**NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS**
**SCHOOL OF SCIENCE**
**DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS**

**INTERDISCIPLINARY POSTGRADUATE STUDIES PROGRAM IN ADMINISTRATION AND ECONOMICS OF TELECOMMUNICATION NETWORKS**

**M.Sc. THESIS**

# On the Use of Optimization Techniques for Strategy Definition in Multi Issue Negotiations

**Kyriaki P. Panagidi**

**Advisors:** **Efstathios Hadjiefthymiades,** Assistant Professor NKUA
**Konstantinos Kolomvatsos,** PhD Candidate NKUA

**Athens**

**October 2012**

**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗ ΔΙΟΙΚΗΣΗ ΚΑΙ ΟΙΚΟΝΟΜΙΚΗ ΤΩΝ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΔΙΚΤΥΩΝ**


**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

# Εφαρμογή Τεχνικών Βελτιστοποίησης για τον Καθορισμό Στρατηγικής σε Διαπραγματεύσεις Πολλαπλών Χαρακτηριστικών


**Κυριακή Π. Παναγίδη**

**Επιβλέποντες:** **Ευστάθιος Χατζηευθυμιάδης,** Επίκουρος Καθηγητής ΕΚΠΑ
**Κωνσταντίνος Κολομβάτσος,** Υποψήφιος Διδάκτωρ ΕΚΠΑ



**ΑΘΗΝΑ**

**Οκτώβριος 2012**

**M.Sc. THESIS**

On the Use of Optimization Techniques for Strategy Definition in Multi Issue Negotiations


**Kyriaki P. Panagidi**

**A.M.:** ΜΟΠ297



**Advisors:** **Efstathios Hadjiefthymiades,** Assistant Professor NKUA
**Konstantinos Kolomvatsos,** PhD Candidate NKUA

**Examiners:** **Efstathios Hadjiefthymiades,** Assistant Professor NKUA
**Dimitris Varoutas,** Assistant Professor NKUA



October 2012

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

# Εφαρμογή Τεχνικών Βελτιστοποίησης για τον Καθορισμό Στρατηγικής σε Διαπραγματεύσεις Πολλαπλών Χαρακτηριστικών


**Κυριακή Π. Παναγίδη**

**Α.Μ.:** ΜΟΠ297

**Επιβλέποντες:** **Ευστάθιος Χατζηευθυμιάδης,** Επίκουρος Καθηγητής ΕΚΠΑ
**Κωνσταντίνος Κολομβάτσος,** Υποψήφιος Διδάκτωρ ΕΚΠΑ

**ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:** **Ευστάθιος Χατζηευθυμιάδης,** Επίκουρος Καθηγητής ΕΚΠΑ
**Δημήτρης Βαρουτάς,** Επίκουρος Καθηγητής ΕΚΠΑ



Οκτώβριος 2012

# ABSTRACT

In this thesis, we deal with the problem of decision making in automated negotiations. We consider the case where software agents undertake the responsibility of representing their owners in such negotiations. The final aim is to provide an efficient algorithm in which software agents will act in a scenario of concurrent negotiations. Agents have no knowledge on the opponents' characteristics. Negotiations are held for the exchange of products between buyers and sellers with specific returns. Each product is characterized by a set of issues. For example, a product could be characterized by its price, delivery time, and so on. The buyer is involved in concurrent negotiations with a number of sellers.

We propose algorithms that try to solve the problem of handling the uncertainty with the final aim of maximizing the entities rewards. The reward is calculated as a weighted sum of the discussed issue values. We focus on the buyer side and define specific methodologies for defining the weights that affect the utility of the buyer. Moreover, we propose a methodology for changing the strategy of the buyer in order to reach the optimal agreement. We are based on the widely known Particle Swarm Optimization (PSO) algorithm that is implemented by software agents' movements in N-dimensional space to reach the optimal solution. We present a number of experiments for the proposed methodologies that show their performance and we compare our results with results found in the literature.



**SUBJECT AREA**: Artificial Intelligence

**KEYWORDS**: software agents, automated negotiations, multi-issue negotiation, weights definition, Particle Swarm Optimization, Virtual Force Algorithm, Simplex

# ΠΕΡΙΛΗΨΗ

Στην παρούσα διπλωματική εργασία αναλύεται το πρόβλημα της λήψης απόφασης σε συστήματα αυτόματων διαπραγματεύσεων. Σκοπός είναι να σχεδιαστεί ένας αποδοτικός αλγόριθμος βάσει του οποίου οι πράκτορες λογισμικού θα δρουν σε ένα σενάριο ταυτόχρονων διαπραγματεύσεων.Οι πράκτορες δεν έχουν καμία πληροφόρηση για τα χαρακτηριστικά των αντιπάλων.Οι διαπραγματεύσεις πραγματοποιούνται με απώτερο στόχο την ανταλλαγή προϊόντων μεταξύ αγοραστών και πωλητών με συγκεκριμένα ανταλλάγματα. Κάθε προϊόν χαρακτηρίζεται από μια ομάδα χαρακτηριστικών. Για παράδειγμα, ένα προϊόν μπορεί να χαρακτηριζεται από την τιμή, από το χρόνο παράδοσης, κλπ.

Κάθε αγοραστής αντιστοιχίζεται στις αυτόματες διαπραγματεύσεις με έναν αριθμό πωλητών. Προτείνουμε αλγόριθμους που προσπαθούν να επιλύσουν το πρόβλημα προσέγγισης αβεβαιότητας με τελικό σκοπό τη μεγιστοποίηση της ανταμοιβής των χρηστών. Η ανταμοιβή υπολογίζεται ως το άθροισμα με τα αντίστοιχα βάρη των χαρακτηριστικών. Εστιάζουμε στην πλευρά του αγοραστή και ορίζουμε μεθοδους για τον υπολογισμό των βαρών που επηρεάζουν τη χρησιμότητα του χρήστη. Πιο συγκεκριμένα, προτείνουμε μεθόδους για την αλλαγή της στρατηγικής του αγοραστή με στόχο να προσεγγίσουμε την καλύτερη συμφωνία. Ακόμα, χρησιμοποείται ο αλγόριθμος της θεωρία του Σμήνους (Particle Swarm Optimization Algorithm) ώστε μέσω της κίνησης στο Ν-διαστατο χώρο να συγκλίνουν οι πράκτορες λογισμικού στη βέλτιστη συμφωνία. Παρουσιάζεται, τέλος, ένας αριθμός από πειράματα για τις προτεινόμενες μεθόδους για να αξιολογηθεί η απόδοσή τους και να συγκριθούν τα αποτελέσματα με τη σχετική βιβλιογραφία.



**ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ**: Τεχνητή Νοημοσύνη

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ**: πράκτορες λογισμικού,αυτόματες διαπραγματεύσεις, διαπραγματεύσεις πολλαπλών χαρακτηριστικών, επανακαθορισμός βαρών

*I would like to dedicate this thesis to my beloved family,*
*to my friends and my colleagues*
*for their support*

## CONTENTS

## LIST OF FIGURES

# LIST OF TABLES

# PREFACE

This master thesis was carried out in the interdisciplinary postgraduate studies program "Administration and Economics of Telecommunications Networks" of the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens. The goal of my research has begun with the ambition of exploring a problem which is not so theoretical but found in everyday applications. More specifically, our aim was to study automated negotiations and to develop optimized decision making methods in order to reach the best agreement in the field of automated real life negotiations.

I would like at this point to express my warmest thanks to my thesis supervisor, Assistant Prof. Efstathios Hadjiefthymiades, for guidance and valuable contributions during my study of the master thesis. I would also like to thank the Konstantinos Kolomvatsos, Ph.D. candidate of Department of Informatics and Telecommunications, for his helpful guidance and advice throughout the preparation of this work. Their continuous monitoring of the progress of the master thesis, their meaningful remarks and their help for resolving any problem have contributed in the final formation of this thesis.

# 1. INTRODUCTION

Nowadays, Electronic Commerce is rising exponentially due to the fact that offers to buyers and sellers a dynamic space to interact. In Electronic Commerce applications, one can find virtual places, called electronic marketplaces, where both parties (buyers and sellers) can exchange/negotiate efficiently for products with specific returns. The interaction between entities, in such places, is called negotiation. Actually, negotiations are defined as a decentralized decision making process that seeks to find an agreement that satisfies the requirements of two or more parties. Software agents can “go shopping” for a user by taking specific user preferences and return with recommendations of purchase which meet these specifications. Moreover, agents can represent sellers and undertake the responsibility of selling products in the most profitable prices. So, there is the need for software agents to be intelligent, e.g. to make reasonable decisions answering questions like what, when, how etc.

In thesis, we propose algorithms to enhance the intelligence of software agents to accomplish their goals. Our aim is to develop decision making algorithms which can be applied to real life negotiations. Software agents act in concurrently, one-to-many, limited knowledge and multi-issues negotiations. In this case, buyers have direct interactions with sellers and utilize a number of sun-negotiators (threads) for having a possible agreement. We focus on the buyer side and as our problem is quite complicated, we have designed four algorithms that can dynamically change her strategy during negotiations without the need of a coordinator. The first three algorithms redefine the weights of product’s issues resulting to a change in the calculation of the utility function. The methods which are used for weights configuration are Heuristic, Simplex and Analytic Hierarchy Process trying to optimize the user’s utility. In these algorithms, the value of the utility will increase if and only if the seller will improve the proposed offer. Also, we approach our problem through moving software agents in the N-dimensional space applying the Particle Swarm Optimization algorithm (PSO). The N-dimensional space depicts the number of issues which take part in negotiation. The offer made by each software agent is at every round a new position in space, which depends on local and global best of rest software agents as well as the velocity of the same. All methods have been simulated and their performance is extensively compared.

The rest of master thesis is organized as follows:

Chapter 2 contains an analysis of electronic commerce and its growth. Chapter 3 presents an extensive overview of the software agents and their characteristics. Chapter 4 defines the electronic marketplaces as well as the examples of relevant electronic marketplaces. In chapter 5 we discuss the negotiations as a whole. We present the negotiations' definition, the types in which negotiations can be classified, the negotiations' strategies and mechanism. Our problem and the relevant proposed work follow in the chapter 6. In chapter 7, the methodologies for the configuration of weights are presented combined with the description of their algorithms. Chapter 8 contains a number of experiments for the proposed methodologies and the assessment of their performance. Finally, in chapter 9 we discuss the open issues and the future work which can be further studied.

# 2. ELECTRONIC COMMERCE

## 2.1 Introduction

The increasing acceptance of Internet accompanied by the World Wide Web technology has changed our way of conducting business and financial transactions thereby helping to realize the goal of Adam Smith's *invisible hand*. Adam Smith's *invisible hand* argument presented the concept that opening up a market will result in a globally efficient mechanism where buyers have the ability to choose freely what to buy and sellers or service providers are allowed to choose freely what and how to produce and sell. This in turn leads the market to settle on product and price distributions that are beneficial to all members of the community. Electronic Commerce (E-Commerce) is defined by the Electronic Commerce Association as [1]:

"*any form of business or administrative transaction or information exchange that is executed using any information and communications technology*" .

Moreover, E-Commerce is referred in literature as the business practice related to buying and selling goods, products or services, in the Internet.

It is widely accepted that the E-Commerce is rising exponentially, nowadays, due to the reason that it offers to buyers and sellers a new environment where the two parties could negotiate in electronic marketplaces (E-Marketplaces) combined with efficiency, reliability, security, and smaller cost. An E-marketplace is a virtual environment where entities (e.g., buyers and sellers) interact in order to exchange products for specific returns. The owners of the commodities want to sell them to every interested member of the community. On the other side, there is a demand for these products and a group of entities is willing to pay in order to obtain them. Products could be of different type (books, newspapers, articles, electronics, etc). E-Marketplaces are highly dynamic based on the number and the behavior of the involved entities. They provide convenience and means for quick response to requests for available products or services. They provide a number of advantages. A typical one is that they can offer traditional merchants an additional channel to advertise and sell products to buyers thus potentially increasing sales. In addition, online markets are more efficient than their physical-world counterparts, thus, lowering transaction costs for both merchants and buyers. For example, the low transaction costs are one reason why Amazon.com, a virtual bookstore, can offer a greater selection and lower prices than its physical-world competitors. Through such mechanisms buyers and sellers not known in advance can interact for the exchange of products/services for specific returns.

Electronic payment (E-payment) has helped to the further growth of the E-Commerce, which is defined as [1]:

"*the operation through which the owner of a product takes specific benefits from selling it to a buyer*"

At first step required for an E-payment action, sellers define a price that is based on their cost and propose it to potential buyers. If buyers decide to close an agreement then the act of payment is held. Measures should be taken in order to facilitate the payment process as well as to control fraud cases. Furthermore, E-Commerce should define mechanisms for the manipulation of other issues such as sellers' advertisements, customers/providers registration, and information searching and trust mechanisms.

### 2.2 Types of transactions in E-Commerce

E-Commerce contributes to the creation of a wide and open business market, which is capable of offering efficiency to purchase actions. Nowadays, there are many categories of virtual enterprise applications, found in the Web, which are classified by several researchers, depending on the application context. E-Commerce is divided, taking into consideration the entities involved (consumers or businesses), in four categories. In Table1 we can see the discussed classification scheme.

Table1: E-Commerce classification

| | Consumer | Business |
|---|---|---|
| Consumer | Consumer-to-Consumer<br>Example: Ebay | Consumer-to-Business<br>Example: PriceLine |
| Business | Business-to-Consumer<br>Example: Amazon, Dell | Business-to-Business<br>Example: IBM, SAP |

- B2B (Business-to-Business): The entities involved in the transaction process are enterprises, not final consumers. Businesses negotiate with each other such as manufacturers selling to distributors and wholesalers selling to retailers. In this way, the enterprise buys products, which are used as raw material to other products. Pricing is based on quantity and, often, it is negotiable.

- B2C (Business-to-Consumer): This category is related to applications that support commercial transactions among final consumers and enterprises. This category is well studied through several enterprise Websites offering their products through electronic catalogs. The final consumer can place electronic orders and pay for them. Websites such as Amazon and Dell are good examples of this category.
- C2B (Consumer-to-Business): A consumer posts his project with a budget online and companies as soon as possible review the consumer's requirements and bid on the project. The consumer reviews the bids and selects the company that will complete the project. The difference between this category and the Business-to-Consumer is related to how the product or services are traded. In this category, the final consumers indicate to the enterprise what they want to buy and how much they would like to pay for the product. In Business-to-Consumer category, the process is the opposite: the enterprise gives the exact price of its products.
- C2C (Consumer-to-Consumer): There are many Websites offering free auctions, and forums where users can buy and sell. Such Websites handle payments by utilizing known online payment systems like PayPal. Users can easily send and receive money online, for example the eBay's auction service. In this category, there is not the figure of an enterprise as an legal entity but a consumer who offers some product or service.

