (of figure 5.2), these values must be converted to be in the range of [0,1]. This is done via a simple conversion: $U'^b(\phi_k(t)) = \frac{U^b(\phi_k(t))}{U^b(\phi_k(1))}$.

	1	2	3	4
k	160	159.87	159.29	158.03
b	10	10.01	10.18	10.79
$U^b(\phi_k)$	-0.87	-0.87	-0.86	-0.85
$U'^b(\phi_k)$	1	1	0.99	0.97

TABLE 5.2: Negotiation history sample.

Now, using the closest matching strategy algorithm (see figure 5.2), f_1 together with $P_f = \{\delta^k = 0.0, \beta = 0.34\}$ gives the smallest error $E(f) = 0.0018$. Since $E(f) < \Delta_E$, b will try to predict the value RP_k in order to generate an offer to propose. The utility function U'^b that b will use is based on U^b (section 3.4.1.2):

$$U'^b(x) = \frac{x - RP_k}{\phi_{k\to b}(1) - RP_k}$$

thus,

$$RP_k = \frac{x - \phi_{k\to b}(1) \times U'^b(x)}{1 - U'^b(x)} \tag{5.2}$$

	1	2	3	4
U^k	1	1	0.99	0.97
$\phi(k)$	160	159.87	159.29	158.03

TABLE 5.3: Predicted utility value for seller k.

Using f_1, together with $P_f = \{\delta^k = 0.0, \beta = 0.34\}$, the predicted utility values for seller k are given in table 5.3. Now using equation 5.2 and the input values in table 5.3, we have the following:

$$RP_k(2) = 49.76$$
$$RP_k(3) = 78.80$$
$$RP_k(4) = 92.1$$

The average of these numbers, 73.56, is the predicted value of RP_k (as detailed in section 5.1). This value is within the quantitative range of b and has the utility value of $U = 0.206$ according to the buyer's utility value function. Thus the buyer will propose an offer with value $U + \Delta_U$ or 0.206 to k. Since it is within k's range, it will be accepted. On the other hand, if b only uses the standard negotiation strategy (allocated by the coordinator), the offers it will propose to k are displayed in table 5.4. From this, only the offers at 10 will be accepted by k since their values are within k's acceptable domain, which is [50, 160]. Nevertheless, they have lower utility value for b compared with what it would have achieved had it followed the closest matching strategy algorithm (0.01 compared to 0.206). Furthermore, b is able to find an agreement at time 5, which is at least 5 rounds earlier than if it negotiates in a standard way. This early achievement can then be used as a basis for other negotiations in our concurrent negotiation setting in order to achieve better outcomes for b.

	1	2	3	4	5	6	7	8	9	10
$U^b(\phi_{b\rightarrow k})$	1.00	1.00	1.00	1.00	0.99	0.97	0.91	0.77	0.50	0.01
$\phi_{b\rightarrow k}$	10.00	10.01	10.03	10.18	10.79	12.66	17.42	28.07	49.63	89.00

TABLE 5.4: b's generated offers to propose to k.

As can be seen from this example, if the buyer agent can accurately predict the reservation value of a seller (based on its previous counter-offers), it is sometimes able to secure a good outcome compared to negotiating in the way outlined in chapter 3. However, this need not always be the case since the buyer uses a number of assumptions in order to make the prediction. Thus, we would like to test the algorithm in various negotiation settings in order to investigate its performance. The details of this evaluation are given in the subsequent section.

5.3 Evaluation

Here we evaluate our adaptive negotiation approach, focusing particularly on its performance in terms of the utility value of the final agreement and the number of agreements achieved (as per chapter 3). Once again, *empirical evaluation* is used as the method of measurement here to evaluate the requirement *partial information* stated in section 1.2. Again, general *hypotheses* are formed to express the intuitions about the causal factors within the model and the *experiments* are then conducted to either support these hypotheses or go against them. The structure of this section consists of two parts: the experimental setup is described in section 5.3.1 and the *hypotheses* are presented and evaluated in section 5.3.2.

5.3.1 Experimental Setup

For our experimental setup, we reuse the basic experimental environment described in section 3.5.2 and only add/update the commitment related settings. For simplicity, we assume that $\mathbf{F_s}$ is composed of only different time dependent strategies. Thus:

$$\mathbf{F}_s = \begin{cases} f_1(t) = \delta^k + (1 - \delta^k) * (\frac{t}{t_{k_{max}}})^{\frac{1.00}{\beta}} \\ f_2(t) = e^{(1-\frac{min(t, t_{k_{max}})}{t_{k_{max}}})^{\beta} ln(\delta^k)} \end{cases}$$

The parameter set P_f is composed of two parameters, δ^k and β, with the quantitative domains of $[0 \ldots 0.1]$ and $[0 \ldots 2]$, respectively. To sum up, the independent variables are given in table 5.5 and the dependent ones are listed in table 5.6.

Variables	*Descriptions*	*values*
t_{max}	buyer's deadline	[20 ... 30]
$\mathbf{F}_s$	strategy functions	f_1, f_2
δ^k	member of P_f	[0 ... 0.1]
β	member of P_f	[0 ... 2]
Δ_t	applied learning time	[4 ... 10]
Δ_k	assumed difference between the deadlines	[1 ... 6]
Δ_E	error threshold	[0.001 ... 0.1]
Δ_U	utility added value	[-0.01 ... 0.01]
P^s	percentage of sellers that only use $f \in \mathbf{F_s}$	[0 ... 100]

TABLE 5.5: The independent variables.

The seller agents in this evaluation are characterized in a similar fashion to the ones set up in the previous experiments (see section 3.5.2). The only difference is that in any given experimental run, a fixed percentage of the sellers (P^s) will now use the strategy that belong to $\mathbf{F_s}$. The rest of the sellers will use different strategies (e.g. tit-for-tat and trade-off - see section 2.2.1 for more details).

Variables	*Descriptions*
U	the utility value of the final agreement
N	the number of successful negotiations
A	the accuracy of the buyer's prediction

TABLE 5.6: The dependent variables.

5.3.2 Hypotheses

In a similar way to section 4.3.2, we want to evaluate the performance of our model; specifically by using the new adaptive negotiation strategy against different types of sellers. We aim to assert the results in terms of the number of successful negotiations and the average utility value of the final agreement obtained. Next, we aim to find the most influential variables and to find out how to set their values in order to gain best overall performance in each environment. We now turn to the specific hypotheses.

