Universidad Politécnica de Valencia

Departamento de Sistemas Informáticos y Computación

MSc.Thesis Proposal

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

<u>Title:</u>

Genetic-Aided Multi-Issue Bilateral Bargaining Model for Complex Utility Functions

Submitted by: Víctor Sánchez Anguix

Supervisors: Dr. Ana Garcia Fornes
Departamento de Sistemas Informáticos y Computación
Dr. Vicente Julián
Departamento de Sistemas Informáticos y Computación

Contents

Part I

Introduction

Motivation

Nowadays, the number of computational devices that are present in our everyday life has considerably grown. The use of technology looks to help us achieve a better quality of life, to make our life easier and more comfortable. However, due to the increasing number of devices, it is necessary that the technology itself adapts to the needs of the user, instead of the human being the one that adapts to technology. In that sense, Ambient Intelligence (AmI) tries to cover that necessity: it looks to offer personalized services and provide users with easier and more efficient ways to communicate and interact with other people and systems [1, 2].

Agent technology has been appointed as a proper technology for the support of AmI solutions [1, 3, 4]. As a matter of fact, agents show interesting characteristics for AmI environments since they are reactive, proactive and social [5]. First, reactiveness allows agents to change their behaviour according to some new conditions in the AmI environment (new users, new services, etc.). Second, proactiveness makes possible for agents to act autonomously according to the user's goals, which results in a smooth and non-intrusive interaction with the AmI user. Last, agent's social behaviour allows several heterogeneous entities to cooperate and offer new complex services to the AmI user.

In the last few years, researchers have shown a growing interest in automated negotiation. Negotiation can be defined as a process in which a joint decision is made by two or more parties. The parties first verbalize contradictory demands and then move towards agreement by a process of concession-making or search for new alternatives [6]. Therefore, automated negotiation consists in that such joint decision is automatically decided by means of autonomous entities (e.g., agents representing different users). The participant parties in a negotiation process have opposing preferences, thus negotiation can be considered as a conflict resolution mechanism.

Since the decade of the 50's, automated negotiation has been studied in game theory. Game theory researchers focused on reaching optimal solutions under the assumption of unbounded computational resources and complete information of the opponent preferences. Some of the most important theoretical results come from game theory, like the work of Nash [7], Rubinstein [8] and Binmore [9]. Although game theory studies obtained interesting results, most of them can not be applied to negotiations carried out by computer systems since there are limitations on the information of the opponent preferences, and computational resources available[10, 11]. However, artificial intelligence (AI) researchers have focused on working in such environments where there is not perfect knowledge about opponent preferences, and computational resources are bounded. Consequently, AI goal has been to reach good solutions instead of optimal solutions. However it must be noted that both fields' results are not opposed. Game theory results are a powerful tool in order to compare the results and properties of the negotiation methods devised in AI.

First AI works in automated negotiation are related to the area of Negotiation Support Systems (NSS) [12, 13, 14]. Basically, they are decision support systems that help humans in selecting acceptable negotiation strategies in narrow domains. For instance, Vedder et al. [12] developed a system that suggests negotiation strategies to police negotiatiors in situations of hostage rescue. Another example of this kind of systems is PERSUADER [13, 14], which uses CBR techniques to help humans in selecting strategies for labor negotiations. Nevertheless, nowadays most of the works in automated negotiation do not focus in this kind of applications. Additionally, since human negotiation across the internet could be time consuming, automated negotiation has been seriously considered as a key for electronic commerce. More concretely there has been a growing interest in developing agent mediated electronic commerce [15, 16, 17]. The properties shown by software agents prove specially interesting in electronic market environments. Nowadays agent-mediated electronic commerce is still a very active line of work.

AmI domains are not alien to conflict situations where automated negotiation is needed. For instance, shopping malls may be converted into ubiquitous environments where several vendors offer their products to passing shoppers [18, 19]. In many cases, the shoppers know what they want but do not have time to check every shop that offers such product. A possible way of enhancing the customer experience is to automatically negotiate with all of the vendors. A list with the best agreements may be presented to the user through its mobile device. This way, the user does not have to check every possible shop since its mobile device has negotiated with every shop taking into account the user preferences. Nevertheless, there are also benefits for the vendors since automated negotiation allows a more flexible commerce than classic e-commerce. For instance, they may negotiate issues such as price, payment method, discounts, and dispatch dates, which is what often happens in traditional non-electronic commerce. Flexibility in e-commerce produces a fidelization of the clients since the vendor is able to adapt as much as possible to the client preferences. Therefore, automated negotiation is a proper technology for e-commerce-based AmI applications such as shopping malls.

Despite the specific domain where automated negotiation has been applied, Artificial intelligence has traditionally studied multi-issue negotiations where utility functions are represented as a linear combination of the issues involved in the negotiation process [20, 21, 22, 17]. In linear utility functions, issue values are usually monotonic, so these functions usually have a single global optimum and consequently, the utility function is easy to optimize. Nevertheless, most real world problems are hardly modelled by linear utility functions since they have a higher degree of complexity than the one offered by linear utility functions (e.g. e-commerce [23, 24, 25]). Some of the issues in the negotiation setting may present interdependence relationships. Thus, the value of the utility function may be drastically changed by the positive/negative synergy of interdependent issues. The result is that the utility function is no longer linear, and therefore there may be several local optima. Optimizing non-linear utility functions is hard by itself (e.g. may require non-linear optimizers such as simulated annealing, genetic algorithms, etc.), and so is learning opponent preferences and looking for good agreements. Utility functions that have the trait of being non-linear are usually known in the literature as complex utility functions.

In the last few years, there has been an effort to research negotiation strategies that are capable of working with such complex utility functions where issues may have relationships of interdependence. Works in these complex domains have focused on negotiation strategies that require a mediator [23, 25, 26, 27], or non-mediated strategies that are devised for very specific utility functions [24]. However, non-mediated strategies are more interesting from the point of view of AmI environments due to the fact that users enter and leave the system in an extremely dynamic way. Thus, it may be difficult to find a trusted mediator for every possible user. Although non-mediated strategies are more interesting from the point of view of different domains, there has been lack of work in non-mediated strategies for complex utility functions. The work of Lai et al. [28] presents a non-mediated strategy for general utility functions, which obviously includes complex utility functions. The strategy is based on the calculation of current iso-utility curves and a similarity heuristic that sends offers from the current iso-utility curve that are the most similar to the last offers received from the opponent. However, the entire calculation of the iso-utility curve may require an exhaustive exploration of the utility function, which may not be tractable in the case of a large number of issues. Furthermore, if the exploration of one's own utility function is not performed in an intelligent way, it may result that most of the offers sampled are of no use for the negotiation process since they might not interest the opponent. Mechanisms that sample as few offers as possible are needed, specially for environments where devices may have limited computational resources as AmI environments.

Research Goals

Albeit the fact that some scholars have deeply studied the automated negotiation topic, not much concern has been given to the specific domain of Ambient Intelligence. This domains presents certain particularities that need to be carefully treated. As it has been stated above, Ambient Intelligence applications mainly rely on mobile technologies and devices with extremely bounded computational capabilities. Most works on automated negotiation have focused on achieving economic efficiency (pareto efficiency, utility maximization, Nash bargaining point) but they have overlooked some aspects such as computational efficiency (number of offers sampled during the negotiation process, number of negotiation rounds, etc.). The problem is still more important in the area of complex utility functions, where the optimization process carried out by each agents in order to explore its own preferences requires additional computational resources.

In this work, a non-mediated bilateral multi-issue negotiation model for AmI environments is presented. Its main goal is to optimize the computational resources while maintaining a good performance in the negotiation process. The proposed model is inspired in the seminal work of Lai et al. [28]. The three main differences between this present work and the work of Lai et al. are: (i) The present approach assumes that it is not possible to exhaustively search the utility function. Before the negotiation process starts, each agent samples one's own utility function by means of a niching genetic algorithm (GA) [29, 30]. The effect of this sampling is that offers obtained are highly fit and significantly different;(ii) A few additional samples are obtained during the negotiation process by means of genetic operators that are applied over received offers and one's own offers. The heuristic behind this sampling is that offers obtained by genetic operators have genetic material from one's own agent and the opponent's offers. Thus, these new offers may be interesting for both parties. (iii) Genetic operators act as a learning mechanism that implicitly guides the offer sampling and selection of which offers must be sent to the opponent. Results show that the proposed work outperforms similarity heuristics that are able to sample the same number of offers before the negotiation process starts. Additionally, it is also shown how the proposed strategy is capable of achieving similar results than similarity heuristics that sample the entire utility function with far fewer samples. This result is accomplished due to the learning mechanism provided by genetic algorithms.

It must be stated this thesis is enclosed in the work carried out by the GTI-IA research group. GTI-IA's research is focused on the area of multi-agent systems. Multi-agent systems are distributed systems formed by special software entities named as agents. What makes multi-agent systems different from other distributed system is that these special entities may be autonomous, reactive to changes in the environment, proactive according to its user's goals, and social (they may need to cooperate with other agents in order to fulfill their own goals).

The group has been granted with several active research projects related to the area of multi-agent systems: *Agreement Technologies* (Consolider Ingenio CSD2007-00022), *Advances on Agreement Technologies for Computational Entities* (PROMETEO 2008/051), *COST Agreement Technologies* (IC0801-AT), *MAGENTIX II: Una plataforma para sistemas multiagente abiertos* (TIN2008-04446/TIN), *Organizaciones Virtuales Adaptativas: Arquitecturas y Métodos de Desarrollo* (TIN2009-13839-C03-01). Next, we will describe how this thesis is related to the different research projects carried out in our group.

- *Agreement Technologies* (Consolider Ingenio CSD2007-00022): This project is a joint effort among GTI-IA (UPV, Valencia), IIIA (CSIC, Barcelona), and CETINIA (Universidad Rey Juan Carlos, Madrid). The goal of this project is designing new mechanisms that allow distributed systems to coordinate and cooperate in open and dynamic environments. Automated negotiation is pointed out as one of the core technologies that will allow computers to cooperate and coordinate in such dynamic open environments.

- *Advances on Agreement Technologies for Computational Entities (ATfCE)*: This is a project granted by *Generalitat Valenciana*. It also focus on agreement technologies as the proper paradigm for coordinating new generation distributed systems. How automated negotiation is related to agreement technologies has already been stated above.

- *COST Agreement Technologies*: This is a European project where the goal is the same as the one stated in the two previous projects.

- *MAGENTIX II: Una plataforma para sistemas multiagente abiertos*: Magentix is a multi-agent support platform for multi-agent systems. Multi-agent systems require special infraestructures in order to be conveniently deployed. Since automated negotiation is bound to be one of the key technologies in agent research, it is necessary to analyze how current infraestructures need to be adapted to support automated negotiation.

- *Organizaciones Virtuales Adaptativas: Arquitecturas y Métodos de Desarrollo*: A Virtual Organization is a complex entity where dynamic collections of individuals and institutions agree to share resources (software services, computational resources, etc.). Some works have already stated that Multi-agent systems (MAS) and agent organizations are one of the possible technologies to implement VO's. However, since Virtual Organizations may be situated in dynamic environments, they need to adapt themselves in order to maintain a certain level of performance. Automated negotiation is necessary for the different partners to reach a consensus concerning the changes to be applied to the Virtual Organization.

Thesis Structure

The remainder of this thesis is organized as follows. First, we review the state-of-the-art in automated negotiation. Two main sub-areas are reviewed: negotiation models for complex utility functions and negotiation models with learning capabilities. Why we study the first subarea is clear. We are interested in utility functions that are able to capture interdependence relationships among negotiation attributes. Learning capabilities are included because they allow to reach more efficient economic results than those that do not include them. Moreover, learning capabilities also allow to reach faster agreements, which is specially interesting in Ambient Intelligence domains. We also analyze those models that include both traits. At the end of each review, we include a brief discussion where we deeply analyze the main characteristics of both sub-areas.

In the next part we describe our proposed negotiation model. A chapter is dedicated to explain the negotiation protocol and its main advantages. After that, we describe the negotiation strategy. More specifically, we focus in the sampling carried out during the pre-negotiation phase in order to discover own good offers, the concession strategy, the acceptance criteria for opponent offers, the offer selection mechanism and, most importantly, the evolutionary sampling which is carried out during the negotiation process to discover new offers that are interesting for both agents. The final chapters of this part includes the experiment design and the results obtained by the proposed model. These results are analyzed and compared state-of-the-art similarity heuristics.

The final part of this thesis include a discussion of the work carried out during this thesis, work to be carried out in the future, and a list of publications related to this thesis.

Part II

State-of-the-art

Chapter 1

Complex Utility Functions Negotiation Models

Negotiation is an interaction and cooperation mechanism whose goal is to look for an agreement among two or more parties. For instance, buyers and sellers may negotiate about the price of a product in a electronic commerce domain. However, this approach is not the most convenient one in every real-world domain. As a matter of fact, most real-world negotiation processes negotiate about more than a single attribute. In the case of electronic commerce, buyers may include other negotiable services such as guarantee time, delivery time, and product quality. It is possible that a buyer is interested in incrementing the paid ammount in order to get an improvement in one of the mentioned services. The single-attribute approach results much more unnaceptable in more complex domains like team formation, resource distribution o every other situation that does not involve just money. The state-of-art in automated negotiation detected this problem and the focus of research evolved from single-attribute negotiations to multi-attribute negotiations.

Negotiation processes normally consist in the exchange of proposal between the involved parties. One of the key issues in negotiation strategies is how to value received proposals, and how to generate proposals that are likely to be accepted by the different parties. In processes where just a single attribute is involved it is quite clear how to evaluate and generate proposals: The value of the attribute. However, it is not easy to give a valuation when there is a need to negotiate for several attributes. The multi-attribute utility theory [31, 32] comes into play in this part. This theory provides mechanisms for the valuation of proposals composed of multiple attributes. This is normally accomplished by means of utility functions. First works in automated negotiation assumed that the attributes of the negotiation were independent in the utility function. Thus, proposals were valuated by means of a linear function where attributes where weighted according to its preference.

Despite the fact that linear functions perform well in some simple domains, there are still possible scenarios where they become poorly suited [33]. Just as an example we could think of a water market domain where two parties negotiate over the exploitation of several water resources. One of the parties desires to satisfy its water needs whereas the other party has got rights over several water exploitations. In this negotiation, the different attributes are the water exploitations to be included in the deal. Even although the provider offers a proposal whose ammount of water could satisfy the buyer, the value of the proposal may turn into a low utility for the buyer if the water sources are too distant. Thus, some attributes have a negative effect over the value of others and preferences can no longer be represented as linear utility functions. There is a need to provide complex utility functions that are capable of representing these complex preferences. Furthermore, the negotiation strategies that performed well in domains with linear utility functions may not perform equally in the case of complex utility functions. In fact, the search space for each agent is much more complicated. Therefore, new negotiation strategies adapted to complex utility functions are needed.