Given the ubiquity and importance of transactions in various contexts, research for developing models and techniques handling transactions have attracted a lot of attention. The research field of Intelligent Software Agents (IAs)[2] combine social psychology, operations research, and more recently agent mediated E-Commerce. The need for research has emerged due to the large body of theoretical and empirical market literature that has been published in the field, particularly in the area of auction based protocols. E-Commerce transactions over the Internet have been the driving factor for this research which has presented a large number of challenges in the design of E-Marketplaces. Currently, designers need to deal with geographically distributed traders who have multiple complex factors that they need to consider in their negotiations. In general, a market designer’s task is to create a meeting place for buyers and sellers; and a transaction protocol that enforces a set of rules. Such rules will lead to the “*desired*” outcome. This outcome will be represented as the final allocation of the traded objects and by the exchange of payments between the participants. Market design is still in infancy stages and comprises of tools and methodologies such as (i) equilibrium analysis, (ii) mechanism design theory, (iii) experimental economics and (iv) computation.

# 3. INTELLIGENT SOFTWARE AGENTS

## 3.1 Introduction

People recognize eCommerce's convenience and ability to offer a quick response to requests for available products or services. As this adoption spreads, the incentive for employing software agents increases to enhance and to improve the trading experience [4]. The entities participating in electronic transactions can be represented by Intelligent Software Agents (IAs). IAs take action based on human decision making behavior. IAs can help users to perform some actions involving search, negotiation, trade off and so on to improve effectiveness. They can act without coordination representing the interests of their owners. For those reasons, IAs represent one of the most interesting approach and innovative technology satisfying massive individualized needs of the participants.

## 3.2 Definition of Intelligent Software Agents

Software agents or Intelligent Agents (IAs) represent one of the most interesting approach and innovative technology satisfying massive individualized needs of the participants. Software agents offer greater flexibility and adaptability than traditional software components. Agent-oriented software engineering allows developers to use high-level and flexible abstractions to represent and understand Web-based enterprise application systems. Rapid integration of distributed agents provides opportunities to build such software. IAs are autonomous components that have their own goals and beliefs and can reason about their present and future behavior offering opportunities for rapid, incremental development of Web-based enterprise application systems.

There is no universally accepted agreement for a definition of an IA, probably because each definition rises from each application domain; some of the IA definitions follow:

1. *"Intelligent software agents are programs acting on behalf of their human users" [*19].
2. *"Intelligent software contains features as perception, interpretation of natural language, learning and decision making" [*20]*.*
3. *"A piece of software which performs a given task using information gleaned from its environment to act in a suitable manner so as to complete the task successfully. The software should be able to adapt itself based on changes occurring in its*

*environment, so that a change in circumstances will still yield the intended result."* [52].

4. *"Software agents carry out certain operations on behalf of a user or another program with some degree of independence or autonomy combined with a set of goals or tasks for which they are designed"* [21].
5. *"Intelligent Agents are computerized servants, it is software that communicates, cooperates and negotiates with each other. They have the ability to take over human tasks and interact with people in human like ways. They are bringing technology into a new dimension simplifying the use of computers, allowing humans to move away from complex programming languages creating a more human interaction"* [22].

In general, the definition of IAs, on the one hand is based on the definition of the term agent as a person or thing (who is authorized to act on behalf of a third party) and on the other hand, it is based on software, which can be adapted to the individual preferences and parameters of its human user. Such software can operate without users' intervention for solving a specific problem. Users only need to specify a high-level goal, leaving "how" and "when" decisions to the IA. The reasoning mechanisms of IAs can range from a set of simple rules to more complicated algorithms such as Neural or Bayesian networks [23].

Any IA has an instructor (a person or a superior software program), who instructs it to operate a certain functionality. The communication with the environment is held through a specific interface. Adding in a simple software agent the attribute of "intelligence", it is converted into an "intelligent software agent". Intelligence is the degree of reasoning and learned behavior. It is the IA's ability to accept the user's statement of goals and carry out the task delegated to it [23]. In the IA's environment, there should be specific rules representing the user profile with her preferences. These rules should be included in a user model or some other form of management system, which will contain a framework for the construction, maintenance and enforcement of user policies. The IA should be capable of a dynamic reassessment of its environment and discovering new relationships, connections or concepts independently from the human user. Users should exploit IA's capabilities in anticipating and satisfying their needs. IA's dynamic interface, in general, provides means for defining input as well as output information having the capability of noticing events fired by its environment.

According to Figure 1, an IA indicates the following characteristics:

- *Goal oriented*: IA is highly aware of what the user needs and is responsible for deciding how is going to satisfy her. The decision is a high-level task and is, usually, separated into smaller sub-tasks, which can be addressed more effectively;
- *Collaborative*: IA should not blindly accept and execute instructions, but should take into account that the user provides ambiguous or erroneous information. So the IA should have the ability to modify the human user requests and ask for additional information or clarifications. For example, an IA should check things by asking questions to the user or even refuse to execute certain tasks, because they would put an unacceptable high load on network tasks;
- *Autonomous*: the IA should operate independently, without the direct intervention of her instructor, be able to take initiatives and control over her actions and internal state;
- *Rational*: the IA should always try to do what is asked for and act in order to achieve the user's goals reasoning about its goals, acquired information and knowledge about other IAs and users;
- *Adaptive*: the IA should recognize the preference of her user, her habits and working methods based on previous experiences;
- *Social*: the IA should interact with other IAs, programs or humans through her interface and avoid conflicts;
- *Adaptable*: the IA should dynamically assess which actions to execute and when or add new capabilities changing her behavior;

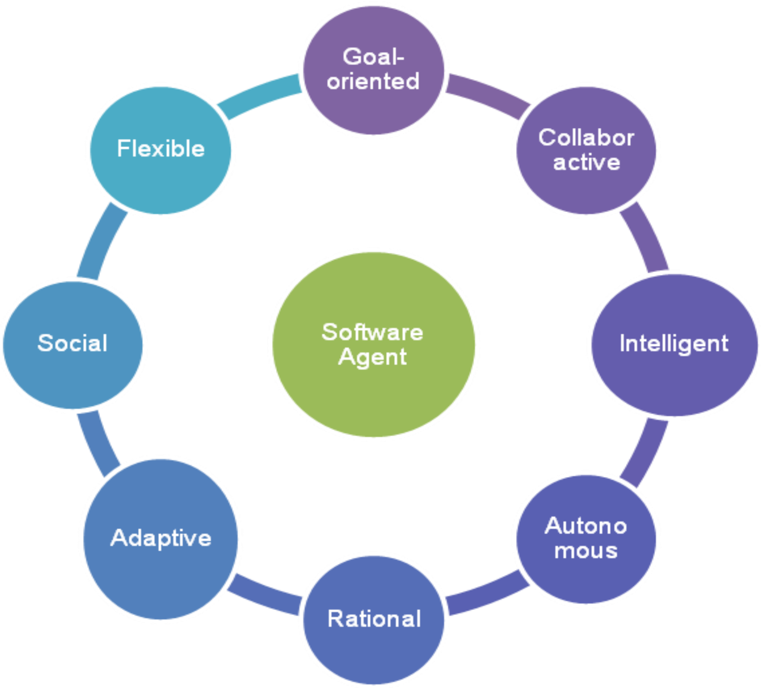


Figure 1: Characteristics of an IA

### 3.3 Types of Intelligent Software Agents

Universities, research centers and companies, such as IBM and Microsoft, had funded for doing research in the area of IAs and had developed applications with notable results. The types of IAs, that have been used, in this low-level applications are [24]:

- Desktop IAs: IAs manage user emails by sorting detailed e-mails automatically into special subjects, make entries in the electronic agenda and carry out assistance for the non-expert users of standard software.
- Network IAs: IAs access distributed information in networks, in order to fulfill the user needs. Networks IAs are further divided into Internet and Intranet IAs.
- Personal IAs: They interact directly with a user, presenting some "personality" or "character", monitoring and adapting to the user's activities, learning user's style and preferences, and automating or simplifying certain rote tasks. Microsoft’s IAs "Bob" or "Paper Clip" is simple examples built using this technology.
- Mobile IAs: It is sent to remote sites to collect information or perform actions and, then, return with results. "Touring" IAs visit sites to aggregate and analyze data, or perform local control. Such data intensive analysis is often better performed at the source of the data rather than shipping raw data; examples include network management IAs and Internet spiders.
- Collaborative IAs: They communicate and interact in groups, representing users, organizations and services. Multiple IAs exchange messages to negotiate or share information. Examples include online auctions, planning, negotiation, logistics/supply chain and telecom service provisioning.

Nevertheless, more complicated and integrated applications have been developed based on use of IAs. The work presented in [23] presents the current trends and promising research fields in the area of IAs. The respective applications domains presented in this above work are the following:

1. *Systems and Network Management*: It is one of the earliest application domains to be enhanced using IA’s technology. Recently by moving to client/server computing has intensified the complexity of systems being managed, especially in the area of LANs. As network centric computing becomes more prevalent, this complexity further escalates. As a result, users in this area (primarily operators and system administrators) need greatly simplified management, in the face of rising complexity. IA architectures have existed in the systems and network management area for some time, but these IAs are generally static rather than dynamically adaptive IAs.

However, IAs are used to enhance systems management software. For example, they can help filter and take automatic decision for actions at a higher level of abstraction, and can even be used to detect and react to patterns in system behavior. Furthermore, they can be used to manage configurations dynamically and adapt to context changes (e.g., user behavior).

2. *Mobile Access / Management*: As computing becomes more pervasive and more importance is given to networks, support of mobile users becomes another challenging task. Users not only do they want to access network resources from any location, but also they want to access those resources despite bandwidth limitations of mobile technology such as wireless communication, and despite network volatility. IAs, which (in this case) reside in the network rather than on the users' personal computers, can address these needs by persistently carrying out user requests. In addition, IAs can process data at the source and ship only compressed/aggregated answers to the user, rather than overwhelming the network with large amounts of raw data.
3. *Mail and Messaging*: Messaging software (such as e-mail clients) has existed for some time, and is also an area where IA function is currently being used. Users today would like the capability of automatically prioritizing and organizing e-mails or to use an organizational application instead. IA can facilitate all these applications by allowing mail handling rules to be specified ahead of time, and letting IAs operate on behalf of the user according to those rules. Usually, it is also possible (or at least it will be) to have IAs that deduce these rules by observing a user's behavior and trying to find patterns in it.
4. *Information Access and Management*: It is a highly active domain, given the rise in popularity of the Internet and the explosion of data available to users. IAs try to help users not only with search and filtering activities, but also with categorization, prioritization, selective dissemination, annotation, and (collaborative) sharing of information and documents.
5. *Collaboration*: It is a fast-growing domain where users work together on shared documents, using personal video conferencing or sharing additional resources through the network. One common denominator is shared resources; another is teamwork. Both are driven and supported by the move to the network centric computing. Not only do users in this area need an infrastructure that will allow robust, scalable sharing of data and computing resources, but they also need

functionality to help them to actually build and manage collaborative teams of people.

6. *Workflow and Administrative Management*: Administrative management includes both workflow management and domains such as computer / telephony integration, where processes are defined. In these domains, users need not only to make processes more efficient, but also to reduce the cost of human errors. Much as in the messaging area, IAs can be used to ascertain, then automate user wishes or business processes.
7. *E-Commerce*: It is a growing area fuelled by the popularity of the Web. Buyers need to find services and product information (including technical specifications, viable configurations, etc.) matching their needs. They need to obtain expert advice both prior to the purchase and support afterwards. Sellers need to find buyers and to provide them expert advice about their products as well as customer service and support. Both sellers and buyers need to automate the handling process of their "electronic financial affairs". IAs can assist in E-Commerce in a number of ways. IAs can "go shopping" for a user by taking specifications and returning with recommendations of purchases which meet those specifications.

   More specifically in the E-Commerce, IAs can be used both by sellers and buyers. Buyers use IAs in order to execute actions like search for goods, multi-issue assessment of the sellers etc. Buyers' upper goal is to find goods (products or services), which correspond to the most to their preferences, and at the same time they have the minimum cost. Buyer's IA try to reach with seller 's IA an agreement, in which the profit will be maximized and the purchase process will more efficient and effective due to sinking costs. In real world the interactions / negotiations work under strictly incomplete knowledge on the characteristics of the other entity. Such characteristics are the negotiation deadline, the lower or higher acceptable price, etc. Entities want to maximize their profits and follow a negotiation strategy while trying to buy or sell products. Users may pose money, time and product type limitations to their IAs and wait for their results. Furthermore, IAs could represent product sources in the E-Marketplaces while being capable of handling many clients simultaneously.