HYPOTHESIS 18. If there are sufficient sellers that use the predefined strategy functions, the buyer gains better outcomes using the learning negotiation strategy.

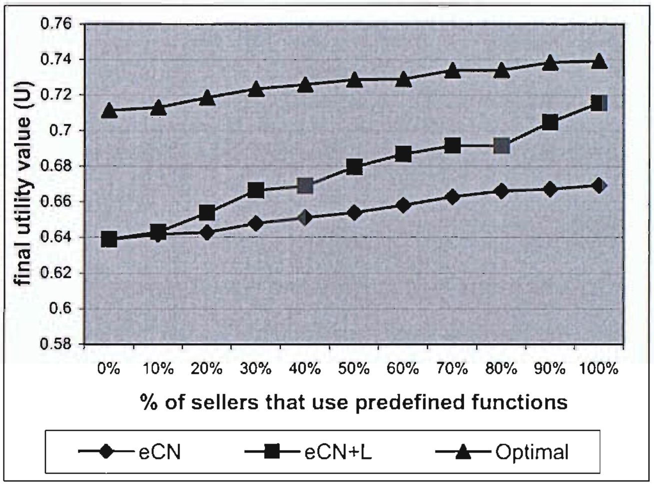


FIGURE 5.5: Final utility value for varying P^s.

To this end, figure 5.5 shows the average utility value of the final agreement achieved by the buyer in our standard concurrent negotiation (eCN - as detailed in chapter 3) and with the learning negotiation strategy applied (eCN + L). We also include the optimal outcome which is computed as per section 3.5.6. As can be seen, the greater the percentage of the sellers that use only the predefined strategy functions (P^s), the greater the utility value the buyer agent can obtain and the smaller the gap between the result achieved and the optimal. This can be explained by two reasons. First, the buyer agent in (eCN + L) uses the same basic negotiation strategy as eCN, the learning strategy is only used when the agent is reasonably certain that it can recognize its opponent strategy. If it is not certain, it will use the specific negotiation strategy allocated by the coordinator, just like its counterpart in eCN. Now in most cases, this ensures that the result achieved by (eCN + L) is not less than that of eCN. Second, when the prediction of the seller's reservation value (RP_k) is accurate (see below), the offer generated by the buyer is likely to be accepted by that seller. This will, in turn, give the buyer a reasonably high utility value since the generated offer is based on the value of that prediction and this is close to the seller's actual reservation value.

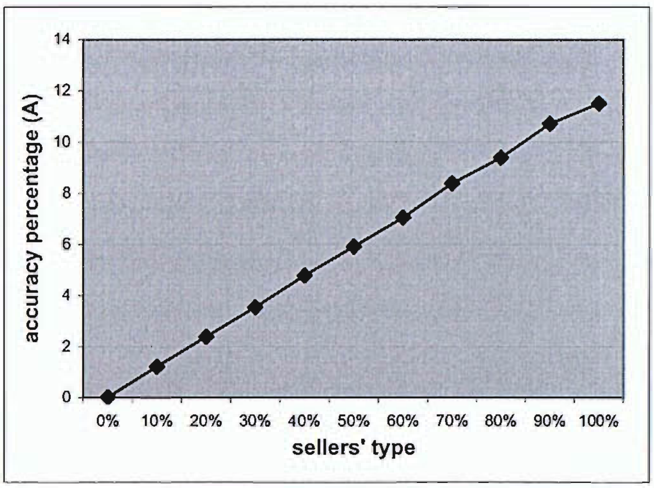


FIGURE 5.6: Buyer's prediction accuracy vs sellers' types.

To validate this line of argument, we measured the accuracy of the prediction of the seller's reservation value compared with the actual value (we consider a prediction is accurate if the generated value differs from the actual value by less than 5%). The results are displayed in figure 5.6. As can be seen, 11.5% of the buyer's predictions are accurate when 100% of the participating sellers use the predefined strategy functions[3]. When these predictions are accurate, this helps the buyer to propose high value offers to the seller that get accepted. Consequently, the final utility value achieved is increased.

	0%	*10%*	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*	*80%*	*90%*	*100%*
eCN	957	959	961	962	964	966	967	968	970	972	973
eCN+L	957	960	963	965	967	971	974	977	978	980	983

TABLE 5.7: Number of successful negotiations.

Now, not only does the final utility value increase, the number of successful negotiations also rises, albeit somewhat modesty (see table 5.7). Recall the example in section 5.2, if the buyer only uses the strategy allocated by the coordinator, it might only come to an agreement at time 10 (the other times, its offer value is outside the range of the seller). Now if the deadline of the seller is less than 10, no agreement will be reached. On the other hand, by using the adaptive negotiation strategy, the buyer is able to come to an

[3]The accuracy is somewhat limited because the buyer has to make its predictions based on its assumption about the seller's deadline and utility function and these are very difficult to approximate in the environments we consider.

agreement at time 5, which also has a higher utility value than the one it might have generated at time 10. As can be seen, in some cases, by making a correct prediction, the buyer is sometimes able to find an agreement where no agreement would be found if the buyer only follows the standard negotiation strategy.

HYPOTHESIS 19. The learning strategy performs most effectively when it is given sufficient negotiation time.

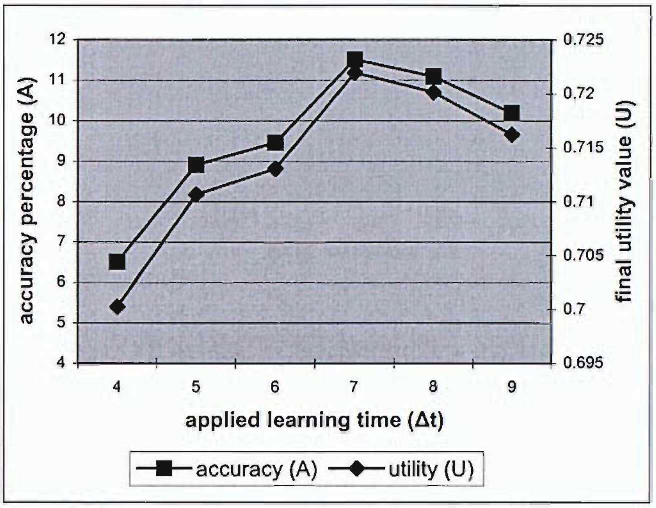


FIGURE 5.7: Buyer's prediction accuracy vs applied learning time.