Current research in automated negotiation has started to provide negotiation strategies and complex utility functions that allow parties to work in these complex domains. In this chapter, some strategies for the complex utility function problem are reviewed.

1.1 Utility graphs

Robu et al.[24] introduces the idea of utility graphs and a negotiation strategy capable of working with utility graphs. The selected protocol is the bilateral multi-issue negotiation and it focuses on offers based on bundles of items (binary attributes) for electronic commerce.

Utility graphs are graphical models that relate negotiation issues that are dependent. Nodes represent negotiation issues whereas arcs connect issues that are at least included in one cluster. Intuitively, a cluster of items is a group of issues that have some joint effect in the final utility of a bundle offer. Therefore, the final utility of an offer can be viewed as the sum of the joint effects of each cluster. Consequently, items connected through arcs have certain dependency since they appear in a cluster. Since the size of the cluster is not theoretically limited, it is possible to represent interdependence relationships among attributes that are not restricted in its cardinality.

Robu et al.[24] propose a negotiation strategy where the buyer preferences and the seller preferences are modeled through utility graphs. In this approach, the seller is the one that carries out an exploration of the negotiation space to search for situations where both parties are satisfied. In order to accomplish this task, the seller has a model about the structure of the opponent utility graph due to past deals or information provided by experts [34]. This is specially feasible in electronic commerce domains. Robu also provided an strategy based on utility graphs that is capable of selecting which offers to send and learning/updating the model of the opponent.

1.2 Mediation for Complex Agreements

Klein et al.[35] is one of the seminal works in the complex utility function case. They argued that in most real-world domains attributes have interdependence relationships that directly affect on the utility of proposals. He devised a mediated negotiation protocol with the goal of the design of negotiation strategies for complex utility function domains. More specifically, Klein et al. focused on bundles of items (binary attributes). Preferences are modelled via a preference matrix. The content this matrix represents the value decrement or increment in the utility function when two different issues appear in a bundle offer. Therefore, this complex utility function can only take into account interdependent relationships that involve a pair of attributes. The mediator proposes offers to the different parties. Each party can reject or accept the proposal with a certain weight strength (weak or strong). If every party accepts the deal, or there is a weak reject and a strong accept, then the mediator mutates one of the attributes and sends back the new proposal to every participant. If a deal is rejected by one of the parties, then the mediator returns to the previous accepted deal and mutates one of the issues. This process is repeated until a fixed number of proposals is reached. Two different agents are studied as players in Klein et al.[35]: a hill-climbing agebt and one more complex agent based on simulated annealing. Hill-climber agents only accept a deal if it improves the best deal reached until the moment, whereas annealing agents have a probability of accepting a deal even if it does not improve the best past deal. The results show that the annealing approach obtains better deals, specially when all negotiating agents employ an annealing approach.

Ito et al. propose a multilateral negotiation strategy for complex utility functions based on the figure of a mediator[36]. The idea of this work is to find a consensus between more than two parties. Complex preferences are modeled through weighed constraints that add value to the utility function when they are satisfied. The number of attributes involved in one constraint is not limited a priori. Thus, the interdependence cardinality that the complex utility function is capable of taking into account is unlimited. Since agent preferences are complex, each agent samples its search space to find high utility areas. High utility areas are explored by means of simulated annealing in order to find local optimal bids. After such local optimal bids have been found, the agents find equally valued contracts in the near region for each bid. Finally these regions are sent to a mediator which looks for overlapping regions between agents and selects the one that maximizes the social welfare. The negotiation strategy was enhanced in [37] by allowing several negotiation steps. Agents are limited with respect to the number of regions they can send to the mediator. This solves scalability problems and keeps good deals providing that agents select the region bids to be sent intelligently. The mediator ask agents to submit promising space regions. Then, he finds overlapping regions between the bids received from the agents and sends them back to the other agents. By means of simulated annealing, each agent finds its locally optimal bids in the overlapping regions. Wide areas that include the local optimal bids are sent to the mediator. Finally, the mediator selects the overlapping region with the highest value for the social welfare function.

In [38], Lai et al. present a bilateral negotiation strategy based on an unbiased mediator. The focus of the work is to provide agents with proper a negotiation protocol and strategy that works in environments where agent preferences are complex and utility functions are not explicitely given (preference elicitation). The only assumption is that agents can compare which offer they prefer in a limited set of offers. The search space is divided into negotiation baselines so that agents do not need to search in a n-dimensional space. One of the parties, proposes an offer to the mediator in the negotiation baseline. Then, the mediator works with both parties in order to find a point that is Pareto optimal for both parties in the current negotiation baseline. If the point found by the mediator is not accepted by the other party the negotiation line is updated.

1.3 K-Alternating protocol for Complex Agreements

Lai also proposed a protocol and negotiation strategy [28] based on the alternating protocol. However, the parties are allowed to make k offers in each negotiation step in order to explore better the negotiation space. In each step, one of the parties chooses the offer from its iso-utility curve (points with the same utility) that is closer to the last offer received from the opponent that reported more utility. Then, it generates $k-1$ offers that are in the neighbourhood of the chosen offer. The k offers (chosen offer plus $k-1$ generated offers) are sent to the other party. The main advantage of this method is that it is general and does not assume any particular mechanism to model complex preferences.

1.4 Discussion

Complex utility functions has been a hot topic in the last few years. Research in this direction is still in its early stages and therefore there is still a long road ahead. Complex agent preferences cannot be modeled in a perfect way, thus there are multiple valid mechanisms that can be applied with success in different situations. The modeling mechanisms that have been analyzed in this paper are utility graphs[34, 24], weighted constraints[36, 37] and preference matrix[35]. On the other hand, there are works that do not take into account any specific complex utility function. For instance, Lai et al. presented a strategy that is independent of the underlying complex utility function [28] and a strategy that assume that agents may not have its utility function explicitely represented [38].

In the reviewed works, we can classify the different works according to the cardinality of the interdependence that the complex utility function is able to represent. Some works like Klein et al. [35] are able to represent interdependencies between pairs of attributes, whereas other works like Robu et al.[34, 24], and Ito et al.[36, 37] are not restricted in the number of attributes that form an interdependency relationship in the complex utility function. It must be highlighted that unrestricted interdependence cardinality in complex utility functions is preferred due to the fact that these models are able to cope with more complex problems. However, the number of complex utility functions with such characteristics is currently limited to weighted constraints[36, 37] and utility graphs[34, 24]. The study of new models for complex utility functions is a potential are of future work in the area.

Another issue that can be identified in these works is whether they use an unbiased mediator or not. Negotiations carried out by means of a mediator usually get better results for every party. This is usually related to the fact that agents are less reluctant to share more information with an unbiased mediator than with an opponent. Mediated strategies may help to cope with the problem of complex utility functions since a more exploratory strategy may be applied. Works that apply a mediated strategy are [36, 37, 38, 35]. Nevertheless, an unbiased mediator is not always possible in every real-world domain. Therefore, non-mediated strategies are also required in some domains. It has been detected that there is a lack of works that do not use mediator. In fact, only Robu et al.[24, 34] and Lai et al. [28] use a non-mediated strategy.

Finally, it has been detected that there is a lack of work in learning and opponent modeling when agents use complex utility functions. The reasons for this lack are two: the recent use of complex utility functions and the fact that learning in this domains is much more complex than learning in the linear case. Robu et al. [34, 24] applies explicit learning mechanisms to help the negotiation process, whereas Lai et al. [28] employs an implicit learning mechanisms that is able to adapt to opponent preferences since it sends offers from the current iso-utility curve that are closer to the last offers received from the opponent. The use of learning mechanisms in the case of complex utility functions is acknowledged as a potential area of work. A brief sketch of the discussion presented in this chapter can be found in Table 1.1.

Despite the fact that some work has been done regarding complex utility function, it mainly focus on mediated strategies. Mediated strategies may not be available in some AmI domains since they mainly consist of ad-hoc networks with limited bandwith. Mediated strategies as the ones proposed above make a great use of network bandwith, and therefore they may not suitable for AmI domains. Among non-mediated strategies, we are more interested in those negotiation models that are independent of the underlying complex utility function since they may be adapted to a wide variety of domains. However, many current models only focus on economic efficiency and do not take into account computational efficiency. Our goal is providing a solution for this problem.

	Protocol	Preference Model	Interdependency Cardinality	Mediation	Learning
Klein 2003[35]	Bilateral	Preference matrix	Binary	Yes	No
Robu 2005[24]	Bilateral	Utility graphs	Unrestricted	No	Yes
Ito 2006[36]	Multilateral	Weighted constraints	Unrestricted	Yes	No
Hattori 2007[37]	Multilateral	Weighted constraints	Unrestricted	Yes	No
Lai 2008b[38]	Bilateral	Preference elicitation	Independent	Yes	No
Lai 2008[28]	Bilateral	Independent	Independent	No	Yes

Table 1.1: Work classification in complex utility functions

Chapter 2

Negotiation Models With Learning Capabilities

One way to speed up the negotiation process and achieve good deals is to know the preferences of the opponent. Unfortunately, this is not possible in real situations. Consequently, alternatives must be found in order to reach good agreements fast. Even although it is logical to think that parties are willing to reach good agreements fast, reality is not as perfect as theory and parties involved in a negotiation process are reluctant to share information about their preferences. For instance, there is certain exploitation risk when revealing information about the reservation value and the private deadline established for the negotiation process due to the fact that self-interested parties could simply offer opponent reservation value when deadline is near. This forces the opponent to accept the worst deal possible. This situation becomes more plausible in open dynamic environments where parties may not have any information about the opponent intentions.

Albeit parties are not willing to share information with their opponents, it is still possible to try to reach good agreements for every party. Automated learning is one of the most researched areas in computer science. There are several algorithms and models that allow computers to deal with a vast set of learning problems. The goal of applying learning techniques to automated negotiation is to model the opponent preferences in order to reach good deals and possibly speed up the negotiation process.

Basically, there are two different trends in automated learning. They are related to the moment when the learning process is carried out. On one hand, the first approach is what is known as *offline learning*. In *offline learning*, the learning process is carried out prior to dealing with the real problem. The model remains static until it is re-trained with new data. When applied to automated negotiation, *offline learning* takes place after the negotiation process has been terminated with the new data generated from the negotiation. The new model is used in future negotiations. On the other hand, *online learning* supposes a dynamic adaptation of the model when placed in the real enviroment. More specifically, the adaptation takes place as the negotiation process advances. The goal is to adapt the current model, which may have been built from scratch at the start of the negotiation, to the real preferences of the opponent. Obviously this last approach is much more interesting from the automated negotiation viewpoint although is also harder to provide solutions with *online learning* since very little information may be revealed during the negotiation process.

In this chapter some negotiation strategies that make use of learning techniques are reviewed. At the end of the chapter, a discussion about these works is included.

2.1 Observing concessions from opponents

Jonker et al. presented a bilateral barganing model [39] where offers for a specific negotiation step are computed taking into account the previous offer utility and a concession step. Each attribute adjusts its value to fit the new computed utility, always bearing in mind the importance the agent grants to the specific attribute. In [17], Jonker et al. assume that negotiators are willing to share certain information about their preferences before the negotiation process. This is specially true in electronic commerce domains where the buyer might be willing to reveal what he considers important to trusted sellers. The model assumes independent issues that are linearly combined. The information revealed to the seller is some of the attribute weights of the linear utility function. During the negotiation process, the heuristic tries to guess the importance the opponent gives to the other issues by observing the concession rate. Seller concessions are made taking into account which attributes are more important for the buyer. The objective of the learning process is to adapt to buyer preferences during the negotiation process.

2.2 Similarity Criteria as implicit opponent modeling

Trade-off consists in decrementing the benefit obtained from some issues in order to get the decremented benefit as an equivalent increase in the benefit obtained by other issues. Intuitively, given a certain offer, a trade-off would be a different offer with exactly the same utility. The general idea is that although we get the same utility, we may offer a bid that is more interesting for the opponent. Faratin et al.[40] proposed a bilateral negotiation strategy based on a trade-off mechanism. The idea behind the strategy is to offer a bid with the same utility value than one's last bid, but closer to the last bid received from the opponent. The concept of similarity is adressed as a fuzzy similarity problem. The start point for the algorithm is the last bid received from the opponent. The algorithm performs in sequential steps. At each step random points from an iso-utility curve closer to one's own last offer are calculated. The bid selected as start point for the next step of the algorithm is the one closer to the opponent last bid. This process continues until reaching the iso-utility curve of the last offer sent. Although it does not model explicitely the preferences of the opponent, the mechanism has certain online adaptability to the preferences of the opponent because of its trade-off heuristic. Similarly, Lai et al.[28] proposes a negotiation strategy where the similarity criteria is the euclidean distance. It is capable of adapting to opponent preferences through implicit mechanisms.

2.3 Bayesian approach

From the large set of methods that solve the problem of explicit learning, one of the techniques that possibly fits better to the negotiation domain is probably the bayesian approach. Bayes is not only a powerful learning techinque for problems where no prior information is available, but also provides a computationally cheap update mechanism compared to other techniques like support vector machines and neural networks. This features make the Bayesian approach adequate for the negotiation problem.

One of the first approaches that used bayesian learning methods for automated negotiation was the work of Zeng et al. [41]. Its goal is to learn the opponent's reservation point using Bayesian learning in bilateral negotiations. However, this work only focused on single-issue negotiation. Later, Narayanan et al.[42] designed a negotiation strategy for bilateral negotiation that aimed to provide solutions in non-stationary environments(changes are possible during the negotiation process. Therefore, opponents may change strategies during the negotiation process. They used Markov chains and Bayesian learning in order to learn negotiation strategies that obtained optimal results under the above assumptions. However, the algorithm was only devised for single-issue negotiations. Hindriks et al. [43] presented a work where bayesian learning is used to learn attribute preferences in bilateral multi-attribute negotiations.

2.4 Genetic Algorithms

Genetic Algorithms (GA's) have also contributed in the state-of-art of offline learning in automated negotiation. The seminal work of GA's in Automated Negotiation was Oliver et al. [44]. Their work focus in evolving negotiation strategies for bilateral multi-issue negotiations. In their experiments, strategies were sequential rules and thus they were coded as chromosomes of the GA. A rule is a utility threshold for the proposals that come from the opponent and a proposal to make to the opponent in case that its offer is rejected. A random population of negotiation strategies is generated as initial population of the GA. In order to obtain the fitness of each individual, it is necessary to test the strategies against several opponents. The strategies that averaged the best pay-offs are selected as the parents of the new population, which is created through genetic operators . Although GA's allow to evolve negotiation strategies (obtaining implicit adaptability to the environment) the expressivity of the model, simple rules, is far from the complexity needed in real negotiation problems.