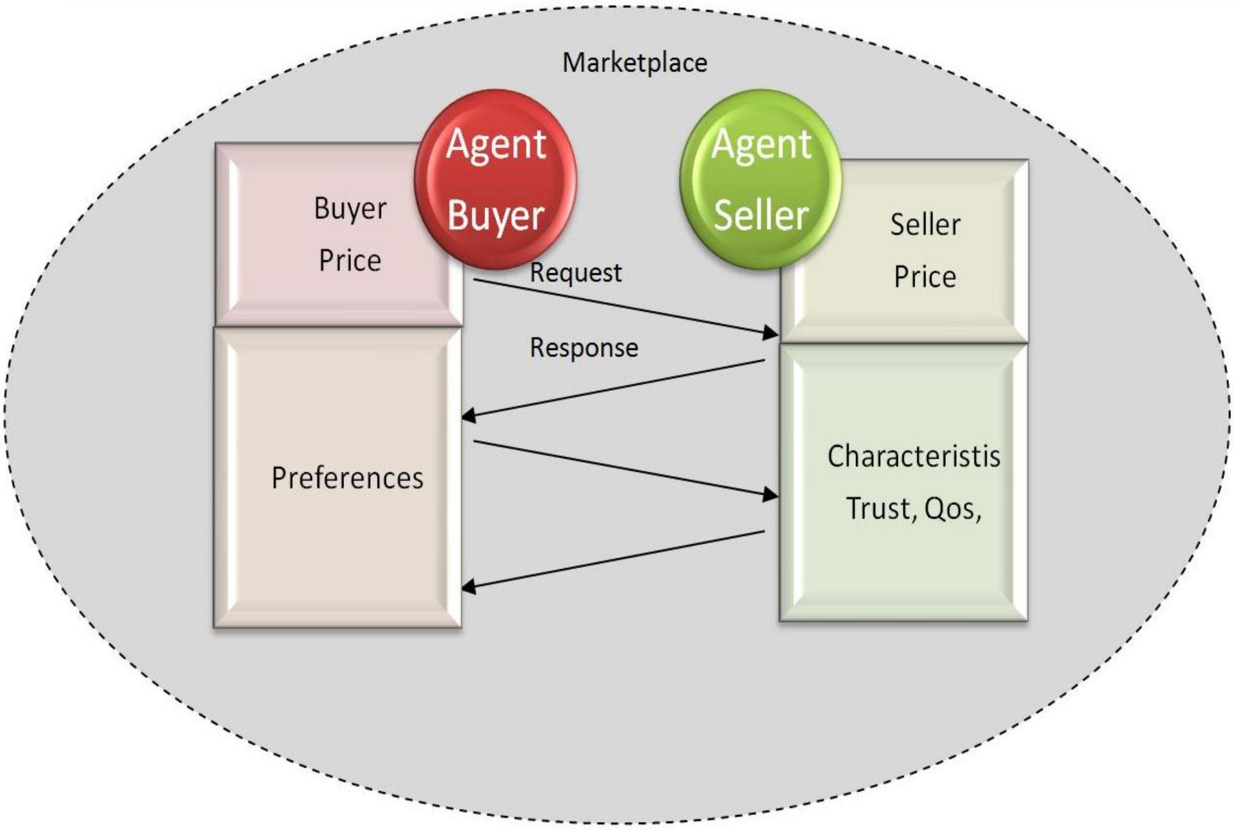


Figure 2: Example of seller's and buyer's negotiation

8. *Adaptive User Interfaces*: Although the user interface was transformed by the advent of graphical user interfaces (GUIs), for many users remain difficult to learn and use. As capabilities and applications of computers improve, the user interface needs to accommodate the increasing complexity. As user populations grow and diversify, computer interfaces need to learn user habits and preferences and be adapted to every individual. *Interface agents* can help in both problems. IA technology allows systems to monitor the user's actions, develop models of user abilities, and automatically to be triggered when problems arise. When combined with speech technology, IAs enable computer interfaces to become more human or more "social".

### 3.4 Barriers

Whereas IAs appear to be beneficial for users conducting E-Commerce, it is important to reveal their limitations. First of all, IAs, as long as they search websites, should have access to their catalogues. Second, according to the characteristics of these sites, the user goals have to be specified. Third, users have to obtain information such as prices, product's issues, returning policies, delivery time, while switching from one site to another. Last, but not least, security problems may occur when submitting sensitive information; most of these operations are complex. It is, also, reported that the progress of the E-Commerce faces three obstacles, such as slow response time, lack of user friendliness and poor website design. It is widely accepted that IAs have the ability to address some of these obstacles, for example mobile IAs handle slow response time.

# 4. ELECTRONIC MARKETPLACES

## 4.1 Definition of Electronic Marketplace

An Electronic Marketplace (E-Marketplace) can be considered as a virtual location where entities that are not known in advance can cooperate in order to achieve common goals. These entities have their own preferences and strategies. Furthermore, entities participating in an E-Marketplace work in open environments where either their preferences or their type and number may change continually. E-Marketplaces are characterized by their dynamically involved entities. We can name such markets as open nature (i.e., the number and the mechanism of the markets). . One of the most important characteristics of E-Marketplaces is that they do not have barriers in contrast to physical markets. The number of buyers and sellers could change at every time, thus, transforming the basic characteristics of the markets such as the demand or the supply of products. Buyers and sellers may join and leave at every time without notification. Moreover, entities may continuously change their preferences.


.

Figure 3: Electronic Marketplace

For example, buyers may choose to care for various products or sellers may change their prices. These issues should be treated by a mechanism provided by the market. For instance, an increased number of buyers lead to an increased demand for products. On the other side, a reduced number of sellers may cause problems in the steady operation of market. The market with a few sellers may be transformed in an oligopoly, in which sellers impose their prices regardless the buyers' demand. Most of the proposed E-marketplace's models are classified in the following two categories:

- Direct transactions among providers and consumers
- IA-based brokered transactions

The direct transactions model has the advantage that purchases can be privately negotiated [4]. Sellers can advertise their goods to buyers either directly or through online catalogs. Buyers can select from the available products the one that fits best in their preferences. Buyers and sellers can propose prices and negotiation is possible. IAs can be involved in the direct transaction model and several commercial and non-commercial online shopping services have been developed. IAs are used to assist sellers in finding bargains, facilitate negotiation among buyers and sellers, or provide help in locating appropriate items over a distributed E-Marketplace.

### 4.2 Information Marketplaces

Information Marketplaces (IM) are places where participants negotiate for the exchange of information commodities. Information goods could be images, videos, music, software code, electronic articles, etc. For these goods, their owners want to sell them to every interested member of the community. Usually, there are groups of market members facilitating them to accomplish their tasks. Such entities manage issues that are related to administration or mediation processes. It should be noted that the referred products could be whole information pieces or even more units of information. For example, in the application domain of electronic articles a user may want to retrieve the entire journal or some of the articles published in it.

### 4.3 Examples of E-Marketplaces

A number of examples of E-Commerce's environments suitable for commercial transactions using IAs are presented in the following sections.

#### 4.3.1 BargainFinder

BargainFinder is considered as the first shopping IA system for online price comparisons [6]. The price comparison begins with the information about which product the buyer want to buy. With this information, BargainFinder requests the price of nine sellers Web sites, used for this purpose. The requests are similar as from a Web browser in.

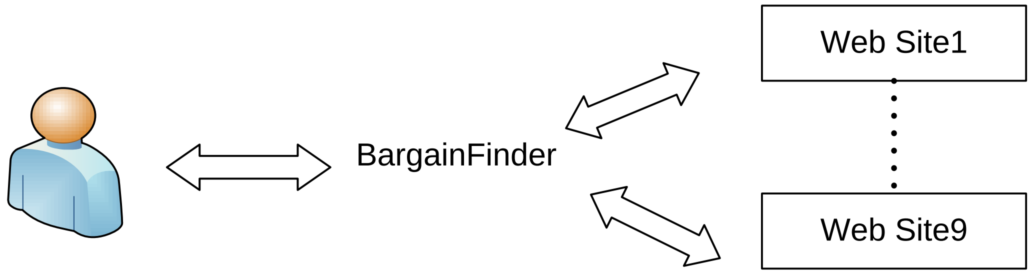


Figure 4: Requests in the BargainFinder

Despite of being a limited system in comparison to other IA systems, BargainFinder raises a series of considerations about the automatic comparison of prices in the Web. One of them is about the refusal of many sellers in taking part of this process. This happens because the sellers do not want their prices to be compared through BargainFinder. Hence, they block access to their Websites. Services that increase the price of the product are practically ignored by the IA and, consequently, are not considered in the buying decision of the consumer.

### 4.3.2 Jango

Jango has the same philosophy as BargainFinder. It allows price comparison based on the available information in the seller Website [9]. Jango can be considered an evolution of BargainFinder because its access is not blocked to any Websites. For this reason, Jango requests are handled as if they were built directly from the browser of any user. The seller's Website does not distinguish the access that is accomplished by Jango from the others that are accomplished by users. Afterwards, Jango presents the information obtained from the Websites so that the consumer can compare the product prices.

### 4.3.3 Miner

The Miner family of IAs is a set of tools whose main objective is to help people to find information in the Web [17]. The main idea is to bring multiple information sources together in one place. Searching is performed by a number of IAs working in parallel, collecting answers and aggregating them. The Miner family provides brokerage services including:

- BookMiner – searches for books in registered Brazilian and international bookstores to match user's specifications.
- CDMiner – searches for music titles in Brazilian and international music stores to find the user's preferences.

These two services work similarly. Each query process can be divided into five main steps, as follows:

- user submits a query;
- the Miner server gets the query and dispatch IAs;
- each IA queries the target store;
- each IA receives and parses the query results;
- the server unifies, formats and sends the results to the user.

Figure 5 presents the discussed steps.

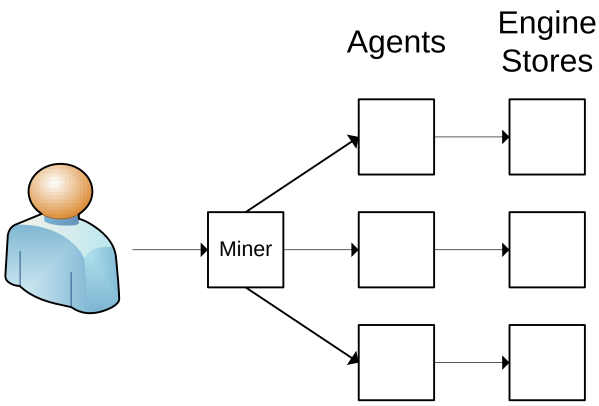


Figure 5: Miner family - Steps of a query task.

### 4.3.4 Tete-a-Tete (T@T)

T@T provides IAs for buyers and sellers. These IAs interact to each other and negotiate trying to satisfy the needs of buyers and sellers. The seller's product is defined as a complete assessment based not only on the price of the product offered, but also on other dimensions, such as the delivery time, support services, brand and reputation. The additional comparison features guarantee a complete perspective for the real value of the product for the buyer. T@T for the seller automates the negotiation process. For the buyer, T@T provides assistance during the negotiations offering decision support functionality to determine which seller has the best offer. Thereby, T@T provides an interface that allows the complete presentation of sellers' characteristics and presents which products match the buyer's needs.

### 4.3.5 Kasbah

Kasbah is an online multi-agent system where users create IAs to help during the negotiation process. This way, a user, who wants to buy or sell a specific product, first creates an IA with strategic directions and sends her to a central E-Marketplace [50]. The objective of this IA is to complete an acceptable agreement according to the defined strategies. Some examples of information defined in such strategies are reservation prices and the deadlines. The reservation prices are defined as the range of acceptable product's prices for the buyer or seller. The deadline is defined as the time that the IA should use to conclude a transaction. Once buyer and seller IAs find complementary interest (i.e., one wants to sell and the other wants to buy the same product) the negotiation starts. This process is direct and the only action which is allowed into negotiation protocol is a bid by the buyer IA, while the seller IA can only answer "yes" or "no" to this bid. The message exchange happens until IAs reach an agreement. In the meantime, the buyer bids are based on the specified three strategies corresponding to the degree of anxiety:

- If IA is "anxious", then matches with a linear function
- If IA is "cool-headed", then matches with a quadratic function
- If IA is "frugal", then matches with an exponential function

### 4.3.6 Market Space

Market Space is an open agent-based market infrastructure. It is based on a decentralized model in which humans and machines can get information about products and services. Everyone is able to announce interests to another [18]. The aims of Market Space is to build an E-Marketplace where searching, negotiation and agreement take place using IAs based on models for information and interaction. The information model is responsible for the use of information in processes automatically. The interaction model contains all the functions needed by the IA to communicate with its environment.

### 4.3.7 MAGMA

MAGMA is a market architecture that includes functionalities required for simulating a real market [4]. Some of these functionalities are: communication infrastructure, mechanisms for storage and transfer of goods, banking and monetary transactions and economic mechanisms for direct or brokered buyer-seller transactions. The aim of this system is to provide all the essential services to the participating IAs. These services are available through an open-standard messaging application program interface (API). MAGMA architecture includes multiple Trader Agents, an Advertising Server, a Relay Server and a Bank. Trader's IAs are responsible for all the required actions in the system such as buying, selling products and negotiating prices. The Advertising Server provides an advertisement service including search and retrieval of ads. The Bank provides a set of basic banking services including, among others, checking accounts, lines of credit and electronic cash. A Relay server was created to facilitate communication between these IAs. As these IAs communicate each other through a socket connection, the Relay server maintains all connections and route messages between IAs. The message routing is based on unique IA names. The MAGMA architecture is presented in Figure 6.

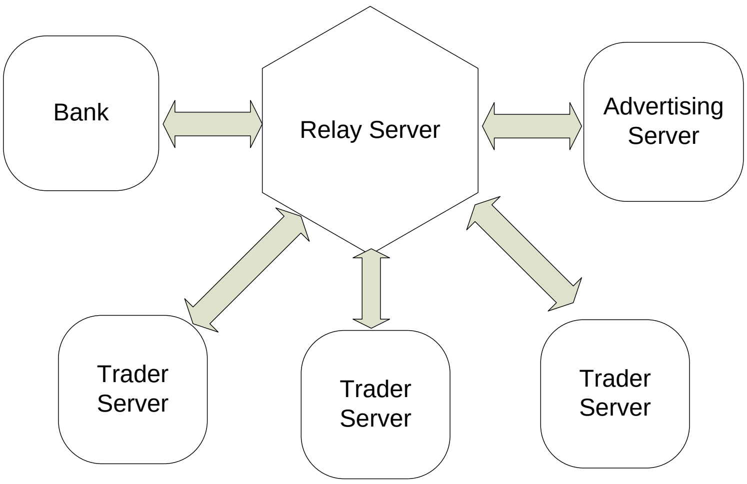


Figure 6: MAGMA architecture

### 4.4 Framework of Electronic Marketplace

A generalized virtual E-Marketplace needs to incorporate mechanisms to facilitate the assessment of the environment's conditions and rules and to assist users through a variety of decision making algorithms. A general negotiation algorithm for the direct transaction model is presented in the following Figure 7 [4]:

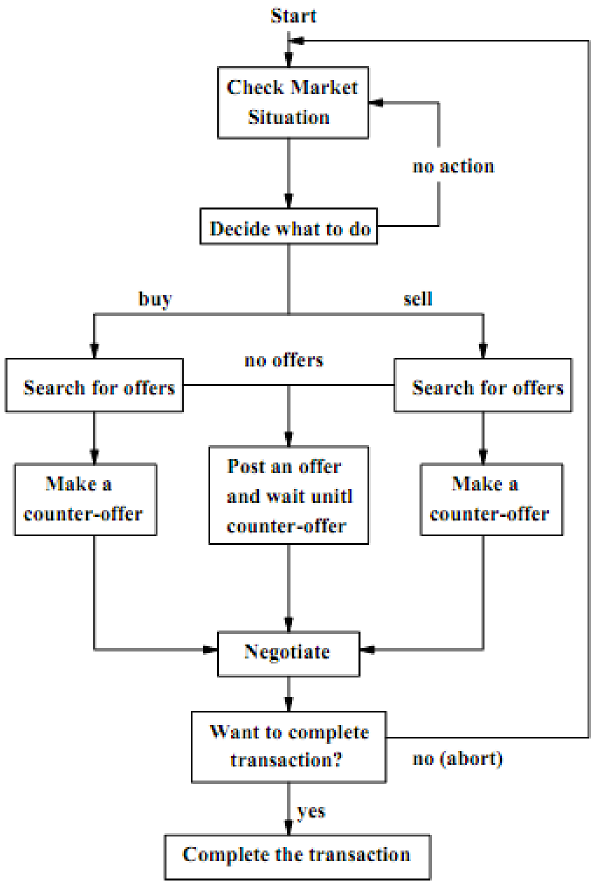


Figure 7: Algorithm for direct transactions between IAs [14].