To evaluate this hypothesis, we measured the buyer's prediction accuracy with different values of the applied learning time (the time in which the adaptive strategy starts to be applied - Δ_t) (see figure 5.7)[4]. As can be seen, the accuracy increases from 6.5% when $\Delta_t = 4$, to 11.5% when $\Delta_t = 7$ and then decreases afterwards. Similarly, the average final utility value achieved also increases and decreases in the same fashion. This fluctuation is caused by the number of samples that the buyer uses to analyze and learn the seller's reservation value. If the sample size is too small (less than 7 in this case), it is likely that there will be more than one possible combination of the function and its parameters that give the smallest error value. Thus, it is more likely that the buyer will not be able to select the correct combination. Consequently, the value learned is inaccurate. On the other hand, if the sample size is too large (greater than 7 in this case), there may not be a combination that gives an error value within the error threshold and, thus, the buyer will not be able to make a prediction. Overall then, we conclude that the buyer is better off applying the learning strategy after 7 rounds of counter-offers received.

[4]This hypothesis is similar to hypothesis 6 in a sense that we have to determine when the buyer agent should adapt its strategy based on the behaviors of the corresponding seller. However, here we do not measure the analysis time as a percentage of the buyer deadline since we are now concerned with the actual sample size for our learning algorithm to be applied.

HYPOTHESIS 20. In order for the model to achieve good performance, the error threshold should not be set too low.

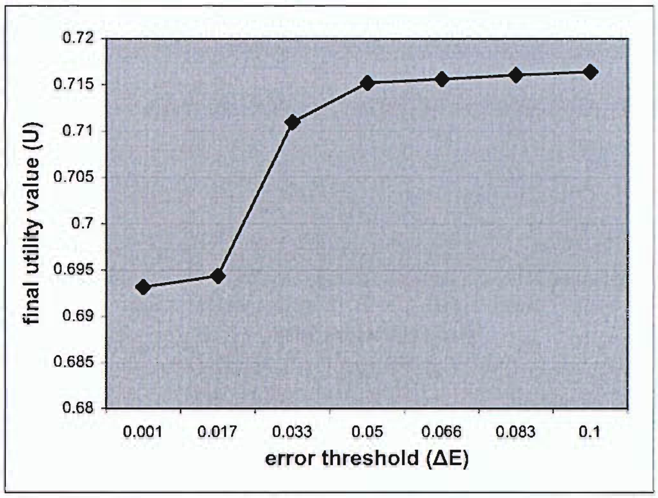


FIGURE 5.8: Buyer's prediction accuracy vs error threshold.

To evaluate this hypothesis, we measure the average utility value of the final agreement with different values of the error threshold (Δ_E) (see figure 5.8). As can be seen, the average value increases from 0.693 when $\Delta_E = 0.001$ to 0.717 when $\Delta_E = 0.05$ and, broadly speaking, stays at the same value when $\Delta_E > 0.05$. When Δ_E is low (less than 0.05 in this case), there are many situations when the buyer accurately predicted the value for RP_k (very close to the actual value) but it produces an error value $E(f)$ just above Δ_E and thus the learning strategy was not be used to generate the offers to propose to that seller. This causes the buyer to revert back to its standard negotiation strategy and miss the chance of obtaining a good agreement. On the other hand, when Δ_E is high, it makes almost no difference to the buyer's results since most of the predicted values will produce an error value $E(f)$ that is less than Δ_E, thus, there a very little change to the final outcome.

We also look at the effect of the utility threshold (Δ_U) on the performance of the model. Recall this was introduced as a method for compensating for the error when predicting the value of RP_k. Now, since the buyer uses a number of assumptions when making its prediction about the seller's reservation value, it is unlikely that the predicted value will match the actual one. Thus, by applying Δ_U to the final utility value of the offer that is proposed to the seller, we expect to minimize the difference between the predicted value and the actual one. To evaluate this hypothesis, we measured the average utility value of the final agreement for different values of Δ_U (see figure 5.9). As can be seen, in general, it turns out that setting a non-zero value for Δ_U does not improve the utility value of the final agreement. The reason for this is that in our experiments, by applying

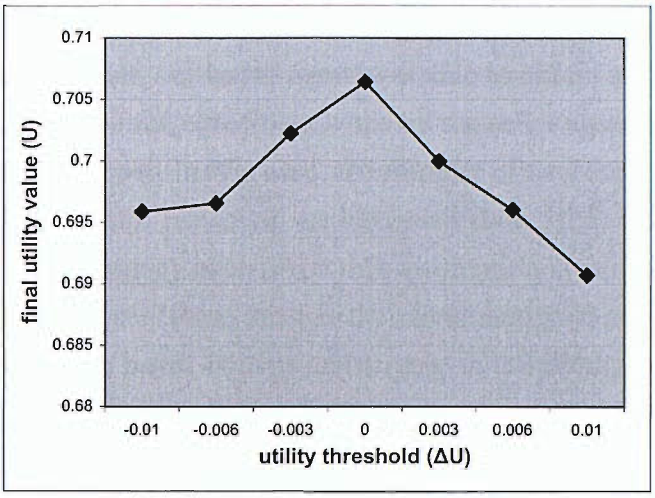


FIGURE 5.9: Buyer's prediction accuracy vs utility threshold.

a non-zero value of Δ_U into the predicted value of RP_k, it either takes the new value of RP_k out of the range of the seller or the utility gained is lower than that which could be achieved if the buyer just follows the standard negotiation strategy. Thus, the value for this parameter should be set to 0 in all cases.

5.4 Summary

This chapter has introduced an adaptive negotiation strategy that the buyer agent can use in order to negotiate more effectively and obtain better outcomes. It is based on predicting the seller's reservation value and generating offers based on this predicted value. This learning strategy is an add-on to the buyer's negotiation strategy developed in chapter 3 and does not standalone. Specifically, in order to make the prediction, the buyer makes use of the values of the previously proposed counter-offers by that particular seller. It then attempts to find a particular strategy that generates the values that is close to these proposed values. Then if this match is sufficiently close, this strategy will be used to predict the reservation value of the seller and this particular value will then be used in generating the next offer to be proposed to that particular seller. If it is not sufficiently close, however, then the buyer just continues using the strategy that was formally allocated by the coordinator.