Faratin et al. [20] designed strategies to evaluate and generate proposals in an alternating offer protocol for non-cooperative bilateral multi-attribute negotiation. Strategies are composed of three families of tactics: according to the remaining time in negotiation, resource quantity and the behaviour observed in the opponent. Remaining time tactics are divided into boulware tactics, which concede slowly at early stages but do it faster as the remaining negotiation time decreases, and Conceder tactics, which concede faster at early stages. Moreover, behaviour dependent tactics are divided into telative tit-for-tat, which reproduces the opponent behaviour in relative terms, random absolute tit-for-tat, which reproduces the opponent behaviour in absolute terms with a random factor of increase or decrease, and averaged tit-for-tat, which reproduces the average relative behaviour of the opponent in a certain window of past time. Thus, Faratin et al. designed a total of six different tactics that were linearly combined for each attribute in order to obtain its value in each negotiation step. Matos et al. [45] enhanced Faratin's negotiation model[20] by adding *offline adaptability*. The general idea is to allow agents to evolve Faratin's strategies [20] in order to adapt themselves to the prevailing environment circumstances. A GA is used as a mechanism of evolution for these tactics. The parameters of the tactics are coded as chromosomes of the GA. In their experiments, populations of buyers and sellers with different strategies negotiate in a round robin way. After each round robin round, strategies are evaluated by means of a fitness function that involves the comparision between the obtained negotiation result, and the number of messages exchanged in the negotiation process. Then, strategies are selected to be the parents of the next population according to their fitness function. In the end, a population of strategies implicitely adapted to the environment is obtained.

Tu et al. [46] worked also on evolving negotiation strategies, but the negotiation strategies are represented as finite state machines (FSM). The general idea is the same that the applied by Oliver et al.[44]: A initial population of individuals, coded as FSM chromosomes, is generated randomly. After that, the evolution process starts by testing the strategies against several opponents. One of the advantadges of using FSMs is that they allow branching and certain memory in negotiation strategies. Arcs have an associated condition regarding the opponent proposal that needs to be satisfied in order to move to the next state. If the condition is satisfied, arcs have also an associated action (proposal) that is performed. After being tested, the strategies with the best fitness are selected as the mating pool for the genetic operators.

Other experiments involving GA and negotiation were carried out by Gerding et al.[47]. The author focuses on negotiation processes where the utility function is a linear combination of the different issues. Furthermore, issues are real values [0,1] that indicate the value of the issue that is assigned to one of the parties. The rest of the value is assigned to the other party. Despite the fact that the chosen representation for negotiation strategies is the one employed by Oliver et al.[44], the work of Gerding et al. is interesting because he introduces the idea of *fairness* and *social awareness* in the evolution of negotiation strategies. The first concept relates to the idea that low valued proposals have high probabilities of being rejected. Thus, agents should make proposals having into account the probability of being accepted by the opponent. The *social awareness* is the ability of agents to reject proposals in the last round provided that they are capable of negotiate with different parties.

Despite the fact that GA's have been used mostly in offline learning, there are also a few works that employ GA's as an adaptive online negotiation mechanism. Krovi et al.[48] proposes a GA for bilateral negotiations that is performed each time a negotiation round ends. The population of chromosomes is randomly initialized with 90 random offers and 10 heuristic offers: the last offer of the opponent and the nine best offers from the previous GA. Adaptive learning is achieved by mutating and crossing offers with the offer proposed by the opponent. Choi et al.[49] enhanced the model with more learning capabilities. More specifically, it is capable of learning opponent preferences by means of stochastic approximation and adapt its mutation rate to the opponent behaviour.

2.5 Utility Graphs revisited

As we mentioned in Section 1.1, Robu et al. [34, 24] introduces the idea of utility graphs. Besides the fact that utility graphs can be used to model complex utility functions, they also may be used to learn opponent preferences. In Robu et al., the buyer preferences are modelled by means of an utility graph. The model is updated as the negotiation process advances. However, the utility graph structure must be known a priori. The model was improved by building the buyer utility graph structure based on past negotiation data and recommendation techniques [34].

2.6 Discussion

One of the main differences among the strategies that have been analyzed above is whether they are applied during the negotiation process (online learning) or not (offline learning). On one hand, strategies that use online learning include [17, 40, 48, 49, 24, 34, 41, 42, 43, 28]. Online strategies are particularly interesting from the point of view of open and dynamic environments where it is possible that agents meet with their opponents just one or a few times. Furthermore, the dynamicity of the environment makes it possible to observe several behaviour or preference changes during the negotiation process. Consequently, mechanisms that are able to adapt during the negotiation process are required. It is acknowledged that learning during the negotiation process is much more complicated since very little useful information may be revealed.

On the other hand, most of the works that employ offline learning methods come from the works that use genetic algorithms. Genetic approaches are focused on learning evolutionary negotiation strategies that can adapt to different enviromental conditions or behaviours

[44, 45, 46, 47]. The use of evolutionary negotiation strategies seeks two goals. First, it is an excellent mechanism to study new negotiation strategies and behaviours that are to be applied under certain environmental conditions. For instance, it is possible to study in the laboratory which strategies work better against competitive agents. However, it requires knowledge about such environmental conditions, which may be quite unpredictable in open and dynamic environments. Second, it is possible to apply a set of reasonably good negotiation strategies to a real environment and optimize them for the prevailing environmental conditions. Nevertheless, it requires a set of negotiation strategies that are known to work well, since randonmly generated strategies may be too poor to be applied to a real environment. Additionally, strategies do not perform equally against different types of opponents, thus mechanisms to identify the type of opponent would be needed. Despite these inconveniences, GA provide a good framework to adapt existing strategies (strategy parameters, new strategies) to the prevailing environment conditions. It is specially true when the environment is not very dynamic and the the type of behaviours that can be observed in the environmental are well known.

It is also observable that there are strategies that maintain an explicit model of the opponent and strategies where the model of the opponent is implicit. In the first case, we can find works such as [17, 24, 34, 41, 43]. Keeping an explicit model of the opponent is specially interesting in domains where it is possible to face the same opponent several times. Explicit models allow to start with past information that may speed up the negotiation process from the beginning. The latter case, implicit models, includes works such as [44, 40, 48, 49, 45, 46, 47, 42, 28]. Explicit models may not be the most feasible approach in situations where it is unlikely that two opponents face more than a few times. Additionally, domains where the prefereces of the agents are very dynamic pose also a big disadvantage for strategies that use explicit models of the opponent since they may need to discard their models continuosly. In this cases, the implicit modeling approach may be more adequate. In any case, both approaches can be mixed: using implicit models for those opponents that are not to be encountered frequently and explicit models for those opponents that are to be found frequently.

Negotiation strategies that apply learning mechanisms also differ in what is the object of learning. This is usually related to whether the strategy explicitely models the opponent or not. In the case of explicit models, the object of learning are usually parameters that represent opponent's preferences. Works that learn parameter of the opponent's preferences include [41, 43, 17, 24, 34]. However, the object of learning may vary when implicit modeling is used. In [44, 46, 47, 42], the object of learning are negotiation strategies that are adapted to work optimally in certain environments. Another approach is to learn optimal parameters for one's own negotiation strategies, as presented in [45]. Additionally, there are also negotiation strategies that use implicit modeling and learn opponent preferences [40, 48, 49, 28].

Finally, it must be noted that most of the works focus on domains where complex utility functions are not used [43, 45, 17, 40, 48, 49]. Additionally, some of approaches like [44, 46, 47, 42, 28] are independent of the underlying utility function. The only work that is explicitely designed to learn in domains of complex utility functions, more specifically utility graphs, is [24, 34]. A brief sketch of the discussion presented in this chapter can be found in Table 2.1.

Since our goal is providing economically and computationally efficiently solutions for automated negotiations in a wide variety of AmI environments, we need to employ learning mechanisms. From this point of view, genetic algorithms may prove extremely interesting since they provide an implicit learning mechanism independent of the underlying utility function.

In the next chapter we present our bilateral negotiation model proposal. It aims to provide solutions for AmI environments where complex utility functions are used to model agent's preferences. Due to the particularities of AmI domains the proposed model does not rely on a trusted mediator, it does not assume an specific complex utility functions, it is optimized to offer efficient economic and computational solutions, and it relies on genetic algorithms in order to reach such solutions.

	Features				
	Protocol	Type of Learning	Type of Model	What is Learnt	Complex Utility Functions
Oliver et al. 1996[44]	Bilateral	Offline	Implicit	Negotiation Strategies	Independent
Matos et al. 1998[45]	Bilateral	Offline	Implicit	Optimal Parameters	No
Zeng et al. 1998[41]	Bilateral	Online	Explicit	Opponent Preferences	No
Krovi et al. 1999[48]	Bilateral	Online	Implicit	Opponent Preferences	No
Tu et al. 2000[46]	Bilateral	Offline	Implicit	Negotiation Strategies	Independent
Choi et al. 2001[49]	Bilateral	Online	Implicit	Opponent Preferences	No
Faratin et al. 2002 [40]	Bilateral	Online	Implicit	Opponent Preferences	No
Gerding et al. 2003[47]	Bilateral	Offline	Implicit	Negotiation Strategies	Independent
Jonker et al. 2004[17]	Bilateral	Online	Explicit	Opponent Preferences	No
Robu et al. 2006[34]	Bilateral	Online	Explicit	Opponent Preferences	Yes
Narayanan et al. 2006[42]	Bilateral	Online	Explicit	Negotiation Strategies	Independent
Hindriks et al. 2008[43]	Bilateral	Online	Explicit	Opponent Preferences	No
Lai et al. 2008[28]	Bilateral	Online	Implicit	Opponent Preferences	Independent

Table 2.1: Work classification in learning

Part III

Genetic-Aided Multi-Issue Bilateral Bargaining Model for Complex Utility Functions

Chapter 3

Negotiation Model

In this chapter we describe the proposed negotiation model. Negotiation models are composed of a negotiation protocol and a negotiation strategy. On the one hand, the negotiation protocol defines the communication rules to be followed by the agents that participate in the negotiation process. More specifically, it states in which moments the different agents are allowed to send messages and which kind of messages the agents are allowed to send. For instance, the Rubinstein alternating protocol specifies [50] that agents are allowed to send one offer in alternating turns. Basically, the negotiation protocol acts as mechanism for the coordination and regulation of the agents that take part in the negotiation process.

On the other hand, the negotiation strategy defines the different decisions that the agent will make at each step of the negotiation process. It includes the opponent's offers acceptance rule, the selection of which offers are to be sent to the opponent, the concession strategy, the decision of wheter the agent should continue in the negotiation process or not, and so forth. Therefore, the negotiation strategy includes all the decision-making mechanisms that are involved in the negotiation process.

The negotiation protocol used can be categorized as an alternating protocol for bilateral bargaining [50]. More specifically, the protocol used is the *k-alternating protocol* proposed by Lai et al. [28]. The proposed negotiation strategy belongs to the family of negotiation strategies that use a similarity heuristic in order to propose new offers to the opponent [21, 28].

3.1 Negotiation Protocol

As it was mentioned above, the negotiation protocol belongs to the family of alternating protocols for bilateral bargaining. In this kind of protocols, two different agents negotiate without the need of a mediator. As it has been previously stated, non-mediated strategies are more adequate for AmI applications since users enter and leave the AmI system in a very dynamic way. Thus, it may not be feasible to find a trusted mediator for every possible pair of agents. Furthermore, in some AmI domains as shopping malls, where there are different competing vendors and lots of potential users, it is difficult to determine who will mediate the negotiation process.

The protocol used is the *k-alternating protocol* proposed by Lai et al. [28]. This protocol is composed of several rounds where the agents exchange offers in an alternating way. One of the agents, called the *initiator*, is responsible for starting the current round. He can accept one of the previous offers received from the opponent in the last round, exit from the negotiation process, or send up to k different offers to the opponent agent. Once the *initiator* has performed one of the possible actions, the opponent agent is able to accept one of the offers he has just received, exit from the negotiation process or propose to the *initiator* up to k different offers. Then, the round ends and a new round is initiated by the *initiator* agent. The negotiation process ends when one of the agents accepts an offer (the negotiation succeeded) or one of the agents decides to abandon the negotiation (the negotiation failed).

Some of the properties of the *k-alternating protocol* proposed by Lai et al. are:

- The protocol is adequate for situations where both agents are equal in power (e.g. none of them has the monopoly over a resource).
- Each agent is capable of sending up to k different offers, being more probable that one

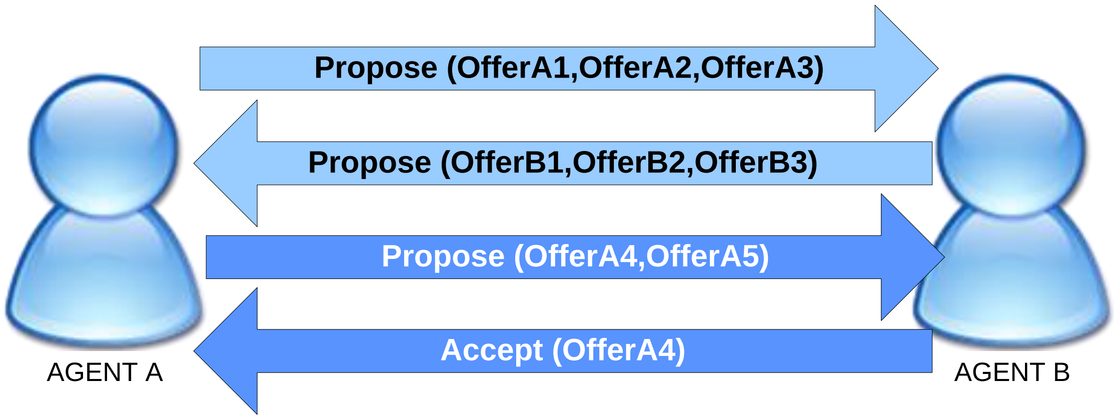

Figure 3.1: An example of two agents in the k-alternating protocol proposed by Lai et al. [28]

of the proposed offers satisfies the requirements of the opponent agent.

- Since k different offers are proposed in each agent's turn more information about opponent preferences can be inferred, increasing the chances of finding a *win-win* situation. This may produce faster agreements, which is inherently interesting for every domain but particularly for AmI domains since it may reduce the number of messages exchanged and thus the bandwidth consumption.

An example of two agents negotiating with a *3-alternating protocol* ($k = 3$) can be observed in Figure 3.1. Agent A is the initiator of the negotiation round, whereas Agent B is the responding agent. The first round starts with 3 offers proposed by the initiator. Once the offers arrive to Agent B, he decides whether he should accept one of them or not. Since the 3 offers are not interesting for Agent B, he decides to counteroffer 3 different offers. When the 3 first offers from Agent B arrive to Agent A, the second round starts. Due to the fact that none of the proposed offers by Agent B are interesting for the initiator, he decides to send 2 offers. The 2 offers from the initiator arrive to Agent B, whom analyzes the offers in order to determine if they are interesting. Since he found out that OfferA4 is interesting, he decides to accept it and thus the protocol ends with an agreement.