The E-Marketplace should include mechanisms for direct negotiations, even if they are not always desirable. For example, it may be expensive for sellers and buyers to find each other in a distributed market system or may wish to remain anonymous. Additionally, entities may want to relegate the time consuming task of negotiating for the best price to intelligent IAs that work on their behalf. In such situations the intermediation service model may be more suitable for the E-Marketplace. In E-Marketplaces, such intermediation service must be facilitated by automated IAs that engage in negotiation and/or bid to find the “best” deals (the highest bid from the seller perspective or the lowest bid from the buyer perspective).

Designing effective economical mechanisms for IA negotiations depends heavily on the model used to describe the interaction of the participants. Researchers in distributed AI and in Economics have used game theoretic models to describe the behavior of participants. From this point of view, players (i.e., the buyers and sellers) provide offers for a particular product and are engaged in negotiations according to their strategies. Each player’s strategy guides the line of actions based on the available information. The problem is that, usually, IAs do not have enough information about the complete evaluation of the item offered by others (possibly dishonest). IAs may have to select from a variety of strategic behaviors the one that will either reveal the private information of the competing IAs or will guard against dynamic strategies that can extract their own private information.

### 4.5 Entities participating in Electronic Marketplaces

The entities participating in electronic marketplaces are the following:

- *Participant*: A participant is represented by an IA. They usually represent buyers, sellers and middle members.
    - *The buyers* (*or customers or consumers*). Buyers are users that search for specific products. Each buyer has a different valuation for every product.
    - *The sellers* (*or providers*). Sellers are entities that have in their property a number of products and want to sell them to a number of potential buyers.
    - *The intermediary.* Intermediary is an intermediate member of an E-Marketplace that facilitates buyers and sellers in their interaction process.

  It is necessary to bring the communication and cooperation idea to an environment that integrates virtual enterprises-sellers and buyers, allowing the creation of an agile and efficient E-Marketplace. To accomplish this, the need of electronic intermediaries arises, with the role of coordinating the relevant information to the process of buying and selling products, reducing the transaction costs. For a complete commercial transaction, many steps of the process of buying and selling need to be contemplated. This process is not related only to place an order and its payment: it also includes many other activities. Mougayar, in [8], divides the process of buying and selling into three steps: pre-sale, sale and post-sale. Besides, for each one of these steps, the work specifies their activities, producing a model of buying and selling, called Buyer/Seller Model as shown in Table 2.

Table 2: Buyer/Seller Model [8]

| | Buyers | Sellers |
|---|---|---|
| Pre-sale | Search/Inquire for product<br>Discover product<br>Compare products<br>Negotiate term | Distribution<br>Promotion<br>Display<br>Pricing policy |
| Sale | Place order<br>Receive acknowledgement<br>Initiate payment<br>Receive product | Receive order<br>Authorize payment<br>Schedule order<br>Build/Retrieve from inventory |
| Post-sale | Request support<br>Give feedback | Ship product<br>Receive payment<br>Support products<br>Market research |

This table presents different perspectives of the process of buying and selling: the seller and buyer perspectives. The activities that are presented in these perspectives can be mapped to the E-Commerce. It is important to highlight that the usage of intermediaries in this process should be accomplished considering the systems that already exist in the enterprises, which already give support to some of the activities presented in the above Table 2. The intermediaries should be integrated with all legacy systems, allowing the process to be as automatic as possible. For example, an enterprise, which supports sales based on credit cards, it possibly has systems having direct communication with the credit card management company. This way, the intermediaries in this scenario should not accomplish this activity. They should integrate themselves to the existing systems giving the necessary information so that the process of buying and selling can follow the rest of its natural flow. Joseph Bailey groups the intermediary functions into those following roles [9]:

- Aggregation – Intermediaries can aggregate products among suppliers to reduce transaction costs. They can aggregate the demand of many buyers or the products of many sellers, while still maintaining the interaction between buyers and sellers.
- Pricing – The intermediary determines the price of products based on their demand and offer.
- Search – The intermediaries are repositories of E-Marketplace information, so they have full access, reducing the search costs.
- Trust – Intermediaries can protect buyers and sellers from opportunistic behavior of the E-Marketplace participants.

The behavior of the participants will be monitored and when there is some kind of abnormal situation, the intermediary will associate it to the participant. The irregular behavior has penalty to the reputation of the user (buyer or seller). The interaction between the participants is based on the rich and dynamic knowledge inherent in the E-Marketplace and the technology of IAs, so the intermediaries can interact in a distributed and adaptable manner.

- *Offer*: Each offer contains one or several intentions with specific issues and price.
- *Product*: A product can represent any good or service that can be traded in a market.
- *Issue*: An issue describes a good or a participant. The more issues of product meet the buyer's requirements, the more the buyer's utility increases. The utility in general is defined as the total satisfaction received from consuming a good or a service [11].
- *Agreement*: An agreement indicates that the two opposite intentions have committed themselves to exchange the product with a specific price. Each IA has a private upper (lower) limit, which is a maximum (or minimum) that must be respected in reaching a deal as shown in Figure 8. If the agreement zone is empty, the deal is not possible.

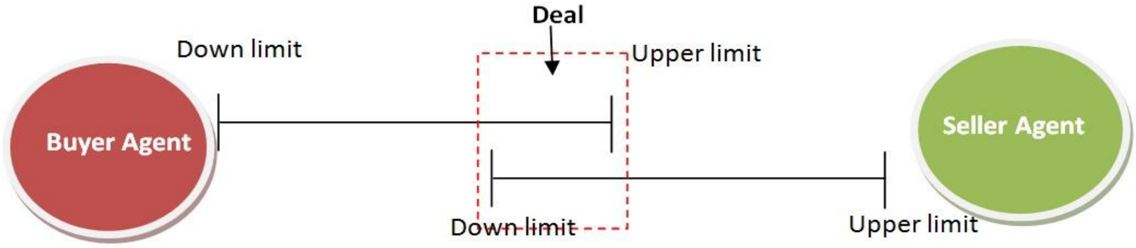


Figure 8: The arrangement zone between seller and buyer

The seller IA will represent the seller interface in this marketplace. It is responsible for informing which products are for sale and the ways of negotiation that are allowed in a commercial transaction with the buyer. The intermediary IA exists with the objective to coordinate the information related to the marketplace. This includes information about buyers and sellers and what is being offered and demanded. In the electronic Marketplace, it is responsible for allowing the activities about the aggregation and search of products presented in the beginning of this section. Many times, the communication between buyers and sellers is accomplished by an intermediary IA. Only after the successful negotiation of the buying and sales terms, the buyer himself will interact with the seller application to finish the commercial transaction, using the parameters already traded by the marketplace IAs. The marketplace IAs were specialized from a generic model of a broker which describes IAs who are able to reason, act and communicate, not only based on information coming from the outside world, but also on the information received from other IAs [10].

# 5. AUTOMATED NEGOTIATIONS

## 5.1 Definition of Negotiations

New computing and communication technologies introduce new opportunities for the design and deployment of IAs. Negotiation in general is a decentralized decision-making process used to search and arrive at an agreement that satisfies the requirements of two or more parties in the presence of limited common knowledge and conflicting preferences. Negotiation, also, could be defined as the process where entities try to agree upon the exchange of a product or as a mean of compromise, in order to reach mutual agreements. The systems that are designed to help and advice buyers and sellers (negotiators) during the various phases of the negotiation process are called negotiation support systems (NSSs). Kersten [25] classifies the NSSs considering the phase of negotiation process to:

- Planning systems,
- Assessment systems,
- Intervention systems, and,
- Process systems.

NSSs are used to structure and analyze the negotiation, elicit preferences and use them to construct a utility function, determine feasible and efficient alternatives, set negotiation tactics, visualize different aspects of the problem, and facilitate communication. The contribution of decision theory to negotiation includes decision rules, decision trees, single or multi issue utility theory, and statistical methods such as forecasting or regression analysis. Decision making theory provides to NSSs the means for the methodological support of the participants. Some approaches based on the negotiation analysis aim at bridging the gap between descriptive behavioral models and normative formal models of bargaining. These approaches have adopted a number of behavioral concepts, including reservation and aspiration levels, the best alternative to the negotiated agreement and distributive negotiations. All these alternative approaches had been measured by quantitative models [25].

Negotiations conducted in the Web are called electronic negotiations (E-negotiations) and systems used in E-negotiations are named electronic automated negotiation systems (EANSs). E-negotiation systems are unlike previous systems deployed on stand-alone computers or networks in terms of the implemented mechanisms and employed technologies. Specifically, the features of intelligent software agents have been denoted for their suitability in distributed computing.

### 5.2 Real life negotiation problems

Real life negotiation problems are typically ill defined and information is not equally distributed among the participants. Participants have only partial knowledge about their counterparts and communication is often ambiguous or imprecise. The main objective of negotiations is to improve the efficiency and maximize the user utility. Methods provided by Artificial Intelligence (AI) are useful in many problems, because IAs can use AI based decision-making mechanisms satisfying bounded information, bounded rationality, and bounded computational characteristics. The lack of knowledge between buyers and sellers can be compensated by the IAs' ability to learn and verify the acquired information. IAs need to be able to update their knowledge about their partners as well as their environment. This capability is the prerequisite for IAs negotiating in Electronic Markets in order to be able to adapt their behavior in changing partners and user preferences.

Negotiations, which take place in E-Commerce, should take into consideration several features, in order to get close to "real world" negotiations. Human behavior in real world negotiations involves complex aspects, such as:

- Multiple issues negotiation: negotiation usually involves several issues, such as price, delivery time, taxes, etc.;
- Similar product suggestion: buyers usually do not know precisely which product to buy. They have only an idea of the desired product. In this case, the negotiation can regard similar alternative products;
- Correlated product suggestion: when a buyer buys a television, a seller can offer a discount in the case that the buyer also buys a video as well;
- Ultimatum: when a participant wants to leave the negotiation, she gives an ultimatum to the opponent indicating that this is her last offer. This ultimatum is used to indicate the desire to leave the negotiation process if the last offer is not accepted.
- Negotiation cost: a buyer can buy the product at hand only to avoid the cost (locomotion, parking, etc.) of trying to find a cheaper one somewhere else;
- Learning: the experience of previous negotiations is usually taken into account in the future;

In negotiations we can distinguish two types of knowledge:

- An entity has knowledge of the opponent's characteristics, and,
- An entity has no knowledge about the opponent's characteristics.

The acquisition of the related information such as the opponent's preferences, reservation price, or deadline allows IAs to increase knowledge about the rest of the participating entities. In the second category, an IA has knowledge about only her status and selects strategy from her space of all possible strategies during the negotiation. If we consider approaches originated in Game Theory, the available knowledge can be depicted by the payoff tables. Such knowledge is necessary in order to be able to determine the moves (offers) that are going to be proposed during negotiation. The lack of knowledge about opponents' types can be compensated by the IAs' ability to learn and verify the acquired information. This capability is the prerequisite for IAs to be able to adapt their behavior when the number and the types of opponents are changing. The same stands for the users preferences.

### 5.3 Negotiation as Distributed Constraint Satisfaction

Like in our research work, negotiation is studied as a process of competitive decision making between self-interested IAs in the presence of incomplete information. The IAs have limited information about the preferences and constraints of each other. They make decisions according to the available information about private preferences, constraints and individual negotiation strategies. The IAs exchange information in the form of offers. An offer is a complete solution which is currently preferred by an IA given its preferences, constraints and the negotiation history. An agreement takes place when a particular offer is accepted by all the negotiation parties. During the negotiation process, the range of possible offers of each party changes according to the available information. This range typically is reduced to final agreement. If the range becomes empty, a deal is not possible and the negotiation ends unsuccessfully. Therefore, negotiation is typically an iterative process of evaluating the offers, updating (e.g. reducing or expanding) the available options, and making the counteroffers according to the individual negotiation strategies. The objective is to find an instantiation of all variables that meets all the constraints at the same time. All other information related to preferences, constraints, offer evaluation and generation criteria of a particular IA are private and hidden from others.

### 5.4 E-Negotiation mechanism

Negotiation is one type of interaction in Electronic Marketplaces. IAs conducting negotiations should exhibit a number of functionalities in order to be able to adapt their behavior at every state of the world. These functionalities are discussed in the following list.

- Information's exchange.
- Coordination. IAs arrange their individual activities in a coherent manner. Collaboration. IAs could work together to achieve a common goal.

A typical example of negotiation is the bargaining, where two IAs (buyer IA and seller IA) exchange offers in an alternating manner (i.e. suggestions about how to exchange goods). This is done till one of them makes an offer that is acceptable by the other. We should notice that the buyer is endowed with money and has a specific valuation about the seller's good. A simple bargaining game usually embodies three moves: accept an offer; quit negotiation; and generate a counter proposal. Every IA can accept a proposal when it is in the area that the reservation prices indicate and the utility for the opponent's proposal is better than the expected utility taken from the upcoming rounds. Normally IA quits negotiation when the opponent proposal is out of the area that is indicated by the reservation prices and a given maximum negotiation time (deadline) is reached. In the rest case, the IA could generate a counter proposal that is offered to the opponent. A sophisticated negotiation could encompass additional moves, such as alternative or correlated products suggestion; and ultimatum.

Different mechanisms may have different properties, so the negotiation's mechanisms should have specific properties like:

1. Simplicity: requires less computational processing and communication overhead.
2. Efficiency: produces a good outcome. What is meant by `good,' however, may differ from one domain to another. One common criterion is Pareto optimality, where no IA could be better off in a different allocation without other IA being worse of it.
3. Distribution: does not involve a central decision maker. Centralization may lead to communication bottlenecks or decreasing reliability due to the single point of failure.
4. Symmetry: not being biased against some IA based on inappropriate criteria. Again, what constitutes "inappropriate" criteria depends on the discussed domain.