The learning strategy has a number of parameters that the buyer can alter in order to provide flexible behavior in a range of different contexts. It operates under the assumption that the sellers use particular types of negotiation strategies. In such cases, our empirical evaluation shows that the buyer is able to gain better outcomes if this assumption is satisfied. In particular, our buyer agent was able to obtain an 8% increase in the average utility value of the final agreement when all the seller agents only use these negotiation strategies. Moreover, in this case, the number of successful negotiations also increase, albeit only modesty. However, we have also shown that the buyer is still able to gain benefit from this strategy even when this assumption is only partly satisfied. As the percentage of these sellers (those that use the aforementioned negotiation strategies) increases from 0 to 90, our buyer obtains an increase in the average utility value of the final agreement that ranges from 3% to 7.5%, respectively. Similarly, the number of successful negotiation also increases.

Chapter 6

Conclusions and Future Work

This chapter presents the conclusions from the work undertaken in this thesis and discusses the main ways in which this research can be carried forward in the future. It starts with a recap of the contribution of this research (section 6.1), followed by an outline of the areas where further research is still needed (section 6.2).

6.1 Research Contributions

The work presented in this thesis has developed a model to manage multiple concurrent negotiations in complex service-oriented negotiation settings. Specifically, novel mechanisms were developed for:

- the *managing phase*: allowing a buyer agent to negotiate simultaneously with a number of service provides in order to obtain a single high value agreement at the end of the bargaining encounter (chapter 3).
- the *commitment handling phase*: flexibly handling commitment and decommitment to negotiated contracts among the participating agents so that agents can make rational choices about their negotiation contracts (chapter 4).
- the *learning phase*: using heuristic-based learning to enable the buyer agent to negotiate more effectively by enabling it to learn the opponent's reservation value so that it can obtain a high value deal (chapter 5).

In more detail, in the managing phase, we have developed a novel coordination mechanism that allows the buyer agent to manage and control multiple bilateral negotiations simultaneously. In particular, we have presented a method for coordinating the various threads involved in the concurrent negotiations. Specifically, the agent uses multiple concurrent threads to negotiate, in which each thread handles a single provider. The result (an agreed contract) from a finished thread can then be used to influence other ongoing threads (so that they can alter their negotiation behaviors in the search for better deals). We have also presented a method to classify the seller agents, both prior to the negotiation and during the encounter. It is also shown, through empirical evaluation, that this classification helps the buyer agent in achieving a higher value agreement in terms of utility value obtained. Our model is computationally tractable (since it is heuristic-based) and is capable of working with incomplete information about the other agents (we do not require any special assumption; however, if we know about the probability distribution of the seller type, it will increase the performance of our model). The model has been empirically evaluated to show that it is capable of saving both time and resources in completing the negotiations (in comparison to the corresponding sequential model). The results are achieved while still producing a better outcome in terms of both the utility value gained and the number of agreements reached. Furthermore, we have also shown that our model outperforms the only other existing model in the literature (namely the work of Rahwan et al).

The model developed in this thesis has also been applied in two real world settings. First, it was successfully used in a commercial project developed by BT that focuses on automatically handling web services among customers. In this case, the model is responsible for finding an alternative replacement for a provider if the selected one ceased functioning. Second, the model was also used in the *CONOISE* project, that focuses on establishing and maintaining a *Virtual Organization* amongst a number of agents from different sources. Similarly, in this case, it was used to find an alternative provider to ensure that the generated VO operates smoothly.

Next, in the commitment handling phase, we have successfully extended the original leveled commitment [Sandholm, 1999b] so that it can be used in our multiple providers context. The commitment model is introduced to relax the constraint imposed in our basic concurrent model that any agreement made during the negotiation is binding on the seller only. Thus, in the initial model, the buyer has the exclusive right to back down from its previously agreed deals without any penalty to itself. This is clearly unrealistic and unfair to the other agents. In the modified model, once committed to a deal, if any agent decides to back down, it will have to pay a penalty fee to the other partner. This applies equally to both the buyer and the seller agents; thus the privileged position of the buyer agent is removed. To accommodate these changes, the buyer's internal reasoning process is modified. Instead of taking any acceptable offers, it commits to an offer if and only if that offer is better than what it currently has and it satisfies the degree of acceptance test (which takes into account a number of parameters including the predicted value of the next offer from all the sellers, how much time it has left, and so on). In addition, the buyer also has an option of keeping more than one commitment at any time. In this way, it can ensure that its chance of obtaining an agreement at the end of the process is increased. However, we show that the value of the final agreement it ends up with is often not as high as the one it would have had if it had only made a single commitment.

Finally, in the learning phase, we have used a heuristic approach to analyze the sellers based upon the counter offers they make. Specifically, we aim to predict the reservation value of individual sellers in order to determine which offer to propose to them. In particular, by having an idea of the reservation value of the seller, the buyer agent can make an offer which is likely to be accepted by that particular seller. Thus, the underlying motivation is that if this offer is accepted, it will be more beneficial for the buyer agent, in terms of utility value, than what could be achieve if the buyer follows the standard negotiation strategy. In more detail, we develop an algorithm to predict the reservation value that is composed of three main steps: (1) collect the samples (which is the offers that have been proposed by a particular seller), (2) find the closest matching negotiation function and its parameters based on those samples and (3) measure the difference between the values generated by this function and the actual values received. If this difference is reasonably small, an offer based on the predicted value will be proposed to the sellers. In a similar way to the previous phases, empirical evaluation has been used to show that by applying this learning based strategy, the buyer agent manages to obtain better outcomes in a number of different negotiation scenarios.

When taken together, these three phases provide an extensive and flexible concurrent negotiation model that a buyer agent can use to find the most appropriate service provider when there are a number of different possibilities. The commitment and the learning phases are designed to smoothly integrate with the managing phase, but they can also be used together to create a more complex and efficient negotiation model.

To sum up, against the requirements stated in section 1.2, our eCN model is

1. *computationally tractable*: all the mechanisms and algorithms developed in eCN are heuristic-based and have very little computational requirement.
2. *incomplete information*: eCN is able to operate without any explicit information about the participating seller agents.

3. *partial information*: eCN is able to exploit partial information about its opponents (e.g. information about their strategies, types and so on) to gain benefit for the negotiating agent if such information is available.

4. *deadlines*: all the agents in eCN have hard negotiation deadline.