3.2 Negotiation Strategy

The proposed negotiation strategy can be classified into the group of strategies that use similarity heuristics to propose new offers to the opponent [21, 28]. The proposal complements some of the benefits introduced in the inspiring work of Lai et al. [28], making it specially interesting for AmI environments. The goal is to optimize the computational resources while maintaining a good performance in the negotiation process. The main traits of the proposed model are twofold. Firstly, it is not necessary to sample the entire utility function. Secondly, the proposed strategy provides an implicit learning mechanism that guides the offer sampling and which of the offers sampled are to be sent to the opponent.

The different decision-making mechanisms of the negotiation strategy can be grouped according to the period where they are applied: pre-negotiation and negotiation. The former group of decision making is applied before the negotiation process starts. Basically, since utility functions are complex and it is not feasible to completely explore them, each agent samples its own utility function by means of a niching GA (*self-sampling*).

The latter group of mechanisms is applied during the negotiation process. It includes the acceptance criteria for opponent offers, the concession strategy, the *evolutionary sampling*, and the selection of which offers are to be sent. The most remarkable part is introduced with the *evolutionary sampling*: genetic operators are carried out over received offers and one's own offers in order to sample new offers that may be of interest to both parties. The *evolutionary sampling* acts as an implicit learning mechanism of opponent's preferences. The result of the evolutionary sampling may be used afterwards when the offers to be sent to the opponent are selected. A brief outline of the proposed strategy can be observed in Algorithm 1. A more detailed outline of the strategy used before the negotiation process and during the negotiation process can be observed in Algorithms 2 and 3.

Algorithm 1 A brief outline of the negotiation strategy

Negotiation Strategy

Pre-negotiation

1. Self-sampling

Negotiation Process

2. Receive opponent offer(s) in case there is any offer

3. Acceptance criteria: Accept and offer and end the negotiation, or reject and continue the negotiation process

4. Concession strategy

5. Evolutionary sampling

6. Select which offers to send

7. Send offer(s) and go to step 2

3.2.1 Pre-negotiation: Self-sampling

When an agent uses complex utility functions to represent its preferences it may be difficult to find own offers with good utility. If the number of issues is not very large the complete sampling of the utility function may be feasible. However, when the number of issues is large, this complete sampling may be an extremely expensive process. For instance, a complete sampling of a negotiation domain formed by 10 integer issues from 0 to 9 requires sampling 10^{10} offers. The cost associated to this sampling can be exorbitant, especially if agent preferences change with a frequency that is greater than the time invested in the sampling. Furthermore, this sampling is unacceptable for AmI domains. Not only it takes too much computational time and power, but it also would need too much storage for the limited devices that are usually found in these domains. The sampling process can be reduced by skipping offers that are of very low quality for the agent (i.e., offers with utility equal to zero).

A possible solution to this problem is to use mechanisms that enable to sample good offers for the negotiation process and skip low quality ones. Due to the highly non-linear nature of complex utility functions, non-linear optimizers are required for this task. The main goal is to sample a set of different offers that have good utility and are significantly different because these offers may point to different regions of the negotiation space where a good deal may be found for the agent.

In this work, a genetic algorithm (GA) was used to solve this problem. GA's are general search and optimization mechanisms based on the darwinian selection process for species [51, 29]. Genetic operators such as crossover, mutation, and selection are employed in order to find near-optimal solutions for the required problem. Nevertheless, classic GA's pose the problem that the entire population converges to one optimal solution. As it has been stated, different interesting offers for the negotiation process need to be explored. Niching methods are introduced to confront problems of this kind [30, 52]. These methods look to converge to multiple, highly fit, and significantly different solutions.

A possible family of niching methods for GA's is the crowding approach [52]. Crowding methods achieve the desired result by introducing local competition among similar individuals. One advantage of crowding methods is that they do not require parameters beyond the classic GA's. Euclidean distance is usually used to assess the similarity among individuals. Probabilistic Crowding (P_C) and Deterministic Crowding (D_C) [52] are two of the most popular crowding methods. They only require a special selection rule with respect to classic GA's. Both rules are employed to select a winner given n different individuals. On the one hand, D_C selects the individual that has the highest fitness value, resulting in an elitist selection strategy. On the other hand, P_C allows lower fitness value individuals to be selected as winners with a certain probability. This probability is usually proportional to the fitness of each individual. P_C behaviour is more exploratory than D_C. In both cases, the niching effect is achieved by applying either of the two rules to those individuals that are similar. Each parent is usually paired with one of its children in such a way that the sum of the distances between pair elements is minimum. For each pair, one of the two crowding rules is employed to determine which individuals will form the next generation. D_C and P_C can be observed in more detail in Equations 3.1 and 3.2, respectively.

$$D_c(s_1, s_2) = \begin{cases} s_1 & f(s_1) > f(s_2) \\ s_2 & f(s_1) < f(s_2) \\ s_1 \vee s_2 & \text{other} \end{cases} \tag{3.1}$$

$$P_c(s_1, s_2) = \begin{cases} s_1 & f(s_1) > f(s_2) \ \wedge \ rand \leq p_1 \\ s_2 & f(s_1) > f(s_2) \ \wedge \ rand > p_2 \\ s_2 & f(s_1) < f(s_2) \ \wedge \ rand \leq p_2 \\ s_1 & f(s_2) < f(s_1) \ \wedge \ rand > p_1 \\ s_1 \vee s_2 & other \end{cases} \tag{3.2}$$

$$\text{with } p_i = \frac{f(s_i)}{f(s_i) + f(s_{i'})}$$

where $rand \in [0,1]$, $f(.)$ is the fitness function, s_1 and s_2 are two solutions, and p_1 and p_2 are the probability of acceptance of both solutions given the pair (s_1, s_2).

The designed mechanism uses a GA that employs crowding methods to find significantly different good offers. This GA is individually executed by the agent before the negotiation process begins. The chromosomes of this GA represent possible offers in the negotiation process, whereas the fitness function used is one's own utility function. A portfolio with D_C and P_C is used. The population has a fixed number of individuals and the whole population is selected to form part of the genetic operator pool. Pairs of parents are selected randomly and multi-point crossover or mutation operators are applied over them. In both cases, the result is two children. Each parent is paired with the child that is more similar to it according to euclidean distance. P_C or D_C is applied to each of the pairs according to a stablished probability p_{dc} and $1 - p_{dc}$ respectively. Those individuals that are selected as winners by the crowding replace the current generation. The stop criterion was set to a specific number of generations. At the end of the process, the whole population should have converged to different good offers that are to be used by the negotiation process as an approximation to the real utility function of the agent. This population, called P, is used as an input for the negotiation process. A more detailed outline of the proposed GA can be observed in Algorithm 2.

3.2.2 Negotiation: Concession strategy

A concession strategy determines which utility the agent will try to achieve at each negotiation step. The agent usually proposes offers that have a utility equal or above the utility level defined by its concession strategy at a specific negotiation round. In this work, we assume a time-dependent strategy, where the utility required by each agent depends on the remaining negotiation time. This kind of concession strategies are adequate for environments such as AmI, where time is a limitation (e.g., limited power devices, goods that loose their value as time passes, real-time environments, etc.). Some examples of concession strategies are *sit-*

Algorithm 2 Pre-negotiation: Genetic algorithm with niching mechanism. Its goal is to sample the agent utility function

```
P : Explored preferences, good quality offers    D_c : Deterministic crowding rule
P_c : Probabilistic crowding rule                p_cr : Probability of crossover operator
p_dc : Probability of DC                         n : Current number of generations
n_max : Maximum number of generations            pair_i : Pair of solutions

Initialize P
n = 0
Do
    n = n + 1
    shuffle P
    P_aux = ∅
    i = 1
    While i ≤ |P| - 1
        p_1 = P_i
        p_2 = P_{i+1}
        If Random() ≤ p_cr
            (c_1, c_2) = crossover(p_1, p_2)
        Else
            c_1 = mutate(p_1)
            c_2 = mutate(p_2)
        EndIf
        (pair_1, pair_2) = argmin_{p_i ≠ p_j, c_k ≠ c_l} ||p_i - c_k|| + ||p_j - c_l||
        If Random() ≤ p_dc
            Add(P_aux, D_c(pair_1))
            Add(P_aux, D_c(pair_2))
        Else
            Add(P_aux, P_c(pair_1))
            Add(P_aux, P_c(pair_2))
        EndIf
        i = i + 2
    EndWhile
    P = P_aux
While n ≤ n_max
Return P
```

and-wait [53] (no concession until the deadline, e.g. one of the agents has monopoly), linear (same concession rate at each step), boulware [20, 6] (no concession until the last rounds, where it quickly concedes to the reservation value), and conceder [20, 54] (at the start, it quickly concedes to the reservation value).

One of the traits of similarity-based strategies is that they are usually independent of the underlying concession strategy. However, this work assumes an environment where agents have similar market power (similar concession rate), and similar computational resources (similar deadlines). Thus, a linear concession strategy is assumed.

In each negotiation round, the agents concede according to their strategy until a private deadline is reached. The minimum utility that an agent a demands for a negotiation round t can be formalized as follows:

$$U_a(t) = 1 - (1 - RU_a)(\frac{t}{T_a}) \tag{3.3}$$

where $U_a(t)$ is the minimum demanded utility level for agent a at negotiation round t, RU_a is the reservation utility, and T_a is the private deadline of the agent.

3.2.3 Negotiation: Acceptance criteria

The acceptance criteria for an agent usually depends on its concession strategy. Normally, an opponent offer is accepted if it provides a utility that is equal or greater than the demanded utility for the next negotiation round. Consequently, given the set of offers $X^t_{b\to a}$ received by agent a from agent b at instant t, the acceptance criteria for agent a can be formalized as depicted in the following expression:

$$Accept^t_a(X^t_{b\to a}) = \begin{cases} \text{accept} & V_a(x^{t,best}_{b\to a}) \geq U_a(t+1) \\ \\ \text{reject} & otherwise \end{cases} \tag{3.4}$$

where $Accept^t_a(X^t_{b\to a})$ is the offer acceptance function, $V_a(x)$ valuates the utility of an offer, $x^{t,best}_{b\to a}$ is the best offer received from the opponent at negotiation round t, and $U_a(t+1)$ is the utility demanded for the next negotiation round.

3.2.4 Negotiation: Evolutionary sampling

One of the keys of the proposed strategy is the *evolutionary sampling*. It provides an implicit mechanism for learning opponent preferences and making an intelligent sampling. Basically, it is based in the application of some genetic operators to offers received from the opponent in the last negotiation round and one's own good offers from P. The idea behind the *evolutionary sampling* is that offers generated by this method have genetic material from the opponent and one's own agent. Therefore, these offers may yield a greater probability of being accepted by the opponent that offers that have been sampled in a blind way. The new offers are added to a special population called P_{evo} which contains offers that have been generated by genetic operators.

Let us consider $X^t_{b\to a} = [x^{t,1}_{b\to a}, x^{t,2}_{b\to a}, ..., x^{t,k}_{b\to a}]$, which is the set of offers sent by agent b to agent a at negotiation round t, and $U(t)$ the current desired utility to generate offers at negotiation round t. For each offer $x^{t,i}_{b\to a}$, a total of M offers are selected from the current iso-utility curve IC_P (offers with a utility equal to $U(t)$) defined in the population P. These M offers minimize the expression:

$$\underset{\substack{C \in IC_P \\ |C| = M}}{\text{argmin}} \sum_{j=1}^{M} ||x^{t,i}_{b\to a} - c_j|| \tag{3.5}$$

where C is the set of M different offers, and $||x^{t,i}_{b\to a} - c_j||$ is the euclidean distance between one of the offers in C and the offer received from the opponent. Thus, these M offers are the most similar ones to $x^{t,i}_{b\to a}$ from iso-utility curve in P and they will be involved in the evolutionary process. Offers are selected from the current iso-utility curve since offers with much greater utility may generate new offers with a utility that is no longer useful in the negotiation process (e.g. a utility greater than the current utility), and offers with lower utility may

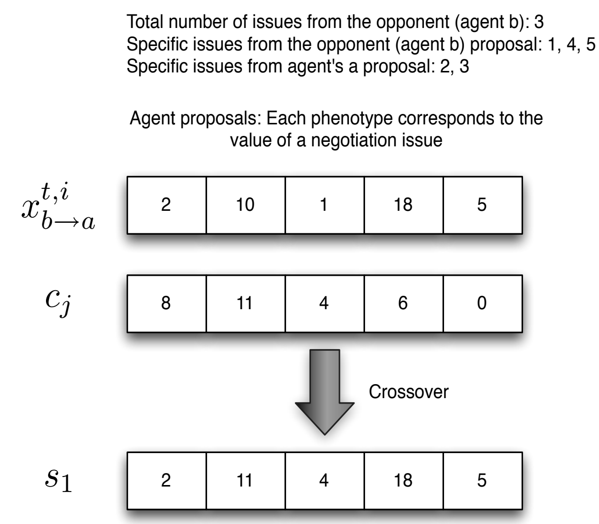

Figure 3.2: An example of a crossover operation

produce new offers that are not to be used until the last rounds of the negotiation process. Furthermore, the M selected offers are the most similar since applying crossover operators over offers that are too different may disrupt the quality of the solution for both agents (the resulting offer is too far from both agents' offers).

Once the M closest offers have been selected, a total of n_{cross} crossover operations are performed for each pair $(x^{t,i}_{b\to a}, c_j)$, where $c_j \in C$. The crossover operator takes two parents and generates one child. More specifically, the number of issues that come from $x^{t,i}_{b\to a}$ is chosen randomly from 1 and $N-1$, with N being the number of issues. The rest of the issues come from c_j. Which particular issues come from each parent is also decided randomly. This way, each agent's preferences are taken into account in an equal statistically manner. Each child is added to a special pool, called P_{evo}, that contains new offers sampled during the different *evolutionary sampling* phases. An example of a crossover operation can be observed in Figure 3.2.

A total of n_{mut} mutation operations are carried out for each generated child by crossover operations. The mutation operator changes issue values randomly, according to a certain probability of mutating individual issues (p_{attr}). When p_{attr} is low, mutated offers are close to the original offer, so the effect is the exploration of the neighbourhood of the offer. The operator is applied n_{mut} times to each child that is produced by crossover operations and to the original offers from the opponent. Mutation also generates new children that are added to the special pool P_{evo}

Note that no offer from P_{evo} is discarded even although their utility may be considered too low for the current negotiation round. The reason for this mechanism is that offers that are not currently acceptable may be interesting in future negotiation rounds due to the concession strategy. Furthermore, since they have genetic material from the opponent's offers, they are more likely to be accepted.