5. Stability: no IA has incentive to deviate from some agreed strategy or set of strategies. In an auction, for example, a negotiation mechanism may require that no IA lies by making a false bid, or that no group of IAs can form strategic coalitions to overcome other IAs.
6. Flexibility: leads to agreement even if IAs does not have complete and correct private information in relation to their own preferences. This feature requires a complementary mechanism for rational investigation and possible refinements of internal decisions during negotiation.

### 5.5 Types of E-negotiations

The main problem in negotiations is to decide how IAs will cooperate before they actually act to accomplish their goals. Each IA would like to reach some agreement that is as favorable as possible rather than disagreement.

Regarding to the number of participants, negotiations can be divided to bilateral or trilateral. A well-known type of negotiation is the bilateral, i.e. buyers and sellers mutually interact. Bilateral negotiation is usually concerned with multi issue contracts. A multi issue contract takes into account not only the product's price but also other important features like quality, delivery time, seller's trust and so on. Trilateral negotiations involve middle member as a third party. Bargaining is an example of bilateral negotiation. Auctioning are characterized as trilateral exchanges. More specific:

- <u>Bargaining- Buyer driven</u>: involves a buyer that negotiates with a seller until an acceptable agreement for both is reached. At first, the buyer searches for a seller, evaluates the products / services, and negotiates with the seller for an agreement. If the negotiation fails, the buyer searches repeatedly for other sellers until an agreement is made with one of them. IAs are suitable for negotiating on behalf of users, because they make a complete valuation of the value of products.
- <u>Bidding- Buyer Driven</u>: involves one buyer and several sellers. The buyer asks for bids and accordingly compares the offers she receives. The buyer chooses the best offer, i.e. the lowest offer that maximizes the buyer utility. IAs are suitable for sellers for initiating bids, accepting bids, comparing them, and notifying the winner. The contract-net [16] is among the well-known protocols that illustrate the bidding exchange.
- <u>Auctioning- Seller Driven:</u> involves one seller, several potential buyers, and a middle member. At first, the seller sets the lowest price of the product / service to be

auctioned. Through the middle member, the seller advertises the product / service and calls for auctions. Then, buyers make offers to the middle member. Finally, the middle member selects the buyer who makes the highest offer regarding the initial seller's offer, i.e. the offer that maximizes the seller's profit. IAs are suitable for finding the middle member, monitoring the offers of buyers, sending offers to the middle member, and following up the progress of the auction on behalf of buyers.

- Clearing- Middle member Driven: involves several buyers, several sellers, and one middle member - the broker. Sellers and buyers submit their requests to the broker regarding their needs. Next, the broker matches needs with offers. If there is a successful matching, the broker informs both buyers and sellers about the outcomes. IAs are suitable for finding the broker, monitoring the offers of buyers and sellers, and sending offers to the middle member.

In addition, E-negotiation can be classified according to the number of parties and the number of product issues. In terms of the involved parties, negotiation setting could be one-to-one, one-to-many or many-to-many. In such cases, we talk about concurrent negotiations. In terms of negotiation issues, a negotiation can involve single issue (e.g. price) or multiple issues (e.g. price, quality and delivery time). The problem of automating one-to-many negotiation has been proven to be hard [25]. This has led to the wide use of highly structured one-to-many negotiation models based on auctions. Various types of traditional auctions, forward and reverse, are being used such as English, Dutch, and Vickery [26]. Although these negotiation models proved to be efficient and easily implementable in online applications, due to their simplicity, and although they fulfill the business needs in certain kinds of scenarios, they fail to support scenarios in which more complex, less structured negotiations occur. They follow a bidding style which considers competitive offers between participants flowing in one direction. A significant limitation of auction based negotiation systems is that they do not allow for interactive negotiation based on exchanging offers and counter offers, and, thus, exploiting the flow of information in both directions. With interactive negotiation, more information can be exchanged, and more flexible negotiation strategies become possible. Moreover, having less structured negotiation rules means exercising different strategies with different opponents becomes possible, contrary to the case of auctions.

Researchers are interested in concurrent negotiation since:

- it is both time efficient and robust when an agent need to negotiate with multiple other agents to make a good deal, and,

- it is essential when an agent requests a service involved multiple agents like supply chain problem.

Most of recent works focused on one-to-many negotiation. The first apparent technique towards automating flexible one-to-many negotiations is designing a complex, IA, which stores information about all current simultaneous negotiations at once in its state. However, this approach has a number of disadvantages. First, from a software engineering point of view, this approach poses a scalability problem, since our IA runs on one machine and may face problems trying to conduct an increasing number of concurrent negotiations. Secondly, adding or removing underlying one-to-one negotiation strategies requires rebuilding the IA. So this cannot be done at run time. There is an apparent need for more flexible, scalable, and reusable component based one-to-many negotiation.

A simple extension for one-to-one is one-to-many negotiation, which defines to reuse the techniques and components that are used in one-to-one negotiations. This offers an advantage over models in which one single complex IA must conduct and directly maintain multiple threads of negotiation. An IA can negotiate with others by creating a number of one-to-one negotiating IAs that negotiate on its behalf, and perform the task of coordinating them (sub-negotiators). Every sub-negotiator conducts a one-to-one negotiation with a different opponent. After each negotiation cycle (one offer and counteroffer), each sub-negotiator reports the results back to the coordinating IA. The coordinating IA then evaluates the situation, and issues instructions accordingly. The buyer IA consists of a coordinating component and a number of sub-negotiators (threads). All threads represent the preferences and constraints of the same buyer, but they may use different negotiation strategies. Similarly, a single selling IA can negotiate with a number of prospective buyers by instantiating a number of sub-sellers, and coordinating them.

### 5.6 Strategy of Automated Negotiations

Strategy selection depends on the negotiator's objectives, preferences, and risk attitude. For a strategy to be effective it has to lead to a solution, which the negotiators' counterparts accept. This means that when constructing the set of possible strategies the counterparts' profile have to be considered. The question that arises is: "How to learn about partners profile in order to devise the adequate strategy?". Usually there are two kinds of information available, that is, the history of previous interactions and the behavior of an opponent during the current negotiation. There is no universal method for handling the available data to learn about opponents' preferences in order to derive optimal moves during the negotiation process. Researchers apply and test various models of data acquisition and inference [12]. Tree types of different groups with different frameworks can be found in literature:

1. <u>Decision making by explicitly reasoning about the opponent's behavior</u>: IAs in this group explicitly reason about their opponent's objectives and behaviors. IAs decide what is the appropriate response to their likely behavior. In this respect, non-cooperative game theory (which is particularly concerned with providing equilibrium strategies in which no IA wants to change its strategy whatever its opponents do) is an interesting approach for analyzing strategic interactions among IAs. Another approach utilizes a Bayesian network that is used for updating the knowledge and beliefs that each IA has about the environment and other IAs. In this approach, offers and counteroffers between IAs are generated based on Bayesian probabilities.
2. <u>Decision making by finding the current best solution</u>: Algorithms in this group focus on finding the offer that maximizes the IA's utility given the IA's issues such as constraints, preferences, current negotiation situation, and the opponent's last offer. Luo [13] develop a fuzzy constraint-based framework for multiple negotiations in competitive trading environments and demonstrate it in a negotiation between a real estate agency and a buyer. Kowalczyk and Bui [14] also use fuzzy constraints to model multiple negotiations, but in their approach the negotiation takes place on individual solutions one at a time. Faratin [15] et al. develop a suite of algorithms for multiple negotiations that covers both concessionary behavior and trade-offs aiming to find a win-win solution for both parties.
3. <u>Argumentation</u>: In the argumentation-based approach, IAs exchange additional information over and above the basic terms and conditions of the contract. This information can be of a number of different forms. Nevertheless, it is always some form of argument which justifies the position of the IA making the argument. Thus, in addition to rejecting a proposal, an IA can offer a critique of the proposal, explaining why it is not accepted (e.g., the price is too high). The way in which argumentation fits into the general negotiation process was defined and a simple negotiation protocol for trading proposals was enhanced by a series of moves which allow the passing of arguments. Moreover, the strategies can be classified according to the operation exercised by the IAs into:

(1) strategies exercised by individual buyer or seller IAs in their one-to-one encounters, and

(2) strategies exercised by the coordinating IAs in organizing and issuing commands to their sub-negotiators. Negotiation strategies of individual sub-negotiators could be identified to:

- Desperate Strategy: This is a very simple strategy in which the time constraints may be important and the IA wants to close a deal fast. In this strategy, as soon as a sub-negotiator finds an acceptable offer, the coordinating IA accepts it and sends messages to the remaining to terminate their negotiation. If more than one sub-negotiator comes up with an acceptable offer, the one with the highest utility is chosen while the rest are terminated.
- Patient Strategy: In this strategy, even if an acceptable deal is found by one or more negotiator(s), those IAs are asked to wait while all other IAs are asked to resume their negotiations. Once all negotiators complete their negotiation process (whether with success or failure), the best offer is chosen. This strategy guarantees that the best possible deal can be reached, but does not pay attention to time constraints. This might be a significant limitation in a marketplace with too many potential suppliers to negotiate with. One variation of the patient strategy is one in which a time limit is set by the user, within if no better deal was found, the negotiation terminates and the best deal so far wins.
- Optimized Patient Strategy: In this strategy, the coordinating IA uses information about one negotiation outcome to influence the performance of other negotiators. The constraints on the utility for the other negotiators are updated in order to avoid unnecessary deals which are not as good as the one already achieved. For example, if the accepted minimum total utility is 5 Monetary Units, and a sub-negotiator has agreed with utility of 7 Monetary Units, there is no point for other sub-negotiators reporting back a deal with utility of 6 Monetary Units even though it is an acceptable deal (according to the initial constraints). In this case, the constraint on the utility for all remaining sub-negotiators is updated to be equal to 7 Monetary Units, causing any deal below to be unacceptable. This also ensures that no sub-negotiator will provide an offer that is worse than the offer received by a fellow sub-negotiator
- Strategy Manipulation Strategies: In this class of strategies, the coordinating IA may modify the negotiation strategies of different negotiators at runtime. For

example, after securing a deal, other negotiators can exercise a take-it-or-leave-it strategy with their opponents.

# 6. GENERAL DESCRIPTION OF MULTI- FEATURE CONCURRENT NEGOTIATIONS

## 6.1 Scenario Description-Concurrent Negotiation with multiple issues

In our research work we are trying to model real life negotiations. In our scenario, one buyer has direct negotiations with a number of sellers. Each buyer creates a number of sub-buyers to interact concurrently with a different seller (one-to-many negotiations). The IAs in both parts have no information about the preferences of their opponents (limited knowledge). Moreover, there is no need for a central decision maker which collects the information of IAs and retransmits it to all IAs. Every seller has a product in his possession with a specific cost and tries to sell it with the highest possible profit. Similarly, the buyer is interested in purchasing a product that is closer to his own preferences and has willingness to pay a certain amount of money. Each product has a number of issues that increase or decrease each player's utility. Issues are categorized as detailed below:

1. Proportional or inversely proportional to the utility: For example, the greater will be the seller's trust, the greater will be the buyer's utility, because trust is a proportional issue, which means that he is negotiating with a trustful seller. The delivery is inversely proportional with the utility, because the greater will the delivery time, the smaller will be the utility, as there is a delay.
2. Negotiable: The value of some issues is negotiable, like the delivery time, because the seller can modify it at will. The issues, that cannot be negotiated, reflect certain state such as trust. The fact that a seller is trustworthy or untrustworthy does not change during a negotiation, and the only effect is to increase or decrease the utility of the buyer.

For example a buyer would like to buy a car with the following characteristics:

- The maximum can be up to 10,000, but also preferably less than 9,000€
- The more warranty has the better, but over 5 years makes no difference
- The delivery time should be 2 to 15 September; preferable time is between 7th and 13th, less preferably after 13th and even less before 7th

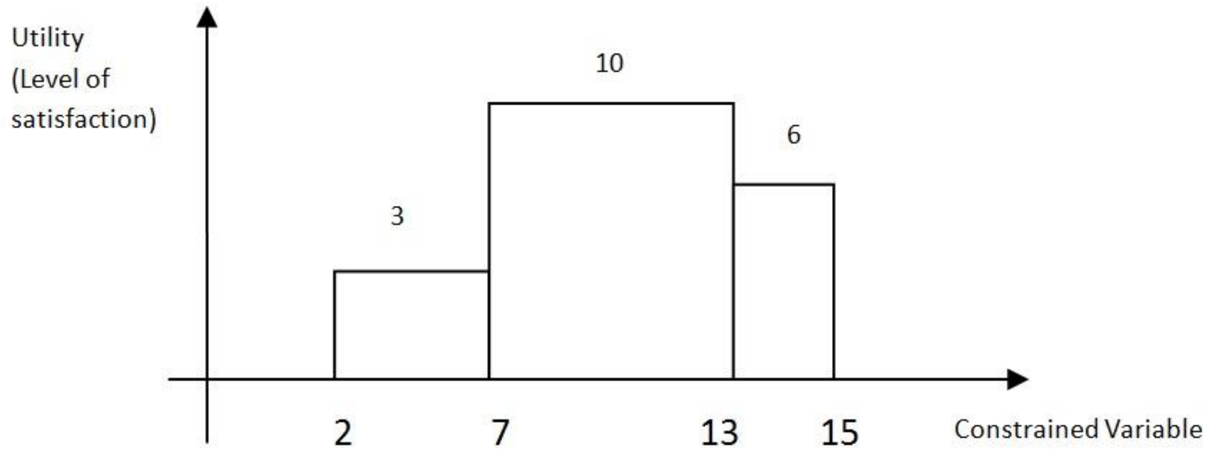


Figure 9: Relation between utility and delivery time

So for different user restrictions for each of the issues of the product, the utility changes dynamically.

At the start of negotiation, each buyer produces a parent thread and a number of threads, each of whom is represented by a thread. Each thread negotiates with a different and unique seller trying to close the 'best' deal. The best agreement is defined, as we mentioned earlier, as the agreement that maximizes the utility of the buyer. The buyer then has 2 parts:

- A parent thread that is responsible for producing a number of thread’ threads
- A set of threads that are responsible to negotiate with individual sellers; each thread follows a different strategy.