5. *concurrent negotiations*: eCN is particularly designed to handle this special type of negotiation.

6. *efficient negotiation outcomes*: it has been shown that, via empirical evaluation, eCN outperforms existing models in the literature and the results obtained are close to the optimal that could have been achieved with complete information.

7. *commitments*: eCN has a commitment model which is:

    - *computationally tractable and flexible*: similar to the basic negotiation model, all the mechanisms and algorithms developed to handle commitments are heuristic-based and have very little computational requirement.
    - *efficient*: again, it has been shown, via empirical evaluation, that the results obtained are close to the optimal.

From the results achieved in the real world applications, as well as from our empirical results, we believe that our model can efficiently and effectively perform in a variety of negotiation settings. However, there are still a number of ways in which this model can be enhanced and these are detailed in the next section.

6.2 Future Work

There are a number of directions in which this work can be extended. First, the negotiation protocol used in our model requires that all the provider agents put forward their decisions at a single time period (see section 3.2 and 4.1.1). This is required in order for the buyer agent to make an informed decision about which offer to accept and which counter-offers to be proposed otherwise. However, in many practical scenarios, not all the providers agents will give their responses at the same time. Rather, it is more likely that the sellers will negotiate with each of the negotiation threads in an asynchronous manner and thus their corresponding decisions will need to be made at different time. This time frame variability will mean that the buyer will not be able to have all the offers from the sellers before making its decision. This is likely to have an adverse affect on the model's performance because the model has not been designed to accommodate this. Thus, future extensions to this work should consider the situation where the buyer agent negotiates with different providers in different time frames. To this end, one possible solution to this is that instead of making decisions based on each individual negotiation thread, the buyer agent will make its decision based on the previous responses from all the provider agents. This, we believe, can potentially give the buyer sufficient information to make it decisions. Also, we have only considered the situations in which the buyer adapts its negotiation strategy based on its analysis of the participating sellers. We have not considered scenarios in which the sellers also have this capability for the reason of efficiency for the buyer's tactics. This limitation needs to be addressed in the future work in order to increase the applicability of our concurrent negotiation model.

The next potential extension is considered with the commitment model. In the work described in this thesis, we have mostly focused on the committing strategy for the buyer agent (i.e. when it should commit, how many commitments it should take on at any one time, and so on). However, we have not looked in detail at the problem of the sellers decommitting. In our model, the buyer agent is passive with regards to these situations. Specifically, if a particular seller decides to back down from its contract (for whatever reason), our buyer agent needs to find a replacement. Now, depending on the situation, it might or it might not be able to secure one. However, we believe that if the buyer actively analyzes the sellers' behavior (possibly by taking into account the negotiation history of the sellers), it can minimize the risk of not ending up with an agreement at the end of the negotiation process. Thus, in the future, we would like to investigate the situation where the buyer could potentially look for a better replacement and actively decommit from its previously agreed contract, leaving itself a higher chance of securing a good final outcome.

The final main area of future work involves improving the way in which the buyer agent can modify its behavior according to the sellers' responses. The motivation behind this is to more flexibly change the buyer's negotiation strategy during the negotiation in order to obtain a higher value offer at the end of the bargaining process. In our basic concurrent model, this has been done via our sellers' classification process (see section 3.4.3), where the buyer potentially adopts different strategies based on the sellers' counter-offers received to date. This is a one time action that is performed, roughly about a third of the way to the deadline for the buyer agent, and it has been empirically shown to improve the performance of the model. We then extended this functionality by specifying an adaptive negotiation strategy for the buyer (see chapter 5). This strategy was designed to be broadly applicable because it is built upon relatively few assumptions about the sellers that the buyer agent is negotiating with. Basically, it analyzes the counter-offers received from a seller in a particular negotiation thread in order to try and predict that seller's lower reservation value and, later on, to use that value to generate an offer for the buyer agent to propose. The underlying intuition is that, if this offer is accepted, it is likely to give the buyer a higher utility value compared to the value of the offer generated by other negotiation strategies. However, although the empirical results show a reasonable improvement in the performance (about 8% in the cases we considered), the accuracy of the prediction is still somewhat low. Given this, future extensions should look into this area and attempt to increase the accuracy of the classification process and to extend the applicability of the adaptive negotiation strategy. In the former case, we could take into account other assumptions about the sellers such as their pricing structure or their negotiation deadline distribution. In the latter case, the algorithm needs to be extended to accommodate more generalized negotiation strategies such as those of game theory based or CSP-based models.

Bibliography

[Alonso, 2002] E. Alonso. Ai and agents: State of the art. *AI Magazine*, 23(3):25–30, 2002.

[Alpaydin, 2004] E. Alpaydin. *Introduction to Machine Learning (Adaptive Computation and Machine Learning).* The MIT Press, Cambridge, MA, USA, 2004.

[Andersson and Sandholm, 1999] M. Andersson and T. W. Sandholm. Time-quality tradeoffs in reallocative negotiation with combinatorial contract types. In *Proceedings ofthe National Conference on Artificial Intelligence (AAAI), Orlando, FL*, pages 3–10, 1999.

[Anthony, 2003] P. Anthony. *Bidding Agents for Multiple Heterogeneous Online Auctions.* PhD thesis, University of Southampton, Southampton, England, 2003.

[Arai *et al.*, 2000] S. Arai, K. Sycara, and T. R. Payne. Experience-based reinforcement learning to acquire effective behavior in a multiagent domain. In *The Sixth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2000)*, pages 125–135, 2000.

[Atkeson *et al.*, 1997] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73, 1997.

[Bannock *et al.*, 1992] G. Bannock, R. Baxter, and E. Davis. *Dictionary of Economics.* Penguin, London, UK, 5th edition, 1992.

[Bapna *et al.*, 2001] R. Bapna, P. Goes, and Gupta A. Insights and analyses of online auctions. *Communications of the ACM*, 44(111):43–50, 2001.

[Binmore, 1992] K. Binmore. *Fun and Games: A Text on Game Theory.* Heath and Company, Lexington, Massachusetts, 1992.

[Bond and Gasser, 1988] A. Bond and L. Gasser. *Readings in Distributed Artificial Intelligence.* Morgan Kaufmann Publishers, Inc., San Mateo, California, 1988.