It can be observed in Algorithm 3 that if the negotiation process lasts n_{round} rounds, the Evolutionary Sampling will have explored a total number of offers that is equal to:

$$\begin{aligned} Samples_{evo} &= n_{round} * ((k * M * n_{cross}) + (k * M * n_{cross}) * n_{mut} + k * n_{mut}) \\ &= n_{round} * k * (M * n_{cross} * (1 + n_{mut}) + n_{mut}) \end{aligned}$$

Then, the number of offers sampled during the negotiation process depends on the number of rounds that the negotiation lasts, k, M, and the number of genetic operators that are performed per offer selected from the iso-utility curve.

3.2.5 Negotiation: Select which offers to send

The next step in specifying the negotiation strategy consists of defining the mechanism to propose new offers. In this case, it is necessary to devise a mechanism that is capable of proposing up to k different offers to the opponent and taking into account the preferences of the opponent. The applied heuristic takes into account the k offers received from the opponent and the offers in P and P_{evo}.

In order to select these offers, k offers from the current iso-utility curve are sent. More specifically, two different iso-utility curves are calculated. The first one is the iso-utility curve calculated using offers in P, called IC_P. The second one is the iso-utility curve calculated using offers in P_{evo}, called IC_E. Basically, the first iso-utility curve has offers that were generated during the *self-sampling* (only taking into account one's own preferences), whereas the second iso-utility curve only has offers that were generated in the *evolutionary sampling* (they may take into account both agents' preferences). The negotiation strategy defines a proportion of p_{pevo} offers to come from IC_E. The rest of the offers come from IC_P.

The offers selected from IC_E are those that minimize the distance to any offer received from the opponent in the previous negotiation round. This selection may be formalized as:

$$\underset{\substack{C \subset IC_E \\ |C| = p_{pevo} * k}}{\text{argmin}} \left(\sum_{j=1}^{C} \min_{x \in X^t_{b \to a}} ||c_j - x|| \right) \tag{3.6}$$

On the other hand, offers are also selected from IC_P. The total number of offers corresponds to a proportion that is equal to $1 - p_{pevo}$. In this case, offers that are the closest to any offer received from the opponent in the previous negotiation round are selected. This selection can be formalized as:

$$\underset{\substack{D \subset IC_P \\ |D| = (1 - p_{pevo}) * k}}{\text{argmin}} \left(\sum_{j=1}^{D} \min_{x \in X^t_{b \to a}} ||d_j - x|| \right) \tag{3.7}$$

The parameter p_{pevo} determines how relevant are the new offers sampled during the *evolutionary sampling* with respect to the offers sampled before the negotiation process. When $p_{pevo} = 0$, the strategy ignores the results that come from P_{evo}. Consequently, only offers that were sampled in the pre-negotiation phase (*self-sampling*) are sent to the opponent. In this particular case, the strategy is equivalent to a negotiation strategy that only samples before the negotiation process and does not take into account opponent's preferences. In contrast, when $p_{pevo} = 1$, the offers sampled during the *evolutionary sampling* are the only ones taken into account. In any case, p_{pevo} is a parameter to be adjusted.

Algorithm 3 Negotiation strategy during the negotiation process

P: Offers from *self-sampling* P_{new}: Offers from *evolutionary sampling*

k: Number of offers of the protocol M: Number of selected offers

n_{cross}: Number of times to crossover n_{mut}: Number of times to mutate

p_{pnew}: Proportion of offers from P_{new}

```
Receive $X^t_{b\to a}$

If $V_a(x^{t,best}_{b\to a}) \geq U_a(t+1)$ then Accept

Update current utility t=t+1
/*Evolutionary sampling*/
For each $x^{t,i}_{b\to a}$ in $X^t_{b\to a}$
    $C = \operatorname{argmin}_{C\subset IC_P, |C|=M} \sum_{j=1}^{M} ||x^{t,i}_{b\to a} - c_j||$

    For each $c_j$ in $C$
        Repeat $n_{cross}$ times
            s1=Crossover($x^{t,i}_{b\to a}, c_j$)
            If s1 $\nsubseteq P_{new}$ then Add($P_{new}$,s1)
            Repeat $n_{mut}$ times
                s2=Mutate(s1)
                If s2 $\nsubseteq P_{new}$ then Add($P_{new}$,s2)
            EndRepeat
        EndRepeat
    EndFor
    Repeat $n_{mut}$ times
        s1=Mutate($x^{t,i}_{b\to a}$)
        If s1 $\nsubseteq P_{new}$ then Add($P_{new}$,s1)
    EndRepeat
EndFor
/*Select which offers to send*/
$k_1 = p_{pnew} * k$
$X_1 = \operatorname{argmin}_{C\subset IC_E, |C|=k_1} \sum_{j=1}^{C} \min_{x \in X^t_{b\to a}} ||c_j - x||$

$k_2 = (1 - p_{pnew}) * k$
$X_2 = \operatorname{argmin}_{D\subset IC_P, |D|=k_2} \sum_{j=1}^{D} \min_{x \in X^t_{b\to a}} ||d_j - x||$

$X^{t+1}_{a\to b} = X_1 \cup X_2$
Send $X^{t+1}_{a\to b}$
```

3.3 Conclussions

This chapter has described the main traits of the proposed bilateral negotiation model for AmI environments. As it has been pointed out thorough the chapter, the main traits of the proposed model are: independence of the underlying complex utility function, aimed to achieve computationally and economically efficient solutions (accomplished by means of learning mechanisms provided by GA).

We have explained the employed negotiation protocol. It is adequate for situations where both agents are equal in power, and, since up to k different offers are sent, it is more adequate to explore opponent preferences and reach faster agreements. After that, the negotiation strategy has been described in depth. First, each agent samples its own complex utility function in order to obtain their own good offers. A Niching GA is used since it is able to obtain good offers that are significantly different. During the negotiation process, each agent performs genetic operators over own good offers and offers sent by the opponent. The aim of these operators is finding new offers which are interesting for both agents. Offers sent to the opponent are selected from the current iso-utility curve. In the next chapter the proposed model is tested in several scenarios to check its performance.

Chapter 4

Experiments

The performance of the devised strategy is detailed in this chapter. The proposed negotiation model was tested against the weighted constraint model proposed by Ito et al. [25]. This model allows to represent unrestricted interdependence relationships among the negotiation issues. Furthermore, if the number of constraints is large, it can represent highly non-linear utility functions. Therefore, it poses a proper testbed for the proposed strategy. Nevertheless, as the work of Lai et al. [28], the proposed negotiation model is general and does not depend on a particular utility function. The model of Ito et al. was selected as a testbed because it provides a well studied utility function [25, 26, 27] that holds enough complexity to study the real performance of the negotiation model.

First, the weighted constraint model is briefly introduced. After that, the negotiation setting employed in the experiments is briefly described. Following, the different experiments and their results are presented. Finally, a brief discussion that summarizes the results of the experiments is included.

4.1 Complex utility functions: Weighted constraint model

The weighted constraint model was first introduced by Ito et al.[25] as a complex utility function to model agent preferences. Let us consider a negotiation model where the number of issues is N, s_i represents the i-th issue, each issue has a domain $s_i \in [0, X]$ that sets its maximum and minimum value, and $\vec{s} = (s_1, s_2, ..., s_N)$ represents a particular offer. These settings conform a N-dimensional space for the utility function.

In the weighted constraint model, a constraint c_l represents a specific region of the space. Whatever point of the space enclosed in that region is said to satisfy the constraint c_l. Basically, the term *constraint* represents an interdependence relationship among the negotiation issues. Each constraint c_l has a certain value $v(c_l, \vec{s})$ that is added to the utility of $\vec{s}$ when the constraint is satisfied by the point $\vec{s}$. For instance, a constraint defined as $c_l = (1 \leq s_1 \leq 10 \wedge 3 \leq s_2 \leq 4)$ and $v(c_l, \vec{s}) = 10$ would hold a utility of 10 for the point (2,3) of the space.

A utility function in the weighted constraint model is formed by l constraints whose values are summed up whenever the constraints are satisfied. The utility of a point $\vec{s}$ given l constraints can be defined as:

$$U(\vec{s}) = \sum_{c_l \in L} v(c_l, \vec{s}) \tag{4.1}$$

where $\vec{s}$ is the point/offer, c_l is a constraint, L is the set of constraints, and $v(c_l, \vec{s})$ is the value of the constraint if it is satisfied (0 otherwise).

As it was stated in [25], although the expression seems linear, it produces a non-linear utility space due to the interdependence among the issues represented by the constraints. Furthermore, the utility function may generate spaces with several local maxima, which makes the problem highly non-linear and very difficult to optimize. Additionally, the agents do not have any knowledge about the possible constraints of the opponent, thus the problem of negotiation is still more difficult.

4.2 Negotiation settings

The aim of these experiments was to evaluate whether or not the proposed model is capable of working in domains where the agents' utility functions are highly non-linear. For that purpose, different negotiation cases where randomly created:

- Number of issues N = [4-7].
- Integer issues. $s_i \in [0, 9]$.
- $L = N{*}5$ uniformly distributed constraints per agent. There are constraints for every possible interdependence cardinality. For instance if N=4, there are 5 unary constraints, 5 binary constraints, 5 trinary constraints and 5 quaternary constraints.
- $v(c_l, .)$ for each n-ary constraint drawn randomly from $[0, 100 * n]$.
- For every constraint, the constraint width for each issue s_i is uniformly drawn from $[2, 4]$. For instance, if the constraint width for issue s_1 is 3, then $(0 \le s_1 \le 3)$, $(1 \le s_1 \le 4)$, $(2 \le s_1 \le 5)$, $(3 \le s_1 \le 6)$, $(4 \le s_1 \le 7)$, $(5 \le s_1 \le 8)$ and $(6 \le s_1 \le 9)$ are all of the possible configurations for issue s_1 in the constraint (just one is used in the constraint).
- Agent deadline $T = 10$. Agents do not know their opponent's private deadlines.
- Agent reservation utility $RU = 0$. Agents do not know their opponent's private reservation utilities.
- Agents do not know their opponent's utility functions

For each number of issues, a total of 100 negotiation cases were generated with the above settings. The execution of each case was repeated 30 times in order to take into account the possible differences between different executions of the methods.

In order to evaluate the quality of the agreements found by the participant agents, some measures were gathered at the end of each negotiation.

- Euclidean distance to the closest Pareto frontier point [55]. This is a measure of economic efficiency for agreements. The closer to the pareto frontier, the better.

- Euclidean distance to the Nash Product [55]. Since both agents have the same concession strategy and the same deadline it is also feasible to study the distance to the Nash Product. It is the point that maximizes the product $u_1 * u_2$ in the Pareto Frontier, where u_1 is the utility of agent 1, and u_2 is the utility of agent 2.

- Number of negotiation rounds. Faster agreements are preferred since a less number of messages are exchanged, less bandwidth is needed, and limited devices need less power to send messages.

4.3 Results

The proposed strategy, which will be named as *Evolutionary Sampling* or *ES*, was compared with two different negotiation models. The first strategy is an implementation of the general framework proposed by Lai et al. [28]. This model is provided with the whole sampling of the utility function, so that it can completely calculate iso-utility curves. It is used as a measure of how close the proposed strategy is to the ideal case where all of the offers are available. The second model assumes that it is not possible to completely sample all of the offers. Therefore, it samples before the negotiation process by means of a niching GA (*self-sampling*) and uses the similarity heuristic ($p_{pevo} = 0$) during the negotiation process, which will be named as *Non Evolutionary Sampling* or *NES* model. The number of samples explored before the negotiation process by the *NES* model is set equal to the number of samples explored by the ES model ($|P| + Samples_{evo}$). Consequently, both the *NES* and *ES* model yield the same computational cost in every experimentation.

Four different experiments were carried out in order to test the proposed model. In the first experiment, the three different models are compared as the number of issues is increased. In the second experiment, the impact of the proportion of offers (p_{pevo}) that are sent from the special pool P_{evo} in the *ES* model is studied. Following, the three models are compared as the number of proposals k increases. Finally, the *ES* and the *NES* model are compared as the size of the population ($|P|$) provided by the *self-sampling* increases .

4.3.1 Experiment 1: Performance study on the number of issues

The goal of this experiment is studying how the proposed strategy behaves for negotiations with different number of issues $N = \{4, 5, 6, 7\}$. It is important that the proposed model is capable of properly handle negotiations with multiple issues since most real world domains, including AmI domains, need to reach agreements for multiple issues. A negotiation setting where agents are limited to $k = 3$ proposals per negotiation round is used. The three different models were tested during this experiment.

The parameters of the *self-sampling* were set to $n_{max} = 100$, $p_{dc} = 80\%$ and $p_{cr} = 80\%$. The number of samples optimized before the negotiation process was set to $|P| = 128$ for the *ES* model and to $|P| = 128 + Samples_{evo}$ for the *NES* model.

The parameters of the *ES* were set to $M = 5$, $n_{cross} = 4$, $n_{mut} = 4$, $p_{attr} = 30\%$, and $p_{pevo} = 100\%$. Therefore, all the offers are sent from the samples generated by the *evolutionary sampling* carried out during the negotiation process.

The distance to the Nash Product, the distance to the closer Pareto Frontier Point, and the number of negotiation rounds were measured for the three models. The results for this experiment can be found in Figure 4.1. Intuitively, since the number of offers sampled remains constant and the number of issues increases, the performance of the *NES* and the *ES* model should be worsened with respect to the results achieved by the model of Lai et al. However, the results for the *ES* do not comply with this intuitive hypothesis. As it can be observed, even although the proposed model and the *NES* model explore the same number of offers, the *NES* obtains worse results than the other two models. This is particularly true as the number of issues increases since the performance of this method drastically decreases. On the contrary, the *ES* model is capable of achieving statistically equal results to the model of Lai et al., which can access the whole iso-utility curve. Nevertheless, the proposed model explores far fewer offers than the complete sampling of the utility function, specially for larger number of issues. For instance, when $N = 6$, Lai et al. has access to 10^6 offers, whereas the proposed model only has sampled an average of 1510 samples (128+ average $Samples_{evo}$).

The *ES* model has been able to achieve similar results to the case where the full iso-utility curve can be calculated, while maintaining the offers sampled in a small number. This result is particularly interesting for AmI domains where agents may be executed in devices with low computational and storage capabilities. Therefore, less samples mean less power consumption and less capacity needed to store them. Moreover, it must be also highlighted that the number of rounds was also lower than the one obtained by *NES*, consequently it means less number of messages sent, less bandwidth needed, and of course less power consumption by the limited devices.