It is worth noting that the parent thread does not function as a coordinator of negotiation. Its main function is to start the threads and to inform them in case of agreement combined with information like agreement’s price, delivery time, seller’s trust etc. depending on user preferences. Each thread exchanges a number of offers with each seller for the desired product and its characteristics. Each desired product is characterized as a set of user preferences expressed as product issues. An example of a set of issues for a book could be:

1. Price
2. Delivery time
3. Quality of Service (QoS)
4. Seller’s trust

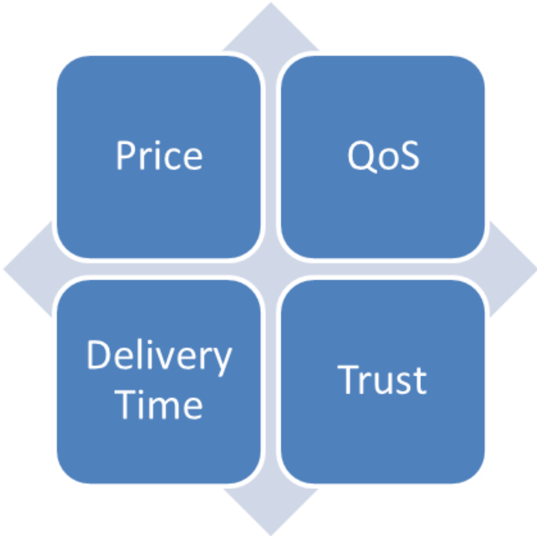


Figure 10: Set of product

We consider absolute lack of knowledge about the seller's and buyer's characteristics (valuation, product cost, deadlines etc) as well as the strategy followed by everyone (patient, impatient, etc). In each round, each player suggests bids for the set of the product in order to increase his personal utility. The buyer or the seller having received an offer, calculates its relative utility and then can accept the offer or reject it with making a counter offer. The negotiation process ends after a finite number of rounds. If the negotiation ends, because it has reached the maximum number of rounds, then we assume that the negotiation ends with a conflict.

The buyer having lack of knowledge of the seller and his characteristics should approximate the optimal deal. But how do we evaluate two or more deals with different issues/sets? How long should a thread wait for a seller when another thread has, already, closed an agreement? Our approach to answer these questions is to modify the weights of issues. When a thread closes a deal with a seller then a message is broadcasted to all other threads that need to reassess the utility function and the weights to each issue. According to the agreement which was closed by the thread tries to emphasize those issues that disadvantage against the agreement while being negotiable.

For example, suppose a thread closes a deal with a buyer who is more reliable than the $\text{thread}_i$, where i is between 1 and maximum number of thread ($N_{threads}$), at a lower price than that is negotiated by the $\text{thread}_i$ like in Table 3.

Table 3: Example of thread values and agreement values

| Feature's Name | $Thread_i$ Value | Value of agreement |
|---|---|---|
| Price | 60 | 44 |
| Trust | 0.6 | 0.5 |
| Delivery | 5 | 6 |
| QOS | 0.6 | 0.8 |

The utility of $thread_i$ is calculated as:

$$U_i = \sum_{j=1}^{m} w_j \circ v_j, \begin{cases} 0 \leq i \leq N_{sub-buyers} \\ 0 \leq j \leq m_{attributes} \end{cases} \quad (1)$$

where w and v are referred as weights and value respectively of each issue in a set of product. If the weight of the issue, which has a greater value than the relevant issue of the agreement, decreases, then its proportion in the utility will decrease too. So we emphasize on those issues that have value "worse" than the agreement's value, because the utility will increase owing to these "bad" issues. In our example the $thread_i$ should focus on price and QoS rather than trust and delivery time.

In this way there is a change in strategy dynamically in conditions of lack of knowledge not only for the seller but also for the thread, which closed deal. This resolves the problem with multiple IAs and without the need for knowledge sharing through coordination. There are similar jobs dealing multi-issue [30], [31], where knowledge is distributed to every thread, so all threads know and estimate their situation in comparison to others. The problem in our approach lies in choosing the right method for evaluating the weights and the user utility. In our experiments we used the Heuristic method, the Simplex, the method of Analytic Hierarchy Process and the particle swarm theory. The description these methods with the way they are used in our experiments.

### 6.2 Related Work

The current approach intends to reach to the best agreement under dynamical changes on the thread's strategy. Threads make decisions in an environment in which the negotiations can be distinguished to the following categories:

1. **One-to-many**: the buyer creates a number of threads to negotiate with individual sellers
2. **Concurrent:** each buyer thread negotiates independently with the rest buyer's threads
3. **Dynamic**: each thread changes the adopted strategy whenever needed
4. **Limited Knowledge**: buyer's threads have no knowledge about the opponent's (seller's) characteristics and limited knowledge about the rest buyer's threads.

The problem is to define an algorithm that can deal with all the above categories in order to reach an optimal solution for the buyer. Current research efforts mainly handle only a subset of the above categories. Authors in [42], [48] try to solve the one-to-one negotiation problem while researchers in [39], [40], [44], [46] use a coordinator in order to change the strategies and to decide about the final agreement. In addition, there is no proposed mechanism that combines multi-issue with one-to-many negotiations.

In the paper [39], the researchers present an Intelligent Trading Agency (ITA), in which they try to achieve one-to-many negotiation by conducting a number of coordinated simultaneous one-to-one negotiations.  Like in our work, there is a number of sub-negotiators that negotiate on the buyer's behalf with a number of sellers.

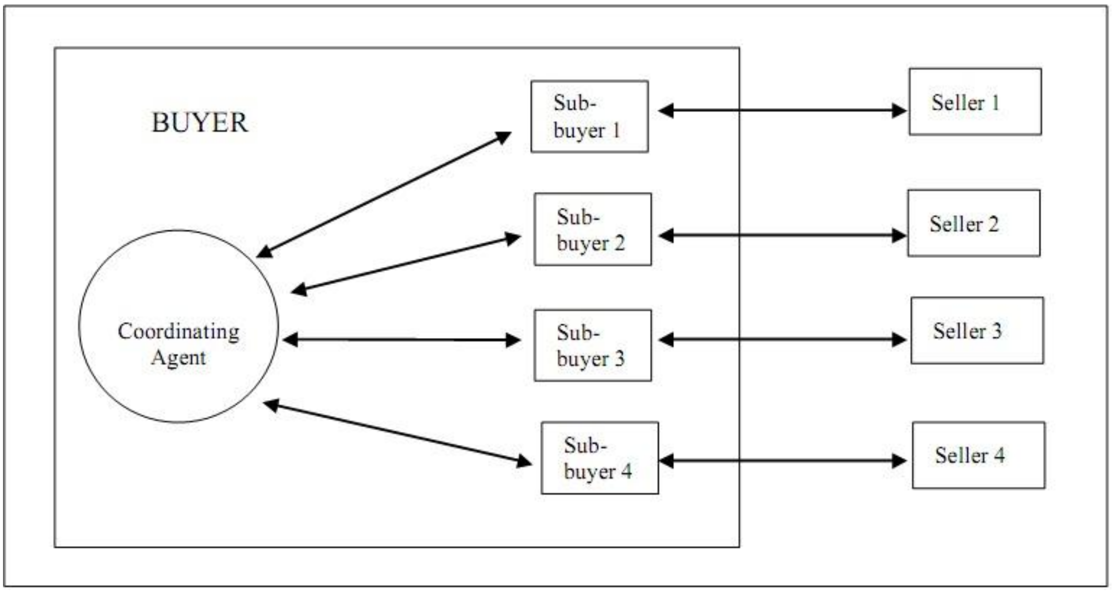


Figure 11: One-to-Many negotiation (One buyer & many sellers)

Necessary, however, prerequisite for ITA is the existence of a coordinator. After each negotiation circle, every IA reports back to the coordinator. The coordinator evaluates how well each IA has done and issues new instructions accordingly. Also its role is to inform when an agreement is reached and to change sub-negotiators' strategy. A drawback is that the coordinator issues commands to change the strategies not according to what is happening in the big picture. Also, the coordinator changes strategies based on a heuristic method and could be improved by reusing the negotiation experiences to improve the final outcomes.

In [40], a model for bilateral multi-issue negotiation is presented, where issues are negotiated sequentially. The issue studied is the optimal agenda for such a negotiation under both incomplete information and time constraints. However a central mediator is used and the issues all have continuous values. The effect of time on the negotiation equilibrium is the main feature studied, from both a game theoretic and empirical perspective. In earlier research [41] a slightly different model is proposed, but the focus of the research is still on time constraints and the effect of deadlines on the IAs' strategies. This contrast with our model, where the best buyer's utility of the outcome and not time is the main issue studied.

Research effort presented in [42] discusses a model for integrative, one-to-one negotiation in which the values across multiple issues are negotiated simultaneously. Similar to [42], we adopt the rationale of a distributed negotiation, which eliminates the need of a central planner. In [42], researchers take the heuristic approach and model IAs are able to jointly explore the space of possible outcomes with a limited (incomplete) information assumption. This is done through a trade-off mechanism, in which the IA selects the value of its next offer based on a similarity degree with previous bids of the opponent. In our design, we do not explicitly model tradeoffs, yet the same effect is achieved through the asymmetric concessions mechanism. In [42], the initial domain information for the issues consists of fuzzy value labels. A mechanism is used by every IA able to utilize any amount of information revealed by the negotiation partner. This is done due to the fact that IAs want to improve the efficiency of the agreements. The negotiation can be further improved by incorporating a "guessing" heuristic, by which an IA uses the history of the opponent's bids to predict his preferences. IAs take into consideration not only their own weights, but also those of the opponent in order to compute the next bid. Unlike our approach, where IAs are under no knowledge about the opponents, the IAs exchange partial preference information for the issues.

The issue of concurrent negotiation is dealt with in [44] and [46]. By considering the negotiation as a distributed constraint satisfaction problem, the authors in [44] represent a framework for one-to-many negotiation by conducting a number of coordinated concurrent one-to-one negotiations and discuss the possible negotiation strategies for a coordinator. However, many-to-many negotiation is not equivalent to multiple one-to-many negotiation and important issues arise such as consistency, coordination, and de-commitment risk. Such problems are too difficult to be handled by the existing protocols.

Nguyen et al. [46] present a heuristic model for coordinating concurrent negotiations and an integrated commitment model that enable agents to reason about when to commit or de-commit. In their model, a coordinator manages several negotiation threads one for each individual seller. The buyer first selects a strategy for the threads based on her preferences, then classifies the sellers according to their behaviors and consequently adapts the right negotiation strategy based on their classification. Once a thread reaches a deal with a particular seller, the deal is a one-side commitment binding to the corresponding seller and can only be dropped after the buyer finalizes all negotiation threads. It is obviously biased in favor of the buyer since a commitment should be a bilateral relationship used to connect two participating agents. To mitigate this problem, it is allowed in seller to de-commit by adopting another model. Both buyer and seller can renege from the previous deal after paying the de-commitment penalty. This model still biased in favor of the buyer since any sellers who have already reached a deal have to wait till all negotiation threads end. On the other hand, breaking the commitments is always a hard decision to make because it is usually more issues beyond de-commitment penalty that need to be considered such as reputation, user feedback, etc.

Sandholm and Lesser [48] discuss the automated negotiation among bounded rational self-interested agents in the context of task allocation domain. A protocol is presented to support commitments by introducing the counter-proposal into CNP. Zhang et al. [49] present a negotiation mechanism for task allocation in a cooperative system. By using two-dimension binary search, agents compromise between their initial proposals and current proposal to generate new proposal and reach an agreement if the marginal gain is more than marginal cost.

The solution we propose in the current thesis is presented in the following section. Specifically, Section 6.3 shows the model we adopt while Chapter 7 describes the several algorithms that have been developed and tested.

### 6.3 Multi Issue Negotiation Modeling

We investigate the case where buyers have direct interactions with sellers and utilize a number of threads for having the optimal possible agreement. The negotiation process involves a number of alternating offers. Every thread has a specific deadline posed by its owner ($T_b$) and the same stands for the seller ($T_s$). In each negotiation (between threads and sellers), the seller starts first and the thread follows if the proposed offer is rejected. The seller proposes an offer at odd rounds, i.e., 2*n*+1, *n* = 0, 1, 2, …, and the thread issues a counter offer at even rounds, 2*n*, *n*=1, 2, …. If a player is not satisfied with the proposed offer, it has the right to reject it and issue a counter-proposal. Every offer involves specific values for the examined issues. This approach is defined as the package deal [30][31].

The seller has in her own property a number of products and she wants to make the most profitable agreement. The seller utilizes a specific utility function ($U_S$) defined as follows:

$$\mathrm{U_S} = \sum_{i=1}^{m} w_i \cdot v_i \qquad (2)$$

where m is the number of issues, $w_i$ and $v_i$ are the weights and values for each one respectively. Moreover, she has a specific deadline for non-zero profit ($T_S$).

Both players have their own strategy for offers calculation. We adopt the approach described in [54] [55]. Each entity has its own reservation values for every issue. We consider an interval [$\min_i$, $\max_i$] where every issue takes its values. These values differ in the buyer as well as in the seller side. Both entities generate their offers based on the following equations:

$$\mathrm{O_i} = \min_i + \phi(\mathrm{t}) \cdot (\max_\mathrm{i} - \min_i) \qquad (3)$$

in the buyer side, and,

$$\mathrm{O_i} = \min_i + (1 - \phi(\mathrm{t})) \cdot (\max_\mathrm{i} - \min_i) \qquad (4)$$

in the seller side. In the above defined equations, $O_i$ depicts the next offer for issue i. As we can see, our model involves a time dependent strategy that is depicted by the function φ(t). This function is defined as follows:

$$\varphi(\mathrm{t}) = \mathrm{k} + (1\text{-}\mathrm{k}) \cdot \left(\frac{\mathrm{t}}{\mathrm{T}}\right)^{\frac{1}{\psi}}, \ \mathrm{T} = \{\mathrm{T_b} \text{ or } \mathrm{T_s}\} \qquad (5)$$

where $k \in [0,1]$, $t=1,2,\ldots, \min(T_b,T_s)$, and $\psi$ is a positive random strategy parameter.

For every issue, we calculate the corresponding utility (actually this is a part of the total utility) based on the following equations:

$$U(v_i) = \frac{v_i - \min_i}{\max_i - \min_i} \qquad (6)$$

if the specific issue is proportional, and,

$$U(v_i) = \frac{\max_i - v_i}{\max_i - \min_i} \qquad (7)$$

if the issue is not proportional.