[Bui *et al.*, 1996] H. H. Bui, D. Kieronska, and S. Venkatesh. Learning other agents' preferences in multiagent negotiation. In Howard Shrobe and Ted Senator, editors, *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference, Vol. 2*, pages 114–119, Menlo Park, California, 1996. AAAI Press.

[Calisti, 2001] M. Calisti. *Coordination and Negotiation for Solving Multi-Provider Service Provisioning*. PhD thesis, EPFL, Lausane, Swizerland, 2001.

[Carley and Gasser, 1999] K. M. Carley and L. Gasser. Computational organization theory. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 299–330. The MIT Press, Cambridge, MA, USA, 1999.

[Castelfranchi and Conti, 1998] C. Castelfranchi and R. Conti. Limits of economic and strategic rationality for agents and M-A systems. *Robotics and Autonomous Systems (Special Issue on Multi-Agent Rationality)*, 24(3-4):127–139, 1998.

[Castelfranchi, 1998] C. Castelfranchi. Modelling social action for ai agents. *Artificial Intelligence*, 103(1-2):157–182, 1998.

[Claus and Boutilier, 1998] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 746–752, Madison, CA, USA, 1998. AAAI Press.

[Cohen, 1995] P. Cohen. *Empirical Methods for Artificial Intelligence*. MIT Press, Cambridge, Massachusetts, 1995.

[David *et al.*, 2002] E. David, R. Azoulay-Schwartz, and S. Kraus. Protocols and strategies for automated multi-attribute auctions. In *Proceeding of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 77–85, Bologna, Italy, 2002.

[Davis and Smith, 1983] R. Davis and R. G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20(1):63–109, 1983.

[Dayhoff, 1990] J. E. Dayhoff. *Neural Network Architectures: An Introduction*. John Wiley and Sons, 1990.

[de Roure *et al.*, 2003] D. de Roure, M. Baker, N. R. Jennings, and N. Shadbolt. The evolution of the grid. In F. Berman, G. Fox, and A. J. G. Hey, editors, *Grid Computing: Making the Global Infrastructure a Reality*, pages 65–100. Wiley and Sons, 2003.

[Decker and Lesser, 1995] K. Decker and V. R. Lesser. Designing a family of coordination algorithms. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS)*, pages 73–80, San Francisco, CA, 1995.

[Decker, 1995] K. Decker. *Environment Centered Analysis and Design of Coordination Mechanisms, PhD Thesis*. PhD thesis, University of Massachusetts, Amherst, USA, 1995.

[Dellarocas, 2000] C. Dellarocas. Contractual agent societies: Negotiated shared context and social control in open multi-agent systems. In *Proceedings of the Workshop on Norms and Institutions in Multi-Agent Systems, 4th International Conference on Multi-Agent Systems (Agents-2000)*, Barcelona, Spain, 2000.

[Dubois and Prade, 1999] D. Dubois and H. Prade. Qualitative possibility theory and its application to constraint satisfaction and decision under uncertainty. *International of Intelligence Systems*, 14:45–61, 1999.

[Dubois *et al.*, 1994] D. Dubois, H. Fargier, and H. Prade. Propagation and satisfaction of flexible constraints. In R. Yager and L. Zadeh, editors, *Fuzzy Sets, Neural Networks and Soft Computing*, pages 166–187. Van Nostrand Reinhold, New York, USA, 1994.

[Durfee and Lesser, 1991] E. H. Durfee and V. R. Lesser. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5):1167–1183, 1991.

[Excelente-Toledo *et al.*, 2001] C. B. Excelente-Toledo, R. A. Bourne, and N. R. Jennings. Reasoning about commitments and penalties for coordination between autonomous agents. In *Proceedings of the Fifth International Conference on Autonomous Agents (Agents-2001)*, pages 131–138, Montreal, Canada, 2001.

[Excelente-Toledo, 2003] C. B. Excelente-Toledo. *The Dynamic Selection Of Coordination Mechanisms In Multiagent Systems*. PhD thesis, University of Southampton, Southampton, England, 2003.

[Faratin *et al.*, 2000] P. Faratin, N. R. Jennings, P. Buckle, and C. Sierra. Automated negotiation for provisioning virtual private networks using fipacompliant agents. In *Proceedings of the Fifth International Conference on The Practical Application of Intelligent Agents and MultiAgent Technology (PAAM2000)*, pages 185–202, Manchester, UK, 2000.

[Faratin *et al.*, 2003] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make negotiation trade-offs. *Artificial Intelligence Journal*, 142(2):205–237, 2003.

[Faratin, 2001] P. Faratin. *Automated Service Negotiation Between Autonomous Computational Agents.* PhD thesis, Queen Mary College, London, England, 2001.

[Fatima *et al.*, 2001] S. S. Fatima, M. Wooldridge, and N. R. Jennings. Optimal negotiation strategies for agents with incomplete information. In *Proc. of the Eighth International Workshop on Agent Theories, Architectures and Languages*, pages 53–68, Seattle, USA, 2001.

[Fox, 1981] M. S. Fox. An organizational view of distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1):70–80, 1981.

[Friedman and Rust, 1992] D. Friedman and J. Rust. *The Double Auction Market: Institutions, Theories and Evidence.* Addison - Wesley, 1992.

[Friedman, 1984] M. Friedman. *The Methodology of Positive Economics.* Allen & Unwin, Boston, USA, 1984.

[Fudenberg and Tirole, 1991] D. Fudenberg and J. Tirole. *Game Theory.* MIT Press, Cambridge, MA, USA, 1991.

[Genesereth and Ketchpel, 1994] M. R. Genesereth and S. P. Ketchpel. Software agents. *Communications of the ACM*, 37(7):48–53, 1994.

[Gibbons, 1992] R. Gibbons. *A Primer in Game Theory.* Harvester Wheatsheaf, New York, 1992.

[Hastings, 1997] K. J. Hastings. *Probability and Statistics.* Addison-Wesley, Reading, MA, 1997.

[Heckerman, 1999] D. Heckerman. A tutorial on learning with bayesian networks. In M. Jordan, editor, *Learning in Graphical Models.* The MIT Press, Cambridge, MA, USA, 1999.

[Hiltrop and Udall, 1995] J. M. Hiltrop and S. Udall. *The Essence of Negotiation.* Prentice-Hall, Englewood Cliffs, NJ, 1995.

[Huhns and Stephens, 1999] M. N. Huhns and L. M. Stephens. Multiagent systems and societies of agents. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 79–120. The MIT Press, Cambridge, MA, USA, 1999.