The reason for this improvement is the intelligent sampling achieved by the use of genetic operators during the negotiation process. On the contrary, sampling only before the negotiation process leads to worse results since it is not capable detecting which offers will be interesting for the negotiation. Both, the *ES* and the *NES* model, have the same computational cost, but the *ES* is obviously preferred since it is capable of achieving a better performance.

4.3.2 Experiment 2: Performance study on p_{pevo}

In this case, the experiment's goal is to study how relevant is the proportion of offers that are sent from the offers sampled during the negotiation process (governed by the parameter p_{pevo}) in the *ES* model. Since all of the configurations sample new offers during the negotiation process, all of them yield a very similar computational cost. In fact, it may only be different if one of the configurations obtains a significantly different number of negotiation rounds. Consequently, the main subject of study in this scenario is the economic efficiency (distance

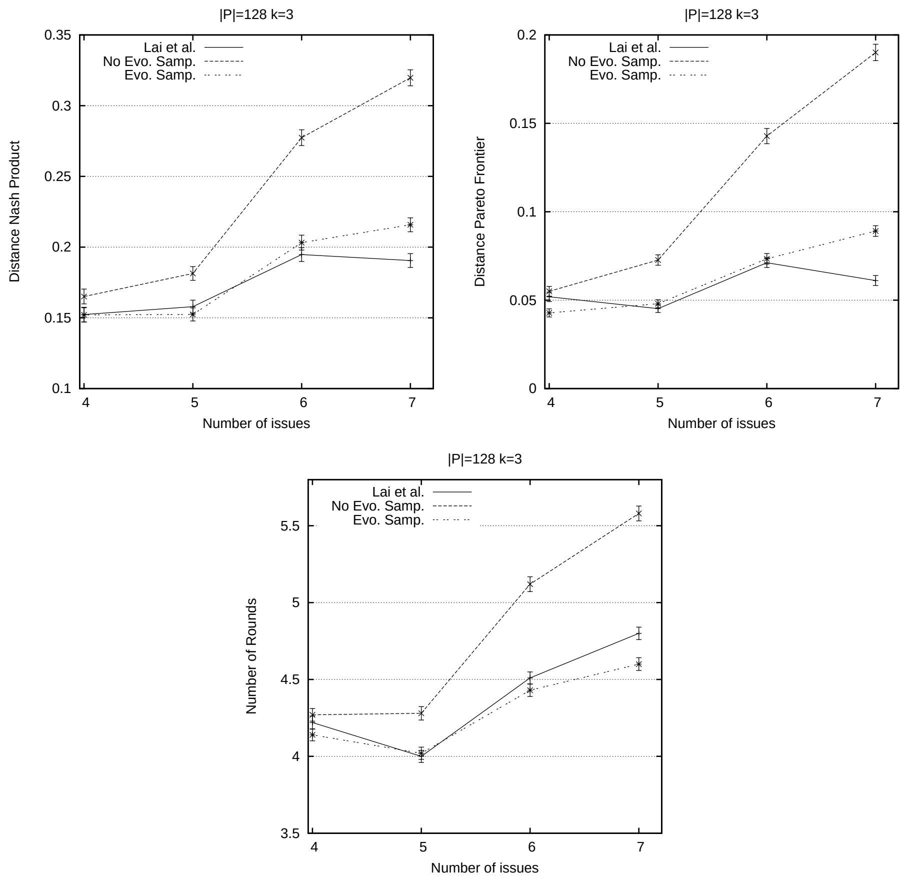

Figure 4.1: Evolution of the distance to the Nash Product, distance to the closer Pareto Point, and number of negotiation rounds in Experiment 1. The graphic shows the mean and its associated confidence intervals (95%)

to Nash and Pareto Frontier), although some improvements in the computational cost may be observed due to a lower number of rounds.

The same conditions from the previous experiment were set ($k = 3$ and $N = \{4, 5, 6, 7\}$), and the same configuration parameters were set for the *ES* ($M = 5$, $n_{cross} = 4$, $n_{mut} = 4$, and $p_{attr} = 30\%$). However, in this scenario we compare the *ES* model results when 1 out of 3 offers ($p_{pevo} = 30\%$), 2 out of 3 offers ($p_{pevo} = 50\%$), and 3 out of 3 offers ($p_{pevo} = 100\%$) come from the offers sampled during the *evolutionary sampling* phase.

The results for this second scenario can be observed in Figure 4.2. The graphic shows that the three different configurations yield similar results for the distance to the Nash Product, the distance to the closer Pareto Frontier Point, and the number of negotiation rounds. This similarity is explained due to the fact that, most of the times, the offer accepted by the opponent is the closest one from the *evolutionary sampling* population (P_{evo}). Therefore, it is always sent as long as the results from the *evolutionary sampling* are not ignored. Nevertheless, it seems that higher values of p_{pevo} have a slightly better economic and computational performance than lower ones. The reason for this slight improvement is that in some cases the offer preferred by the opponent may be the second or third closest from P_{evo}. Due to this small improvement, higher values of p_{pevo} are preferred in practice.

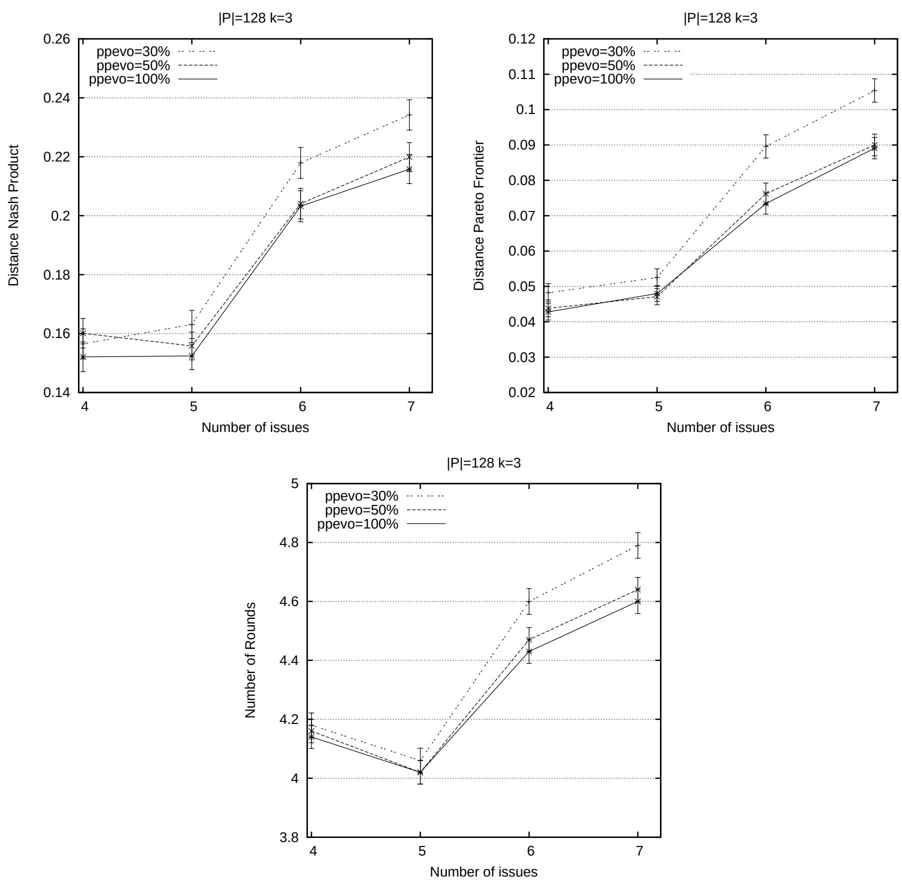

Figure 4.2: Evolution of the distance to the Nash Product, distance to the closer Pareto Point, and number of negotiation rounds in Experiment 2. The graphic shows the mean and its associated confidence intervals (95%)

4.3.3 Experiment 3: Performance study on k

The next experiment aims to study the performance of the three different models (Lai et al., *NES*, and *ES*) as the limit in the number of offers k sent per agent's round is increased. As it was mentioned, the number of offers sent may help to reach agreements faster since agents are capable of finding *win-win* situations. This is very important in AmI environments where devices have limited power and their running time must be optimized. In Lai et al. [28], it was shown how higher values of k helped to reach better agreements. In this scenario, the experiment is repeated in order to evaluate if the differences among the three models still hold for different values of k.

The studied values of k were 1, 3, 5, and 7. The rest of the negotiation setting was configured to use negotiation cases with $N = 6$ issues. The parameters of the *self-sampling* were set to the values employed in the previous tests except $|P| = 256$. The parameters of the *ES* were set to the same conditions described in Experiment 1.

As it can be observed in Figure 4.3, the three models achieve better results as k increases. This results agree with the ones presented in [28]. Although all of the models improve, the differences observed in Experiment 1 still hold for this scenario. The *NES* model gets worst results than Lai et al. and the proposed model. On the contrary, the *ES* obtains results that are statistically equivalent to the case when the full iso-utility curve can be calculated. As a matter of fact, for higher values of k the proposed model gets slightly better results than Lai et al. Nevertheless, the difference between the two of them are not significant enough to be considered as relevant.

It must be noted again that the number of offers sampled for *ES* and *NES* is the same and it is much lower than the complete sampling of the utility function. For instance, in this scenario, the complete sampling consists of 10^6 offers, whereas the other two methods sampled an average of 773 samples for $k = 1$, 1653 for $k = 3$, 2497 for $k = 5$, and 3357 for $k = 7$.

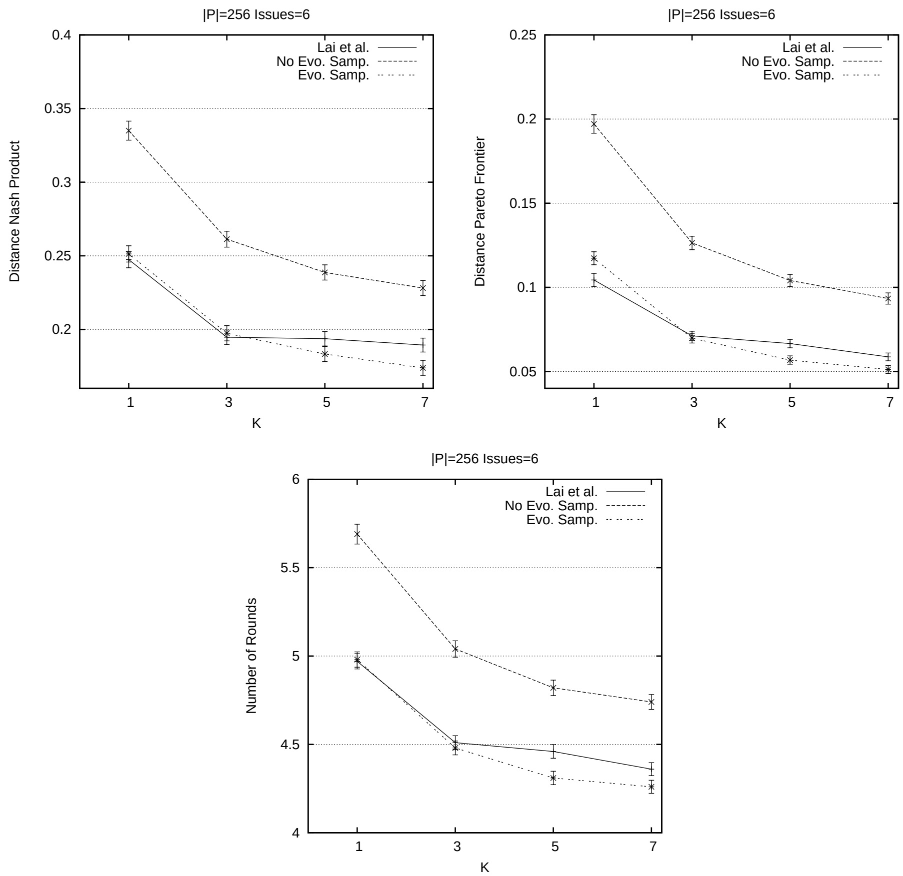

Figure 4.3: Evolution of the distance to the Nash Product, distance to the closer Pareto Point, and number of negotiation rounds in Experiment 3. The graphic shows the mean and its associated confidence intervals (95%)

4.3.4 Experiment 4: Performance study on $|P|$

This last experiment was designed to assess the influence of the population optimized by the *self-sampling* on the performance of the *ES* model and the *NES* model. It is specially relevant to see how many samples needs the *NES* model to achieve similar results to the ones obtained by the model proposed in this thesis. Obviously, more population means more storage needed and more computational cost since it needs to optimize more samples.

The average number of samples explored was analyzed for a negotiation setting where $N = 6$ and $k = 3$. The settings used for the *self-sampling* and the *ES* in previous experiments were repeated for this scenario. The number of sampled offers was increased by allowing more offers to be optimized in the *self-sampling* ($|P| = 128, 256, 512, 1024, 2048, 4096$).

The results for this experiment can be observed in Figure 4.4. The x axis of the graphics show the average number of offers sampled by both models, thus it shows $|P| + average_{rounds} * Samples_{evo}$. In the case of the *NES* model all of the samples were produced before the negotiation process started. Several observations can be made from the data shown in the graphics. On the one hand, it seems that the size of $|P|$ does not affect too much the performance of the *ES* model, since it is more dependent on the exploration carried out during the negotiation process and does not need as much sampling to get results similar to the case where the full iso-utility curve can be accessed. Therefore, the behaviour of the model remained almost constant for different configurations of $|P|$. Again, this behaviour is very adequate for AmI environments since the model can properly work with configurations that do not require too many computational resources. On the other hand, the *NES* model performance increased along with the number of offers sampled. It must be noted that when the number of samples for both methods was 5506, the two of them obtained very similar, almost equivalent, results. Therefore, the *NES* needed 5506 samples to achieve similar results to the same results obtained by the *ES* model for 1510 samples. It can be concluded that *NES* needs $\frac{5506}{1510} = 3.64$ times more samples to achieve similar results to *ES*.

4.4 Discussion

Ambient Intelligence domains are characterized as domains where computational resources are of extreme importance. Users interact with its environment through devices with limited capabilities, thus the efficient use of resources is crucial. Furthermore, the environment infrastructures are usually connected by means of a limited bandwidth wireless connection. Thus, network resources must also be optimized.