# 7. METHODS FOR STRATEGY CONFIGURATION

As mentioned above, the optimization of the user's utility is to estimate the weights of the issues in the utility calculation process. Each thread tries to emphasize on those issues with the worst value compared to the corresponding issues of the agreement. A change in the weights will result a change in the utility function. The value of the utility will increase if and only if the seller will improve those values which the thread falls short compared to the agreement. In this thesis, we propose an automated process of calculating the weights for each issue based on the following: Heuristic, Simplex and Analytic Hierarchy Process. Furthermore, we propose the use of the Particle Swarm Optimization (PSO) algorithm combined with the Virtual Forces algorithm, in which new buyer's offers are created based on the IAs movement. In general, the N issues correspond to an N-dimensional space, where threads move and try to find the optimal solution.

## 7.1 Heuristic Method

The proposed methodology aims to enforce the buyer to converge to an optimal solution through a set of rules for calculating the discussed weights. The optimal solution is represented by the offer with the maximum utility. Each thread can compare its own state (seller offers, etc.) with the agreement state. Weights are going to be changed when an agreement is reported by another thread. Actually, weights are defined again when the final utility is smaller than the utility calculated based on the agreement information. Parameters taken into consideration for weights definition are the issues defined at the beginning of the negotiation (e.g. price, trust, QoS, etc).

The algorithm starts with a comparison between the values of issues of $thread_i$ and the values of issues of the agreement. Each issue then is characterized as an issue that needs a change or not. We represent every issue that needs to be changed with variable $I_C$ and issues that do not need any change in their values with $\bar{I}_C$. The following algorithm depicts the discussed classification process.

**Algorithm** Classification

Inputs: issues,

Outputs: $I_C$ , $\bar{I}_C$

```
    initializeVariables()
    Foreach issue T_i
        If issue.isProportional()   Then
            If  T_i.getValue() < A_i. getValue() Then
                I_C .add(T_i)
            Else
                Ī_C .add(T_i)
            EndIf
        Else
            If  T_i.getValue() > A_i. getValue() Then
                I_C .add(T_i)
            Else
                Ī_C .add(T_i)
            EndIf
        EndIf
    EndFor
End
```

Listing 1: Heuristic Algorithm for classification

In the above described algorithm, the parameter $T_i$ represents the $i^{th}$ issue while $A_i$ represents the agreement in another thread. After that, in order to pay attention on the weights that should be changed (under the rationale that such an approach will lead to a better utility value) we take that:

$$\sum I_{ci} > \sum \bar{I}_{ci} \quad (8)$$

where $I_{ci}$ is the issue weight of every issue in $I_c$ and $\bar{I}_{ci}$ is the weight of every issue in $\bar{I}_C$. Thus, we aim to enhance the effect of the modified issues in the final utility value and take higher results. Additionally, such an alteration indicates a strategy where the buyer pays more attention on the modified issues in order to achieve better agreement with another seller. An interesting extension in the above described algorithm could be the additional support of one or more issues. For example, we could define a little bit increased weight in very important issues like price or the seller trust.

After the weights determination, the thread continues the negotiation with the specific seller until: a) an agreement is true, b) a conflict happens, or, c) another agreement message arrives. Through our algorithm each seller negotiating with every thread will be forced to give better prices leading us to a better deal.

### 7.2 The Simplex Method

The Simplex method is a process for solving linear programming problems. It is a very efficient method used for solving demanding problems. In particular, it is a systematic procedure, which is repeated until the optimal result is reached. The algorithm includes rules to start the process and criteria that determine when it will end. Siskos, in [34], reports on the rationale of the method Simplex, that is:

*This is a walk on the tops of solutions of a hyper-polyhedron A of linear programming improving with every step (peak to peak) the value of the objective function z*

Because hyper-polyhedron A is a convex set, this walk will stop at that peak where there is no improvement, i.e., an optimal solution of linear programming [34].

Therefore, the Simplex method aims to the optimal solution based on a set of conditions. The optimal solution is reached based on the systematic development of the key solutions and optimal control. With this iterative process, the algorithm is capable of computing feasible solutions in a systematic way. The method is based on two concepts:

• The concept of a *feasible solution*, which is the solution (values for the decision variables) for which all of the constraints are satisfied, and,

• The concept of *optimal solution*, in which the objective function reaches the maximum or the minimum.

Simplex's property defines that the optimal solution of a linear programming problem, if any, can always be found in one of the basic feasible solutions [32]. The Simplex method examines the value of an objective function only at the end points of the region of feasible solutions, with a systematic algebraic way. The sequential examination of the maximum / minimum points is performed by an iterative manner, i.e., the same set of procedures and algebraic operations in successive steps are repeated until the optimal solution is reached. Each step of the Simplex method corresponds to selecting a maximum / minimum point of the region of feasible solutions. At each new step the next endpoint in the region is chosen in such a way that the value of the objective function increases (if we try to maximize the objective function) or decreases (if we try to minimize the objective function) and, thus, gradually gets closer to the optimal solution.

The first step in Simplex method is to find a basic feasible solution. After that, the solution is tested for optimization in terms of the objective function and the effect of the input. The input is a non-basic variable used for replacing at least one of the key variables that we already have defined as a solution. If there is an improvement by using the specific input, this gives a new feasible solution. In this point, it should be noted that another important feature of this method is that for every new solution the value of the objective function is at least as optimal as the previous solution. This results in each step of the iterative process to move closer to the optimal solution. Finally, the algorithm defines specific conditions that determine when the optimal solution is met.

Table 4: Advantages and Disadvantages of Simplex Method

| Advantages | Disadvantages |
|---|---|
| 1. *Easy to program:* Any function can be quickly adapted in a software program.<br>2. *Easy to use:* allows an individual to address problems with more than two decision variables.<br>3. Algorithm does not require a derivative function and a relevant orthogonality condition. | 1. *Limited applications:* It only applies in situations where a decimal quantity is appropriate.<br>2. The method is efficient with a few variables. Many problems with real-life practical interest have hundreds of variables.<br>3. *Difficult requirements:* Only problems that can be expressed in a standard form with specific conditions can be solved. |

In our case, we try to implement the Revised Simplex to reassess the weights of the issues. The following is a *linear programming problem* in standard form:

$$\text{Maximize } \sum_{j=1}^{n} c_j x_j \text{ , s.t. } \begin{cases} \sum_{j=1}^{n} a_{ij} x_j \le b_i & (i = 1,2,\ldots,m) \\ x_j \ge 0 & (j = 1,2,\ldots,n) \end{cases} \tag{9}$$

After introducing the variables $x_{n+1}$, $x_{n+2}$,…, $x_{n+m}$ the problem can be written as:

$$\text{Maximize } \sum_{j=1}^{n} c_j x_j \text{ , s.t. } \begin{cases} \sum_{j=1}^{n} a_{ij} x_j = b_i & (i = 1,2,\ldots,m) \\ x_j \ge 0 & (j = 1,2,\ldots,n) \end{cases} \tag{10}$$

*Or in matrix notation:*

$$\text{Maximize } \mathbf{CX} \text{ , s.t. } \begin{cases} \mathbf{AX} = \mathbf{B} \\ \mathbf{X} \ge \mathbf{0} \end{cases} \tag{11}$$

The matrix A has m rows and n+m columns with the last m columns forming an identity matrix. The vector x is of length n+m, and the column b is of length m. The linear programming problem can be solved by the well-known *revised simplex method* [37]. A basic feasible solution $X^*$ partitions X into $X_B$ (m basic variables) and $X_N$ (n non basic variables). This corresponds to the partition of matrix A into $A_B$ and $A_N$, and C into $C_B$ and $C_N$. Each iteration of the discussed method [37] can be described as follows:

1. Solve the system $\mathbf{y} \circ \mathbf{A_B} = \mathbf{C_B}$
2. Choose any column α of $A_N$ such that $\mathbf{y}a$ is less than the corresponding component of $C_N$. If such column doesn't exist, then the current solution is optimal.
3. Solve the system $\mathbf{A_B} \circ \mathbf{B} = \mathbf{A}$
4. Find the largest d such that $\mathbf{X_B}^* - \mathbf{dB} \ge \mathbf{0}$. If no such d is found then the problem is unbounded, otherwise at least one component of $\mathbf{X_B}^* - \mathbf{dB}$ will be equal to zero and the corresponding variable leaves the basis.
5. Set the entering variable to d. Replace the values of the basic variables. Replace the leaving column $A_B$ by the entering column and replace the leaving variable by the entering variable.
6. If such column does not exist, then the current solution is optimal.

Listing 2: Pseudo-code of Revised Simplex Method

There are two main advantages of the revised simplex method over the common simplex method.

1. At every iteration, fewer entities are needed. Recall that in the regular simplex method total entries required is (m+1)X(n+1), whereas in the revised simplex method mXn entries are made.
2. The revised simplex method generates the inverse of the current basis matrix automatically.

In our scenario, the weights correspond to the specific issues and the variables to the weights of the issues. So the optimal equation is the following:

$$\text{Maximize} \sum_{j=1}^{n} o_j w_j \ , \text{ s.t. } \begin{cases} \sum_{j=1}^{n} w_j = 1 \\ \sum_{j=1}^{n} o_j w_j \geq U_{agreement} \\ w_j \geq 0 \end{cases} \quad (j = 1,2,\ldots,n) \qquad (12)$$

The necessary condition for solving the problem (set for maximizing the utility function) is that the utility is calculated to be greater than the utility of the agreement. The definition of weights will lead us to a better position for the buyer (in utility terms). If the function does not have a feasible solution, then the weights remain unchanged.

### 7.3 Analytic Hierarchy Process – AHP

The Analytic Hierarchy Process (AHP) has been developed by Saaty [36] and is one of the best known and most widely used methodologies in multi-criteria problems. It allows users to assess the relative weight of multiple criteria or multiple options against given criteria in an intuitive manner. In case that quantitative ratings are not available, policy makers or assessors can still recognize whether one criterion is more important than another. Therefore, pair wise comparisons are appealing to users. Saaty established a consistent way of converting such pair wise comparisons (e.g., X is more important than Y) into a set of numbers representing the relative priority of each of the criteria. The basic process to carry out the AHP consists of the following steps:

***1. Structuring a decision problem and selection of criteria***

The first step is to decompose a decision problem into parts. In the simplest form, this structure comprises a goal or focus at the topmost level, criteria (and sub criteria) at the intermediate levels, while the lowest level contains the options. Arranging all the components in a hierarchy provides an overall view of the complex relationships and helps the decision maker to assess whether the elements in each level are of the same magnitude so that they can be compared accurately. An element in a given level does not have to function as a criterion for all the elements in the level below. Each level may represent a different aspect of the problem so the hierarchy does not need to be complete [36]. When constructing hierarchies it is essential to consider the environment surrounding the problem and to identify the issues or issues that contribute to the solution. Moreover, it is very important to identify all participants associated with the problem.

***2. Priority setting of the criteria by pair wise comparison (weighing)***

For each pair of criteria, the decision maker is required to respond to a question such as: *"how important is criterion A relative to criterion B"*. Rating the relative "priority" of the criteria is done by assigning a weight between 1 (equal importance) and 9 (extreme importance) to the more important criterion, whereas the reciprocal of this value is assigned to the other criterion in the pair. The weights are then normalized and averaged in order to obtain an average weight for each criterion.

Remark

*Reciprocal: if activity (item) i has a specific numerical rating with respect to activity j, then j has the reciprocal value when compared to i*.

The pairwise comparison information for each component of the problem is represented by a pair wise comparison matrix. If there are n items that need to be compared for a given matrix, then a total of $n \cdot (n-1)/2$ judgments are needed. There are two reasons for this apparent savings in the required number of judgments. First, since any alternative is equally preferred to itself, the diagonal of the matrix is filled by 1. Second, the corresponding positions below the diagonals are the reciprocals of the judgments already inserted. For example, assuming that the pair wise of quality to delivery is 3, it follows that the pairwise comparison of delivery to quality is 1/3.

***3. Pair wise comparison of options on each criterion (scoring):***

For each pair the better option is awarded a score, again, on a scale between 1 (equally good) and 9 (absolutely better), whilst the other option is to assign a rating equal to the reciprocal of this value. Each score records how well option "X" meets criterion "Y". Afterwards, the ratings are normalized and averaged. Comparisons of elements in pairs require that they are homogeneous or close with respect to the common issue; otherwise significant errors may be introduced into the process of measurement [38].

### *4. Obtaining an overall relative score for each option*

In the final step, the option scores are combined with the criterion weights to produce an overall score for each option. The extent to which the options satisfy the criteria is weighted according to the relative importance of the criteria. This is done by simple weighted summation. All elements and priorities as a whole produce the final judgments. Sometimes, after the above computation, the less important elements can be dropped from further consideration, because of their relatively small impact on the overall objective. The priorities can then be recalculated throughout, either with or without changing the judgments [38]. The advantages and the disadvantages of AHP are depicted in Table 5:

Table 5: Advantages and Disadvantages of AHP Method

| **Advantages** | **Disadvantages** |
| --- | --- |
| 1. AHP can take into consideration the relative priorities of factors or alternatives and it represents the best alternative.<br>2. AHP provides a simple and very flexible model for a given problem.<br>3. AHP provides an easily applicable decision making methodology that assists the decision maker to precisely decide the judgments.<br>4. Either objective or subjective considerations with quantitative information play an important role during the decision process.<br>5. Any level of details about the main focus can be listed or structured in this method. This way, the overview of the main focus of the problem can be represented very easily. | 1. There is not always a solution to the linear equations.<br>2. The computational cost is tremendous even for a small problem.<br>3. AHP allows only triangular fuzzy numbers to be used.<br>4. AHP is based on both probability and possibility measures.<br>5. Rank reversal fact should be considered carefully during the application. It defines the changes of the judgment alternatives order when a new one is added to the problem. Validity of the rank reversal is still discussed in the literature.<br>6. AHP has a subjective nature of the modeling process. This means that the methodology cannot guarantee the decisions as definitely true. |

| | |
|---|---|
| 6. AHP has very wide range of usage like planning, effectiveness, benefit and risk analysis, choosing any kind of decision among alternatives.<br>7. AHP relies on the judgments; so the main focus of the problem can be evaluated easily from different aspects.<br>8. Decision maker can analyze the elasticity of the final decision by applying the sensitivity analysis.<br>9. It is possible to measure the consistency of the decision maker's judgments.<br>10. Computer software help decision makers to apply AHP fast and precisely. | 7. When the number of the levels in hierarchy increases, the number of pair comparisons also increases, so that to create the AHP model takes much more time and effort. |

The comparison matrix defined by Saaty employs 1-9 scales. The discussed scales are illustrated with the following comparison matrix in Figure 12 and Table 6 respectively.

$$A=\begin{bmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & . & . & . & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & . & . & . & \frac{w_2}{w_n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & . & . & . & \frac{w_n}{w_n} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & . & . & . & a_{1n} \\ a_{21} & a_{22} & . & . & . & a_{2n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ a_{n1} & a_{n2} & . & . & . & a_{nn} \end{bmatrix}$$

Figure 12: Saaty's comparison matrix

Table 6: Saaty's Scale

| Saaty's Scale | The relative importance of two sub-elements |
|---|---|
| 1 | Equally important |
| 3 | Moderately important with one over another |
| 5 | Strongly important |
| 7 | Very strongly important |
| 9 | Extremely important |
| 2, 4, 6 ,8 | Intermediate values |

However, the data in our problem should be redefined as compared with those of the agreement. How do we categorize features without prior knowledge? These features should self-compute the significance they have, while expressed in different units. Below we analyze some features of an issue and their values presented in Table7 such as:

1. Minimum price can be negotiated (Min).

2. Maximum price that can be negotiated (Max).

3. Negotiable, a Boolean value that determines whether or not the issue is negotiable.

4. Proportionate, Boolean value that specifies that the utility increases or inversely proportionate to the value of utility.