[Jennings *et al.*, 1998] N. R. Jennings, T.J. Normal, and P. Faratin. Adept: An agent-based approach to business process management. *ACM SIGMOD Record*, 27(4):32–39, 1998.

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

[Jennings, 1996] N. R. Jennings. Coordination techniques for distributed artificial intelligence. In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence. SixthGeneration Computer Technology Series*, pages 187–211. John Wiley, New York, 1996.

[Jennings, 2001] N. R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41, 2001.

[Johansson, 1999] S. J. Johansson. *Game Theory and Agents*. PhD thesis, University of Karlskrona/Ronneby, Ronneby, Sweden, 1999.

[Kaelbling *et al.*, 1996] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4(4):237–285, 1996.

[Kalakota and Whinston, 1996] R. Kalakota and A. B. Whinston. *Frontiers of Electronic Commerce*. Addison-Wesley Publishing, USA, 1996.

[Klemperer, 1999] P. Klemperer. Auction theory: A guide to literature. *Journal of Economic Surveys*, 13(3):227–286, 1999.

[Kowalczyk and Bui, 2001] R. Kowalczyk and V. Bui. On constraint-based reasoning in e-negotiation agents. In F. Dignum and U. Cortes, editors, *AgentMediated Electronic Commerce III, Current Issues in Agent-Based Electronic Commerce*, pages 31–46. Springer, 2001.

[Kraus and Schechter, 2003] S. Kraus and O. Schechter. Strategic negotiation for sharing a resource between two agents. *Computational Intelligence*, 19:9–41, 2003.

[Kraus, 1993] S. Kraus. Agents contracting tasks in noncollaborative environments. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pages 243–248, 1993.

[Kraus, 2001] S. Kraus. *Strategic Negotiation in Multi-Agent Environments*. MIT Press, Cambridge, USA, 2001.

[Lomuscio *et al.*, 2003] A. R. Lomuscio, M. Wooldridge, and N. R. Jennings. A classification scheme for negotiation in electronic commerce. *Int. J. of Group Decision and Negotiation*, 12(1):31–56, 2003.

[Luo *et al.*, 2003] X. Luo, H. Leung, J. H. Lee, and N. R. Jennings. A fuzzy constraint-based knowledge model for bi-lateral, multi-issue negotiations in competitive environments. *Artificial Intelligence*, 148(1-2):53–102, 2003.

[Maes *et al.*, 1999] P. Maes, R. H. Guttman, and Moukas A. Agents that buy and sell. *Communications of the ACM*, 42(3):81–91, 1999.

[Malone and Crowston, 1990] T. Malone and K. Crowston. What is coordination theory and how can it help design cooperative work systems. In *Proceeding of the International Conference on Computer Supported Cooperative Work (CSCW-90)*, pages 357–370, Los Angeles, CA, 1990.

[Malone *et al.*, 1988] T. Malone, R. Fikes, K. Grant, and M. Howard. Enterprise: A marketlike task scheduler for distributed computing environments. In B. Huberman, editor, *The Ecology of Computation*, pages 177–205. Amsterdam, 1988.

[Michie *et al.*, 1994] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine Learning, Neuraland Statistical Classification.* Ellis Norwood, 1994.

[Mitchel, 1997] T. Mitchel. Reinforcement learning. In *Machine Learning*, pages 366–390. McGraw-Hill, 1997.

[Moore *et al.*, 1995] A. W. Moore, D. J. Hill, and M. P. Johnson. *An empirical investigation of brute force to choose features, smoothers and function approximators*, volume III: Selecting Good Models, pages 361–379. MIT Press, 1995.

[Neumann and Morgenstern, 1944] J. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, USA, 1944.

[Nguyen and Jennings, 2003a] T. D. Nguyen and N. R. Jennings. Concurrent bi-lateral negotiation in agent systems. In *Proceedings of the Fourth DEXA Workshop on e-Negotiations*, pages 839–844, Prague, Czech Republic, 2003.

[Nguyen and Jennings, 2003b] T. D. Nguyen and N. R. Jennings. A heuristic model for concurrent bi-lateral negotiations in incomplete information settings. In *Proceedings of the Eighteenth International Joint Conference on AI*, pages 1467–1469, Acapulco, Mexico, 2003.

[Nguyen and Jennings, 2004a] T. D. Nguyen and N. R. Jennings. Coordinating multiple concurrent negotiations. In *Proceedings of the third International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 1064–1071, New York, USA, 2004.

[Nguyen and Jennings, 2004b] T. D. Nguyen and N. R. Jennings. Reasoning about commitments in multiple concurrent negotiations. In *Proceedings of the Sixth International Conference on E-Commerce*, pages 77–84, Delft, The Netherlands, 2004.

[Nguyen and Jennings, 2005] T. D. Nguyen and N. R. Jennings. Managing commitments in multiple concurrent negotiations. *Int. J. Electronic Commerce Research and Applications*, 4(3):to appear, 2005.

[Norman *et al.*, 2003] T. J. Norman, A. Preece, S. Chalmers, N. R. Jennings, M. Luck, V. D. Dang, T. D. Nguyen, V. Deora, J. Shao, A. Gray, and N. Fiddian. Conoise: Agent-based formation of virtual organisations. In *Proceedings of the twenty-third Annual International Conference of the British Computer Society's Specialist Group on Artificial Intelligence (SGAI)*, pages 353–366, Cambridge, UK, 2003.

[Norman *et al.*, 2004] T. J. Norman, A. Preece, S. Chalmers, N. R. Jennings, M. Luck, V. D. Dang, T. D. Nguyen, V. Deora, J. Shao, A. Gray, and N. Fiddian. Agent-based formation of virtual organisations. *Int. J. Knowledge Based Systems*, 17(2-4):103–111, 2004.

[Osborne and Rubinstein, 1994] M.J. Osborne and A. Rubinstein. *A course in Game Theory*. MIT Press, Cambridge, USA, 1994.

[Payne *et al.*, 2002] T. Payne, R. Singh, and K. Sycara. Calendar agents on the semantic web. *IEEE Intelligent Systems*, 17(3):84–86, 2002.

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

[Rahwan *et al.*, 2002] I. Rahwan, R. Kowalczyk, and H. H. Pham. Intelligent agents for automated one-to-many e-commerce negotiation. *Twenty-Fifth Australian Computer Science Conference*, 4:197–204, 2002.