The results obtained by the proposed model, while maintaining fairly good economic performance, cope with the problems found in AmI environments. If we assume that limited

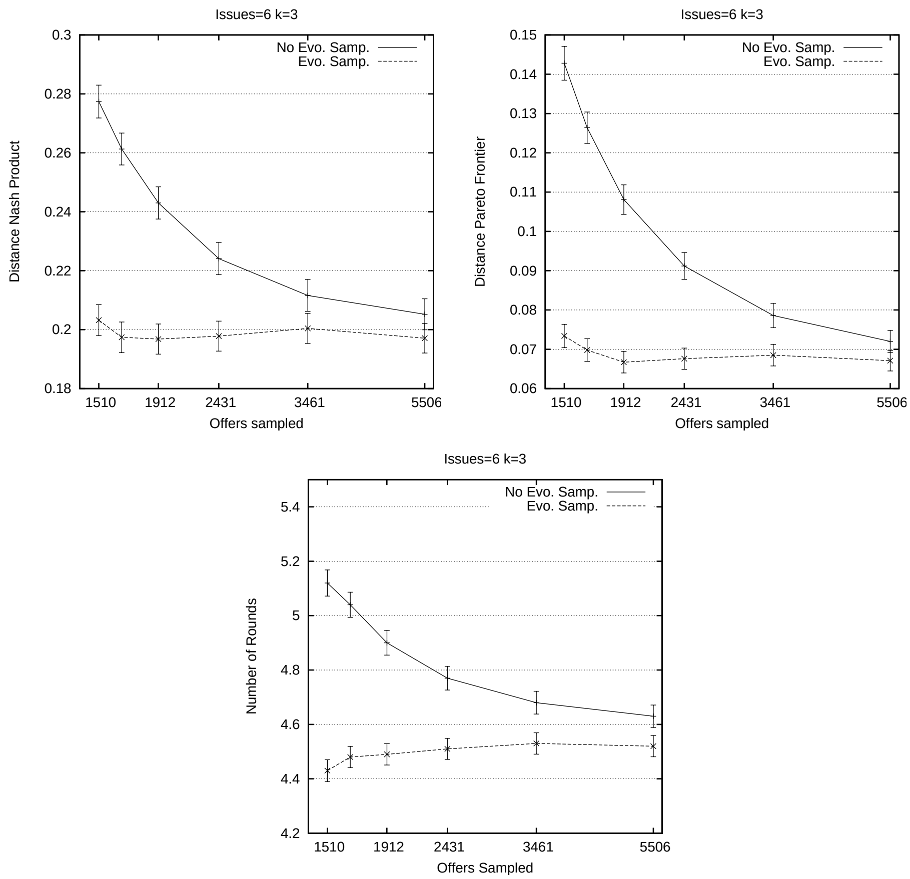

Figure 4.4: Evolution of the distance to the Nash Product, distance to the closer Pareto Point, and number of negotiation rounds in Experiment 4. The graphic shows the mean and its associated confidence intervals (95%)

devices cannot completely sample agent's utility function and store those samples, some mechanisms are needed to samples as few offers as possible. A straightforward method would be sampling some offers before the negotiation process, which is precisely what the *NES* model does. However, this sampling does not take advantage of the information revealed by the opponent in the negotiation process. Most of the offers sampled before the negotiation process may be useless since they are of no interest for the opponent. However, the proposed model takes advantage of this information and employs it to make a more intelligent sample, optimizing the computational resources. Nevertheless, although computational resources are important, economic efficiency should not be ignored in AmI negotiations.

In the previous sections, we could observe the behaviour of the *ES* model in different scenarios. Its performance was compared with a method that samples the same number of offers before the negotiation process (*NES*), and the ideal case where all of the samples of the utility function are available. The results of the experiments can be summarized as:

- The proposed model needs very few computational resources and storage to obtain statistically equivalent results to the ideal case where the all of the offers are available. It obtained similar results in economic performance (distance to Nash, distance to Pareto Frontier) and number of negotiation rounds.

- When the proposed model and the *NES* model sample the same number of offers, the first obtains better results. In fact, the *NES* model needs to sample 3.64 times more offers to obtain similar results.

- The proposed model needs less negotiation rounds to achieve better results than the *NES* model. Therefore, the environment bandwidth is optimized since it needs less messages to be sent in order to reach agreements.

Consequently, the proposed model fits perfectly the conditions needed by AmI environments since it needs less computational resources and it obtains economically efficient results.

Part IV

Conclusions

Discussion

Ambient Intelligence looks to offer personalized services and provide users with easier and more efficient ways to communicate and interact with other people and systems [1, 2]. Since several users may coexist in AmI environments, it is quite probable that their preferences conflict and thus mechanisms are needed to allow users to cooperate. For instance, imagine an ubiquitous shopping mall [18, 19] where buying agents have to help users to buy their needed products and vendor agents have to maximize their users' profits. Automated negotiation provides mechanisms that solve this particularly interesting problem. Some authors have already claimed that in most real world negotiations such as e-commerce [23, 24, 25], issues present interdependence relationships that make agents' utility functions complex. Therefore, the problem of complex utility functions in automated negotiation yields also interest for AmI applications.

A multi-issue bilateral bargaining model for Ambient Intelligence domains that deals with complex utility functions has been presented in this thesis. This work complements the inspiring work of Lai et al. [28] and provides a negotiation model that is proper for Ambient Intelligence applications. The main goal of this work is to achieve efficient agreements while maintaining the use of computational resources low.

The proposed model uses a negotiation protocol where agents are allowed to send up to k different offers in each negotiation round. Before the negotiation process starts, each agent samples its own utility function by means of a niching genetic algorithm. This genetic algorithm gets highly interesting and significantly different offers for one's own utility function (*self-sampling*). After the negotiation process starts, the agents apply genetic operators over the last offers received from the opponent and those offers that are most similar from the current iso-utility curve (*evolutionary sampling*). The desired effect is to sample new offers that are interesting for both parties. Therefore, the opponent preferences guide the sampling process during the negotiation process. The offers that are sent to the opponent are selected from the current iso-utility curve, those that are the most similar to the last offers received from the opponent. An additional mechanism is introduced that allows to give priority to those offers that come from the *evolutionary sampling* iso-utility curve.

Several experimental scenarios have been carried out and studied. In these tests, the proposed model has been compared with a similarity heuristic that has access to all of the possible offers and a similarity heuristic that samples the same number of offers before the negotiation process by means of a niching genetic algorithm (*NES*). The results show that the proposed model needs very few computational resources and storage to obtain statistically equivalent results to the ideal case where all of the offers are available. For instance, the full iso-utility curve consists of 10^6 offers and the proposed model just samples 1510 offers in a negotiation setting where the number of issues is 6, and the number of offers sent per negotiation round is 3. Additionally, although the proposed model and the *NES* model sample the same number of offers, the first one obtains better results. In fact, the *NES* model needs to sample 3.64 times more offers to obtain similar results. The low computational cost and the efficient results make the proposed model very adequate for Ambient Intelligence domains. Next, we will discuss how our proposal relates to similar works in the area.

Faratin et al. [20] presented a negotiation model for linear utility functions where a negotiation strategy is composed of different tactics that may be applied depending on the negotiation time, the quantity of the resource and the behaviour of the opponent. Nevertheless, the model is only applicable in negotiation with linear utility functions, which are easier cases than the ones presented in this present thesis.

Matos et al. [45] determined the successful strategies for different settings using the model proposed by Faratin et al. [20]. They employ an evolutionary approach in which strategies and tactics correspond to the genetic material in a genetic algorithm. In their experiments, populations of buyers and sellers with different strategies negotiate in a round robin way. After each round robin round, strategies are evaluated by means of a fitness function. Then, strategies are selected to be the parents of the next population according to their fitness function. In the end, a population of strategies implicitly adapted to the environment is obtained. They use genetic algorithms as a learning mechanism of negotiation strategies when placed under certain circumstances. There are two differences between Matos et al. work and the present work. Firstly, the negotiation model of Matos et al. is designed for linear utility functions. Secondly, the genetic algorithm proposed in this present work is an implicit learning mechanism of the opponent's preferences that guides the offer sampling during the negotiation process.

Later, Faratin et al. [21] presented a negotiation strategy for bilateral bargaining that is focused on achieving *win-win* situations by means of trade-off. The heuristic applied to perform trade-off is similar to the one employed in this present work. Given an agent's current utility, the offer from the iso-utility curve that is most similar to the last offer received from the opponent is sent. The idea behind this heuristic is that, since the proposed offer is the most similar to the last offer received from the opponent, it is more likely to be satisfactory for both participants. A fuzzy similarity criteria is employed to compare offers. Nevertheless, the use of fuzzy similarity requires some knowledge of opponent preferences. The application of criteria of this kind is complicated in complex utility functions due to the inter-dependencies among the different issues. In this present work, the euclidean distance is used, which does not require any knowledge about the opponent and which is independent of the inter-dependencies among issues.

Fatima et al. [56, 22, 57] analyzed the problem of multi-attribute negotiations in an agenda-based framework. Agendas determine in which order the different issues are to be negotiated when negotiations are carried out issue by issue. Once an agreement has been found on a specific issue, it cannot be changed. Thus, the agents face the problem of which issues should be negotiated first and which strategies should be applied. They studied the optimal agendas for different scenarios. Nevertheless, their work focused on linear utility functions, which does not take into account the possible interdependences among the different issues.

The work of Krovi et al. [48] opened the path for GA's in automated negotiation. Krovi et al. proposed a GA for bilateral negotiations that was performed each time a negotiation round ended. The population of chromosomes was randomly initialized with 90 random offers and 10 heuristic offers (the last offer from the opponent and the nine best offers from the previous round). The idea behind using GA's is that resulting the offers have good characteristics for both agents. However, 60 generations were needed each round in order to obtain the next offer, which may result computationally expensive in large issue domains. Choi et al. [49] enhanced Krovi's model with more learning capabilities. More specifically, it is capable of learning opponent preferences by means of stochastic approximation and adapt its mutation rate to the opponent behaviour. However, these strategies and mechanisms are devised for linear utility functions with few negotiation issues. The performance of these methods is uncertain when there is a large number of issues or complex utility functions are used. This present work also employs genetic operators to obtain new offers, but it is capable of providing solutions for domains with complex utility functions and domains where the number of issues is large.

There have been some works that have studied the problem of negotiation models for complex utility functions. Most of them have focused on negotiation models that are mediated. The seminal work of Klein et al. [23] proposes a mediated negotiation model where agents have their preferences represented by influence matrices. Influence matrices represent binary interdependence relationships between binary issues. Their proposed approach consists in a mediator that generates bids that are voted by the agents participant in the protocol. Ito et al. [25] proposes a mediated negotiation model for multilateral negotiations where agents have their preferences represented by weighted constraints. The agents sample their utility function and carries out a simulated annealing for each point sampled in order to obtain one's own bids. If the utility of such point is above a certain threshold, the constraints that the bid satisfies are sent to the mediator (constraint bid). After receiving bids from the agents, the mediator tries to look for contracts common to the bids received, while maximizing social welfare. Marsa-Maestre et al. [26, 27] further research in the area of mediated negotiation models for complex utility functions. More specifically, they take advantage of the constraint based model by proposing different bidding mechanisms that work in the constraint space instead of the bid space. They also allow for a negotiation protocol that may not be *one-shot*. In fact, the mediator can suggest the relaxation of some constraint bids in order to increase the probability of finding an agreement. Nevertheless, all of these works need a trusted mediator, which may not be available in every domain. Furthermore, their models are highly dependent on the underlying utility function. This present work does not require a mediator and the model is independent of the underlying utility function.

Robu et al. [24, 34] presented a non-mediated bilateral negotiation strategy for agents in electronic commerce. Agent utility functions are based on special graphical models called utility graphs. One of the agents, the seller, is responsible for finding agreements that are satisfactory for both parties. In order to do that, the seller models the buyer by means of utility graphs and tries to learn the buyer's preferences. However, utility graphs are only designed for binary issues. Our work differs in that it is capable of working with general complex utility functions and is also capable of working issue domains that are not necessarily binary.

In Lai et al. [28], a powerful bilateral bargaining model with general utility functions is presented. The negotiation protocol is based on the Rubinstein alternating protocol [50], but each agent is allowed to send up to k different offers in each round. The offer with highest utility is chosen from the k offers received from the opponent in the last round. The offer from the current iso-utility curve that is the most similar to the one chosen by the agent from the offers made by the opponent is selected. This offer from the iso-utility curve becomes a seed from which *k-1* offers in the neighbourhood are generated. The selected offer from the curve and the *k-1* generated offers are sent back to the opponent. Again, the general ideal behind this heuristic is that since the offers are similar to one of the last offers received from the opponent, they are more likely to be satisfactory for both parties. The model proposed in this thesis complements the seminal work of Lai et al. since it adapts similarity models for AmI environments. In the model proposed in this thesis, only a small number of offers are sampled before the negotiation process since it is assumed that the utility function cannot be exhaustively explored. This is specially important for scenarios with a large number of issues and scenarios where devices have limited storage and computational resources. Secondly, the proposed model incorporates an implicit learning mechanism that allows, thanks to genetic operators, an intelligent sampling of new offers that may be of interest for both parties.

The proposed model has been capable of achieving good economical results despite the fact that it uses less computational resources. Therefore, it solves part of the problem regarding automated negotiation in AmI environments. Next, we describe our future work in the line of automated negotiation.

Future Work

In a vast amount of literature, multi-agent systems have been appointed as an appropriate paradigm for complex distributed systems. In the last few years, the community has focused its efforts on resolving conflicts and coordination problems that may arise among agents. Agreement technologies consist of the mechanisms that solve these conflict and coordination problems. Negotiation is addressed as one of these core technologies. Despite the fact that much work has been done in this area, there are still some complex problems that may need new agreement mechanisms. In the next paragraphs we introduce a scenario that represents a type of problems that may need new and as yet unstudied agent-based agreement mechanisms.

Imagine you and your friends have decided to go on a trip together. Now you have to arrange all of the trip details: the city you plan to visit, the number of days the group will stay, accommodation, a flight that will take you and your friends to the city, and the amount of money that the group will spend. Even though the wish of each friend is to go on the trip together, each one may have different preferences and opinions about aspects of the trip: different preferences about the cities, amount of money to be spent, the quality of the hotel and so forth. A personal agent, an agent that acts on behalf of someone according to their preferences, can perform the task of searching for and negotiating with multiple travel agencies in order to get an optimal deal for its user. However, a personal agent only knows about its user's preferences. Ideally, the final deal with a travel agency should take into account the preferences of every member of the group. Therefore, some sort of coordination mechanism among the personal agents of each friend is required.

One could think that the conditions found in the *travelling friends problem* are very specific and rare. However, this view is far from reality. Similar problems may arise in other complex domains where groups of agents may need to negotiate with opponents while solving their inner conflicts. For instance, some of these scenarios are: Agricultural cooperatives markets, Virtual Organizations, and Labor Union negotiations.

The previous examples illustrate that such conditions are also found in very different domain scenarios. The proposed scenarios are inherently complex. Not only may they require very complex mechanisms, but they also represent crucially important domains for our society. Agreement technologies may be employed in these scenarios to study how computationally provided solutions may help people in such delicate domains. For this reason, one of our goals is to study the mechanisms that allow agent systems to provide solutions to these complex problems.

Whether we recall the three examples above or the *travelling friends problem* (we will use this last example from this point on for the sake of simplicity), all of them had a common setting. Although all of the friends (farmers, employees, etc.) had the same joint-goal (to go on a trip together), each one had their own preferences or sub-goals regarding the different options available. Teams are appointed in the agent literature as the paradigm to resolve problems where groups of agents share a joint intention. The group of friends can be seen as a team whose goal is to negotiate a trip together with a travel agency. Obviously, the same comparison can be made with the group of personal agents: they are an agent team, and more specifically an agent-based *negotiation team*. In these kinds of situations, the *negotiation team* has to deal with its own internal conflicts and the conflicts generated during the negotiation with opponents.

Even though teamwork has been actively studied in multi-agent research, not much attention has been paid to *negotiation teams* from the point of view of agent research. Therefore, our research goal consists in providing mechanisms for negotiations that teams carry out with one or several opponents. For that purpose, we will try to employ the knowledge acquired during this Msc. thesis.

Related Publications

- V.Sánchez-Anguix, S. Valero, V. Julian, V. Botti and A. García-Fornes. *Genetic-Aided Multi-Issue Bilateral Bargaining for Complex Utility Functions.* The Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010) pp. In Press. (2010). **CORE A+**

- V.Sánchez-Anguix, A. Espinosa, L. Hernández and A. García-Fornes. *MAMSY: A management tool for multi-agent systems.* 7th International Conference on Practical Applications of Agents and Multi-Agent Systems Vol. 55 pp. 130-139. (2009). **Computer Science Conference Ranking EIC=0.56**

- V.Sánchez-Anguix, J. A. García-Pardo, A. García-Fornes and V. Julian. *Towards soccer simulation as a testbed for adaptive systems and agreement technologies.* 8th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2010) Vol. 71, pp. 19-27, (2010). **Computer Science Conference Ranking EIC=0.56**

- V.Sánchez-Anguix, S. Esparcia, E. Argente, A. García-Fornes and V. Julian. *Collaborative information extraction for adaptive recommendations in a multiagent tourism recommender system.* 8th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2010) Vol. 70 pp. 35-40. (2010). **Computer Science Conference Ranking EIC=0.56**

- V.Sánchez-Anguix, S. Valero and A. García-Fornes. *Tackling Trust Issues in Virtual*

Organization Load Balancing. The Twenty Third International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA-AIE 2010), Vol. 6098, pp. 586-595, (2010). **Computer Science Conference Ranking EIC=0.57**

- S. Esparcia, V.Sánchez-Anguix, E. Argente, A. García-Fornes and V. Julian. *Integrating information extraction agents into a tourism recommender system.* 5th International Conference on Hybrid Artificial Intelligence Systems (HAIS 2010) pp. In Press. (2010). **CORE C**

- V.Sánchez-Anguix, V. Julian, V. Botti and A. García-Fornes. *Towards agent-based negotiation teams.* Working Conference on Human Factors and Computational Models for Negotiation at Group Decision and Negotiation (HuCom@GDN2010) pp. In Press. (2010). **CORE B**

Acknowledgments

This work is supported by TIN2008-04446, TIN2009-13839-C03-01, CSD2007-00022 and FPU grant AP2008-00600 of the Spanish government, and PROMETEO 2008/051 granted by Generalitat Valenciana.

I would also like to thank Ana, Vicente, Vicent, and Soledad for their advice and help during the process of this Msc. Thesis. Thank you.

Bibliography

[1] Juan M. Corchado, Javier Bajo, Yanira de Paz, and Dante I. Tapia, "Intelligent environment for monitoring alzheimer patients, agent technology for health care," *Decis. Support Syst.*, vol. 44, no. 2, pp. 382–396, 2008.

[2] W. Weber, J. M. Rabaey, and E. Aarts, *Ambient Intelligence*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[3] Juan A. Fraile, Javier Bajo, Ajith Abraham, and Juan M. Corchado, "Hocama: Home care hybrid multiagent architecture," in *Pervasive Computing*, Computer Communications and Networks, pp. 259–285. 2009.

[4] P. Remagnino and G.L. Foresti, "Ambient intelligence: A new multidisciplinary paradigm," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, no. 1, pp. 1–6, Jan. 2005.

[5] Michael J. Wooldridge, *Multi-agent systems : an introduction*, Wiley, 2001.

[6] Dean G. Pruitt, *Negotiation Behavior*, Academic Press, 1981.

[7] J.F. Nash, "The bargaining problem," *Econometrica*, vol. 18, pp. 155–162, 1950.

[8] A. Rubinstein, "Perfect equilibrium in a bargaining model," *Econometrica*, vol. 50, pp. 155–162, 1982.

[9] K. Binmore, A. Rubinstein, and A. Wolinsky, "The nash bargaining solution in economic modelling," *Rand Journal of Economics*, vol. 17, pp. 176–188, 1986.

[10] Sarit Kraus, "Negotiation and cooperation in multi-agent environments," *Artif. Intell.*, vol. 94, no. 1-2, pp. 79–97, 1997.

[11] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, M. J. Wooldridge, and C. Sierra, "Automated negotiation: Prospects, methods and challenges," *Group Decision and Negotiation*, vol. 10, pp. 199–215, Mar 2001.

[12] R.G. Vedder and R.O Mason, "An expert system application for decision support in law enforcement," *Decision Sciences*, 1987.

[13] Katia P. Sycara, ," *Negotiation Planning: An AI Approach*, vol. 46, no. 1, pp. 216–234, May 1990.

[14] Katia P. Sycara, "Problem restructuring in negotiation," *Manage. Sci.*, vol. 37, no. 10, pp. 1248–1268, 1991.

[15] Robert H. Guttman, Alexandros G. Moukas, and Pattie Maes, "Agent-mediated electronic commerce: a survey," *Knowl. Eng. Rev.*, vol. 13, no. 2, pp. 147–159, 1998.

[16] Alessio Lomuscio, Michael Wooldridge, and Nicholas R. Jennings, "A classification scheme for negotiation in electronic commerce," in *Agent Mediated Electronic Commerce, The European AgentLink Perspective.*, London, UK, 2001, pp. 19–33, Springer-Verlag.

[17] Catholijn Jonker and Valentin Robu, "Automated multi-attribute negotiation with efficient use of incomplete preference information," in *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, Washington, DC, USA, 2004, pp. 1054–1061, IEEE Computer Society.

[18] S. Keegan, G.M.P O'Hare, and M.J. O'Grady, "Easishop: Ambient intelligence assists everyday shopping," *Information Sciences*, vol. 178, no. 3, pp. 588–611, 2008.

[19] Javier Bajo, Juan M. Corchado, Yanira de Paz, Juan Francisco de Paz, Sara Rodríguez, Quintín Martin, and Ajith Abraham, "Shomas: Intelligent guidance and suggestions in shopping centres," *Appl. Soft Comput.*, vol. 9, no. 2, pp. 851–862, 2009.

[20] Peyman Faratin, Carles Sierra, and Nicholas R. Jennings, "Negotiation decision functions for autonomous agents," 1998, number 3-4, pp. 159–182.

[21] P. Faratin, C. Sierra, and N. R. Jennings, "Using similarity criteria to make negotiation trade-offs," in *4th International Conference on Multi-Agent Systems (ICMAS-2000)*, 2000, pp. 119–126.

[22] Shaheen S. Fatima, Michael Wooldridge, and Nicholas R. Jennings, "An agenda-based framework for multi-issue negotiation," *Artif. Intell.*, vol. 152, no. 1, pp. 1–45, 2004.

[23] Mark Klein, Peyman Faratin, Hiroki Sayama, and Yaneer Bar-Yam, "Negotiating complex contracts," in *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, New York, NY, USA, 2002, pp. 753–757, ACM.

[24] Valentin Robu, D. J. A. Somefun, and J. A. La Poutré, "Modeling complex multi-issue negotiations using utility graphs," in *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, New York, NY, USA, 2005, pp. 280–287, ACM.

[25] Takayuki Ito, Mark Klein, and Hiromitsu Hattori, "A multi-issue negotiation protocol among agents with nonlinear utility functions," *Multiagent Grid Syst.*, vol. 4, no. 1, pp. 67–83, 2008.

[26] Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Juan R. Velasco, and Enrique de la Hoz, "Effective bidding and deal identification for negotiations in highly nonlinear scenarios," in *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, Richland, SC, 2009, pp. 1057–1064, International Foundation for Autonomous Agents and Multiagent Systems.

[27] Ivan Marsa-Maestre, Miguel A. López-Carmona, Juan R. Velasco, Takayuki Ito, Mark Klein, and Katsuhide Fujita, "Balancing utility and deal probability for auction-based negotiations in highly nonlinear utility spaces," in *IJCAI*, 2009, pp. 214–219.

[28] Guoming Lai, Katia Sycara, and Cuihong Li, "A decentralized model for automated multi-attribute negotiations with incomplete information and general utility functions," *Multiagent Grid Syst.*, vol. 4, no. 1, pp. 45–65, 2008.

[29] J. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, 1975.

[30] Samir W. Mahfoud, *Niching methods for genetic algorithms*, Ph.D. thesis, Champaign, IL, USA, 1995.

[31] Jyrki Wallenius, James S. Dyer, Peter C. Fishburn, Ralph E. Steuer, Stanley Zionts, and Kalyanmoy Deb, "Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead," *Manage. Sci.*, vol. 54, no. 7, pp. 1336–1349, 2008.

[32] Ralph L. Keeney and Howard Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, Cambridge University Press, 1993.

[33] Mark Klein, Peyman Faratin, Hiroki Sayama, and Yaneer Bar-Yam, "Protocols for negotiating complex contracts," *IEEE Intelligent Systems*, vol. 18, no. 6, pp. 32–38, 2003.

[34] Valentin Robu and J. A. La Poutré, "Retrieving the structure of utility graphs used in multi-item negotiation through collaborative filtering of aggregate buyer preferences," in *Proceedings of the 2nd International Workshop on Rational, Robust and Secure Negotiations in Multi-Agent Systems (RSS'06). Hakodate, Japan.*

[35] Mark Klein, Peyman Faratin, Hiroki Sayama, and Yaneer Bar-Yam, "Negotiating complex contracts," *Group Decision and Negotiation*, vol. 12, pp. 111–125, March 2003.

[36] Takayuki Ito, Mark Klein, and Hiromitsu Hattori, "An auction-based negotiation protocol for agents with nonlinear utility functions," in *Transactions of The Institute of Electronics, Information and Communication Engineers (IEICE)*, 2006, vol. J89-D, pp. 2648–2660.

[37] Hiromitsu Hattori, Mark Klein, and Takayuki Ito, "Using iterative narrowing to enable multi-party negotiations with multiple interdependent issues," in *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, New York, NY, USA, 2007, pp. 1–3, ACM.

[38] Guoming Lai, Katia Sycara, and Cuihong Li, chapter A General Model for Pareto Optimal Multi-Attribute Negotiations, pp. 59–80, Rational, Robust, and Secure Negotiations in Multi-Agent Systems. 2008.

[39] Catholijn M. Jonker and Jan Treur, "An agent architecture for multi-attribute negotiation," in *Proceedings of the 17th International Joint Conference on AI, IJCAI'01, 2001*. 2001, pp. 1195–1201, Morgan Kaufman.

[40] P. Faratin, C. Sierra, and N. R. Jennings, "Using similarity criteria to make issue trade-offs in automated negotiations," *Artificial Intelligence*, vol. 142, pp. 205–237, 2002.

[41] Dajun Zeng and Katia Sycara, "Bayesian learning in negotiation," *Int. J. Hum.-Comput. Stud.*, vol. 48, no. 1, pp. 125–141, 1998.

[42] Vidya Narayanan and Nicholas R. Jennings, "Learning to negotiate optimally in non-stationary environments," in *Cooperative Information Agents X, 10th International Workshop, CIA 2006, Edinburgh, UK, September 11-13, 2006, Proceedings*. 2006, vol. 4149 of *Lecture Notes in Computer Science*, pp. 288–300, Springer.

[43] Koen Hindriks and Dmytro Tykhonov, "Opponent modelling in automated multi-issue negotiation using bayesian learning," in *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, Richland, SC,

2008, pp. 331–338, International Foundation for Autonomous Agents and Multiagent Systems.

[44] Jim R. Oliver, "On artificial agents for negotiation in electronic commerce," in *HICSS '96: Proceedings of the 29th Hawaii International Conference on System Sciences Volume 4: Organizational Systems and Technology*, Washington, DC, USA, 1996, p. 337, IEEE Computer Society.

[45] N. Matos, C. Sierra, and N. Jennings, "Determining successful negotiation strategies: An evolutionary approach," in *ICMAS '98: Proceedings of the 3rd International Conference on Multi Agent Systems*, Washington, DC, USA, 1998, p. 182, IEEE Computer Society.

[46] Tuan Tu, Eberhard Wolff, and Winfried Lamersdorf, "Genetic algorithms for automated negotiations: A fsm-based application approach," in *Proceedings of the 11th International Conference on Database and Expert Systems (DEXA 2000)*, A. Al-Zobaidie A. M. Tjoa, R. R. Wagner, Ed. 9 2000, p. 1029ff., IEEE.

[47] Enrico Gerding, David Van Bragt, and Han La Poutré, "Multi-issue negotiation processes by evolutionary simulation, validationand social extensions," *Comput. Econ.*, vol. 22, no. 1, pp. 39–63, 2003.

[48] Ravindra Krovi, Arthur C. Graesser, and W. E. Pracht, "Agent behaviors in virtual negotiation environments," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 29, no. 1, pp. 15–25, 1999.

[49] Samuel P. M. Choi, Jiming Liu, and Sheung-Ping Chan, "A genetic agent-based negotiation system," *Comput. Netw.*, vol. 37, no. 2, pp. 195–204, 2001.

[50] M. Osborne and A. Rubinstein, *Bargaining and Markets*, The Academic Press, 1990.

[51] David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[52] Ole J. Mengshoel and David E. Goldberg, "The crowding approach to niching in genetic algorithms," *Evol. Comput.*, vol. 16, no. 3, pp. 315–354, 2008.

[53] Bo An, Kwang Mong Sim, Chun Yan Miao, and Zhi Qi Shen, "Decision making of negotiation agents using markov chains," *Multiagent Grid Syst.*, vol. 4, no. 1, pp. 5–23, 2008.

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

[55] Martin J. Osborne and Ariel Rubinstein, *A Course in Game Theory*, MIT Press, 1999.

[56] Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings, "Optimal agendas for multi-issue negotiation," in *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems.* 2003, pp. 129–136, ACM.

[57] Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings, "Optimal negotiation of multiple issues in incomplete information settings," in *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems.* 2004, pp. 1080–1087, IEEE Computer Society.