[Rahwan *et al.*, 2003] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parson, and L. Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375, 2003.

[Raiffa, 1982] H. Raiffa. *The Art and Science of Negotiation*. Havard University Press, Cambridge, USA, 1982.

[Romp, 1997] G. Romp. *Game Theory: Introduction and Applications.* Oxford University Press, New York, USA, 1997.

[Rosenschein and Zlotkin, 1994] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers.* MIT Press, Cambridge, USA, 1994.

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

[Russell and Norvig, 2003] S. Russell and P. Norvig. *Artifical Intelligence: A Modern Approach.* Pearson Education, Prentice Hall, NJ, USA, 2 edition, 2003.

[Sandholm and Crites, 1995] T. W. Sandholm and R. H. Crites. On multiagent q-learning in a semicompetitive domain. In *Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), Workshop on Adaptation and Learning in Multiagent Systems*, pages 71–77, Montreal, Canada, 1995.

[Sandholm and Lesser, 1993] T. W. Sandholm and V. R. Lesser. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings ofthe National Conference on Artificial Intelligence (AAAI)*, pages 256–262, Washington, D.C., 1993.

[Sandholm and Lesser, 2001] T. W. Sandholm and V. R. Lesser. Leveled commitment contracts and strategic breach. *Games and Economic Behavior*, 35:212–270, 2001.

[Sandholm and Lesser, 2002] T. W. Sandholm and V. R. Lesser. Leveled commitment contracting: A backtracking instrument for multiagent systems. *AI Magazine*, 23:89–100, 2002.

[Sandholm, 1999a] T. W. Sandholm. Distributed rational decision making. *Games and Economic Behavior*, 35(1-2):304–338, 1999.

[Sandholm, 1999b] T. W. Sandholm. Distributed rational decision making. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 201–258. MIT Press, USA, 1999.

[Sen and Durfee, 1994] S. Sen and E. Durfee. The role of commitment in cooperative negotiation. *International Journal on Intelligent and Cooperative Information Systems*, 3(1):67–81, 1994.

[Sen and Durfee, 1998] S. Sen and E. Durfee. A formal study of distributed meeting scheduling. *Group Decision and Negotiation*, pages 265–289, 1998.

[Sen and Weiss, 1999] S. Sen and G. Weiss. Learning in multiagents systems. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 259–298. MIT Press, USA, 1999.

[Shao *et al.*, 2004] J. Shao, W. A. Gray, N. J. Fiddian, V. Deora, G. Shercliff, P. J. Stockreisser, T. J. Norman, A. Preece, P. M. D. Gray, S. Chalmers, N. Oren, N. R. Jennings, M. Luck, V. D. Dang, T. D. Nguyen, J. Patel, W. T. L. Teacy, and S. Thompson. Supporting formation and operation of virtual organisations in a grid environment. In *Proceedings of The UK OST e-Science Second All Hands Meeting 2004 (AHM'04)*. Nottingham, UK, 2004.

[Shen *et al.*, 2002] W. Shen, Y. Li, H. Ghenniwa, and C. Wang. Adaptive negotiation for agent-based grid computing. In *Proceedings of AAMAS 2002 Workshop on Agentcities: Challenges in Open Agent Environments*, pages 32–36, Bologna, Italy, 2002.

[Shoham and Tennenholtz, 1995] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, 73(1-2):231–252, 1995.

[Sierra *et al.*, 2002] C. Sierra, P. Faratin, and N.R. Jennings. A service-oriented negotiation model between autonomous agents. *Collaboration between Human and Artificial Societies - Coordination and Agent-Based Distributed Computing (ed. J. Padget), Lecture Notes in Artificial Intelligence*, 1624:201–220, 2002.

[Singh and Huhns, 2005] M. P. Singh and M. N. Huhns. *Service-Oriented Computing Semantics, Processes, Agents.* John Wiley and Sons, Ltd., 2005.

[Smith, 1980] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 29(12):1104–1113, 1980.

[Specht, 1991] D. F. Specht. A general regression neural network. *IEEE Transactions on Neural Networks*, 2(6):568–576, 1991.

[Sridharan and Tesauro, 2000] M. Sridharan and G. Tesauro. Multi-agent Q-learning and regression trees for automated pricing decisions. In *Proc. Seventeenth International Conference on Machine Learning*, pages 927–934. Morgan Kaufmann, San Francisco, CA, 2000.

[Sutton and Barto, 1998] R. S. Sutton and G. A. Barto. *Reinforcement Learning: an introduction.* The MIT Press, Cambridge, MA, 1998.

[Thompson *et al.*, 2004] S. Thompson, Y. Li, N. Giles, and H. Gharib. iservice and icollaborate - a framework and tool set or restfully creating service composition assistants. *BT Technical Documentation*, 2004.

[Vulkan and Jennings, 2000] N. Vulkan and N. R. Jennings. Efficient mechanisms for the supply of services in multi-agent environments. *Decision Support Systems*, 28:5–19, 2000.

[Wand and Jones, 1995] M. Wand and M. Jones. *Kernel Smoothing*. Chapman & Hall, London, UK, 1995.

[Wellman, 1993] M. P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1:1–23, 1993.

[Wolfstetter, 2002] E. Wolfstetter. Auctions: An introduction. *Journal of Economic Surveys*, 10:367–420, 2002.

[Wooldridge and Jennings, 1995] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[Wooldridge, 1997] M. Wooldridge. Agent-based software engineering. *IEE Proc Software Engineering*, 144(1):26–37, 1997.

[Wurman *et al.*, 2001] P. R. Wurman, M. P. Wellman, and W. E. Walsh. A parametrization of the auction design space. *Games and Economic Behavior*, 35:304–338, 2001.

[Yokoo and Ishida, 1999] M. Yokoo and T. Ishida. Search algorithms for agents. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 165–199. The MIT Press, Cambridge, MA, USA, 1999.

[Zeng and Sycara, 1997] D. Zeng and K. Sycara. How an an agent learn to negotiate. In J. Mueller, M. Wooldridge, and N. Jennings, editors, *Intelligent Agents III. Agent Theories, Architectures, and Languages*, pages 233–244. Springer Verlag, 1997.

[Zeng and Sycara, 1998] D. Zeng and K. Sycara. Bayesian learning in negotiation. *International Journal of Human Computer Systems*, 48:125–141, 1998.