5. Value, the current trading value.

Table7: Example of a set of product

| Feature's Name | Min | Max | Negotiable | Proportionate | Value |
|---|---|---|---|---|---|
| Price | 10 | 100 | True | False | 60 |
| Trust | 0 | 1 | False | True | 0.6 |
| Delivery | 0 | 10 | True | False | 5 |
| Relevancy | 0 | 1 | True | True | 0.6 |

All these elements help to grade each issue on a scale from 0 to 1 on the basis of their importance in order to be expressed in the same units of measurement. The way in which we grade issues is the following:

1. We believe that the higher the final weight of a characteristic the more important it is (the algorithm for the weighs calculation is presented in Listing 3).

2. The thread mainly falls short on the important elements. By the term "falling short", we refer to all those key characteristics that are less efficient in terms of performance in comparison to those of the agreement.

Table 3 shows that the thread should achieve a lower price and a better QoS, if he / she wants to achieve a better deal than it already has. So the most important features are the price and the QoS because the others have achieved better values. It is worth noting that even though the trust is not negotiable, it is reflected in the categorization of weights. We propose an algorithm that has the following steps:

Algorithm: Weights Calculation

Input: Issue ($T_i$), AggreementOffer($A_i$)

Output: Updated weights

Begin

Foreach $weight_i$

$weight_i$ = ε; /*ε close to zero*/

EndFor

Foreach issue $T_i$

 If issue.isProportional() Then

  If $T_i$.getValue() > $A_i$. getValue() Then

   $T_i$.setWeight( $\frac{T_i.getValue() - A_i.getValue()}{T_i.getValue() - T_i.getMin()}$ )

  EndIf

 Else

  If $T_i$.getValue() < $A_i$. getValue() Then

   $T_i$.setWeight( $\frac{T_i.getValue() - A_i.getValue()}{T_i.getMax() - T_i.getValue()}$ )

```
            EndIf
        EndIf
EndFor
End
```

Listing 3: Pseudo-code of Computation of weights

Every weight is initialized to a very small number close to zero. Then the distance between thread's value and the value of those in the agreement is computed and it is normalized between 0 and 1. The greater the distance from the agreement's offer the greater the weight will be. A value close to 1 expresses the importance of the specific issue in the reassessing process of weights. If an issue has a better value than those in the agreement then the weight remains at the same value (close to zero).

As a next step, we utilize the approach of redefining the weight, which is stated on [35]. A new fuzzy comparison matrix figured in Figure 13 differs from Saaty's in that we use membership scales, instead of the 1-9 scales, as the values of the elements.

$$A=\begin{bmatrix} \frac{w_1}{w_1+w_1} & \frac{w_1}{w_1+w_2} & . & . & . & \frac{w_1}{w_1+w_n} \\ \frac{w_2}{w_2+w_1} & \frac{w_2}{w_2+w_2} & . & . & . & \frac{w_2}{w_2+w_n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \frac{w_n}{w_n+w_1} & \frac{w_n}{w_n+w_2} & . & . & . & \frac{w_n}{w_n+w_n} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & . & . & . & r_{1n} \\ r_{21} & r_{22} & . & . & . & r_{2n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ r_{n1} & r_{n2} & . & . & . & r_{nn} \end{bmatrix}$$

Figure 13- Fuzzy comparison matrix

If this comparison matrix is consistent, it should satisfy:

$$r_{ii} = 0.5,\ r_{ij} + r_{ji} = 1,\ \frac{1}{r_{ij}} - 1 = \left(\frac{1}{r_{ik}} - 1\right) \cdot \left(\frac{1}{r_{ki}} - 1\right) \quad (13)$$

This method compares weights in pairs and is more straightforward and easier to use for the decision-makers. The meanings of our membership scales can also be expressed in the same way as Saaty's scale in Table 8.

Table 8: Scale for fuzzy pair-wise comparison

| **Saaty's Scale** | **Scale Values** | **The relative importance of two sub-elements** |
|---|---|---|
| 1 | [0.5, 0.6] | Equally important |
| 3 | [0.6, 0.7] | Moderately important with one over another |
| 5 | [0.7, 0.8] | Strongly important |
| 7 | [0.8, 0.9] | Very strongly important |
| 9 | [0.9, 1.0] | Extremely important |
| 2, 4, 6 ,8 | [0,0.5) | Intermediate Values |

## 7.4 Particle swarm optimization (PSO) based on Virtual Force Algorithm

### 7.4.1 Particle swarm Optimization (PSO)

Particle swarm optimization (PSO) is a computational method developed by Kennedy and Eberhant [36] that optimizes a problem by iteratively trying to improve a candidate solution with respect to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here named particles, and moving these particles around in the search-space according to simple mathematical formulations over the particle's position and velocity. Let $x_i(t)$ denote the position of particle i in the search space at time step t; unless otherwise stated, t denotes discrete time steps. The position of the particle is changed by adding a velocity, $v_i(t)$, to the current position, i.e.,

$$x_i(t+1) = x_i(t) + v_i(t+1) \qquad (14)$$

Each particle's movement is influenced by its local best $P_i$ known position and is also guided toward the best known positions $P_g$ in the search-space. $P_g$ positions are updated as better positions found by other particles. The pseudo-code of the PSO is shown below in Listing 4:

```
Algorithm PSO
Input :         Particles'  Position (Xi) and Velocity (Vi)
Output:         Final Best position of particles
Begin
Foreach particle
        Initialize Xi and Vi
EndFor
Do
        Foreach particle
                Calculate fitness value
                If the current fitness value is better than Pi Then
                        Update Pi
                EndIf
                Choose the particle position with the best fitness value of all the neighbors
                as the Pg
        EndFor
        Foreach particle
                Update Vi
                Update Xi
        End
While maximum iteration or ideal fitness is not attained
End
```

Listing 4: Pseudo-code for PSO algorithm

In short, the advantages of the PSO algorithm are depicted by the following list:

1. PSO have no overlapping and mutation calculation. The search can be carried out by the speed of the particle. During the development of several iterations, only the most optimist particle can transmit information to the other particles.
2. The calculation in PSO is simple. Compared with the other algorithms, PSO can be completed easily in terms of computational complexity.
3. In PSO the number of the dimensions is equal to the variables of the problem.

Additionally, the disadvantages of the PSO algorithm are the following:

1. The method suffers from the partial optimism, which causes the decrease of the accuracy in the directions of particles.
2. The method cannot work with problems of scattering and optimization.
3. The method cannot work with problems of non-coordinate system, such as the solution to the energy field and the moving rules of the particles in the energy field.

### 7.4.2 Virtual Force Algorithm

The VFA algorithm is inspired by disk packing theory [51] and the virtual force field concept from robotics [52]. VFA was developed to be used in Wireless Sensor Networks (WSN). For a given number of sensors, VFA attempts to maximize the sensor field coverage using a combination of attractive and repulsive forces. The sensor field is represented by a two-dimensional grid. The dimensions of the grid provide a measure of the sensor field. The granularity of the grid, i.e., distance between grid points can be adjusted to trade off computation time of the VFA algorithm with the effectiveness of the coverage measure.

The key idea behind this algorithm is that every sensor $s_i$ is subject to positive and negative forces $F_i$ due to other sensors, obstacles and areas of preferential coverage in the grid. Note that $F_i$ is a vector whose orientation is determined by the vector sum of all the forces acting on $s_i$. This virtual force model creates a convenient method to model obstacles (negative forces) and the need for preferential coverage (positive forces). Sensor deployment should take into account the nature of the terrain, e.g., obstacles such as buildings and trees in the line of sight for infrared sensors, uneven surface and elevations for hilly terrain, etc. In addition, based on relative measures of security needs and tactical importance, certain areas of the grid need to be covered with greater certainty. For example, consider four sensors $s_1$, $s_2$, $s_3$ and $s_4$. The force $F_1$ is given as the vector sum of the forces $F_{12}$, $F_{13}$ and $F_{14}$, like in Equation 8.

$$F_1 = F_{12} + F_{13} + F_{14} \qquad (15)$$

The force $F_{12}$ is the force between the sensor $s_1$ and sensor $s_2$, $F_{13}$ is the force between $s_1$ and $s_3$ and $F_{14}$ is the force between $s_1$ and $s_4$. The $F_{ij}$, in general, is expressed as the force between the sensor $s_i$ and $s_j$ in polar coordinate notation. Note that $\bar{f} = (r, \theta)$ implies a magnitude of r and orientation θ for vector $\bar{f}$

$$F_{ij} = \begin{cases} (w_A (d_{ij} - d_{th}), a_{ij}) & \text{if } d_{ij} > d_{th} \\ 0 & \text{if } d_{ij} = d_{th} \\ w_R \dfrac{1}{d_{ij}}, a_{ij} + \pi & \text{if otherwise} \end{cases} \qquad (16)$$

where $d_{ij}$ is the Euclidean distance between sensor $s_i$ and $s_j$, $d_{th}$ is the threshold on the distance $s_i$ and $s_j$, $a_{ij}$ is the orientation (angle) of a line segment from $s_i$ to $s_j$ and $w_A$($w_R$) is the measure of the attractive (repulsive) force. The threshold distance $d_{th}$ controls the way that close sensors get to each other. In our example, if it assumed that $d_{12} > d_{th}$, $d_{13} < d_{th}$ and $d_{14} = d_{th}$, then the $F_{12}$ is an attractive force, $F_{13}$ is a repulsive force and $F_{14}$ is zero, as shown in Figure 14.

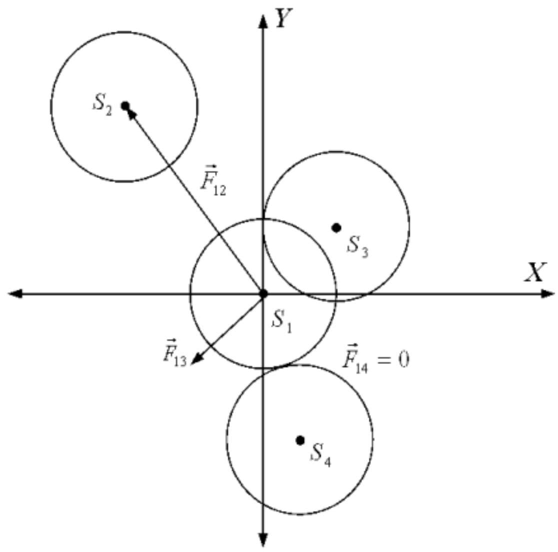


Figure 14: An example of virtual forces with four sensors [8]

```
Algorithm Virtual Force
Input :     d(s_i, P), c_th, d_th, α, β
Output:     Final proper locations of sensor nodes

Begin

loops = 0
MaxLoops = MAX_LOOPS
While (loops < MAX_LOOPS)
/*coverage evaluation */
    For P(x,y)  in Grid, x ∈ [1, width], y ∈ [1, height]
            For s_i ∈ {s_1, s_2, ….., s_k}
                    Calculate c_xy(s_i, P) using (d(s_i, P), c_th, d_th, α, β)
            EndFor
            If coverage requirements are met Then
                    Break from While loop
            End if
    EndFor

    For s_i ∈ {s_1, s_2, ….., s_k}
            Calculate F_ij
            Calculate F_iA
            Calculate F_iR
            F_i = ΣF_ij + F_iR + F_iA,  j ∈ [1,k], j ≠ i
    EndFor

    For s_i ∈ {s_1, s_2, ….., s_k}
            F_i(s_i) virtually moves s_i to its next position
    EndFor
    loops=loops+1
EndWhile

End
```

Listing 5: Pseudocode of the VFA algorithm [8].

In our approach, we are assigning sensors in particles.

### 7.4.3 PSO- VFA ALGORITHM

In our approach, we combine the described algorithms, e. g. the PSO with the VFA. According to the PSO theory, we try to model each $thread_i$ as an individual $particle_i$. Each $particle_i$ has a position $x_i$ (t) at time step t, which represents the offer proposed to seller, and the velocity $v_i(t)$. The next position $x_i(t+1)$ is calculated, as shown in Equation 14.

In addition, each $particle_i$ corresponds to a set of issues and, because of this, we need a method to define the space, in which each $particle_i$ will move from $x_i(t)$ to $x_i(t+1)$. Let us have N issues, we can set our $particle_i$'s move in N-dimensional space. As described in VFA, every issue can create a force $F_i$ on $particle_i$, as presented in Figure 15

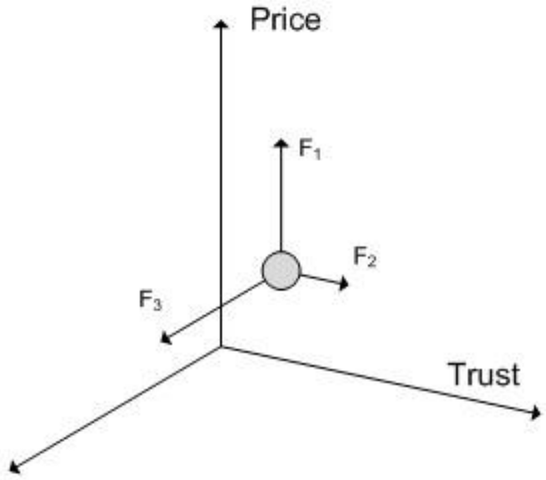


Figure 15- Forces on a particle

The vector sum of all the forces acting on $particle_i$ creates a vector $FV_i$. The combination of $FV_{i,}$ $particle_i$'s global best position ($P_{gi}$) and $particle_i$'s local best position ($P_{li}$) will move the particle to its next position $x_i(t+1)$, as shown in Figure 16

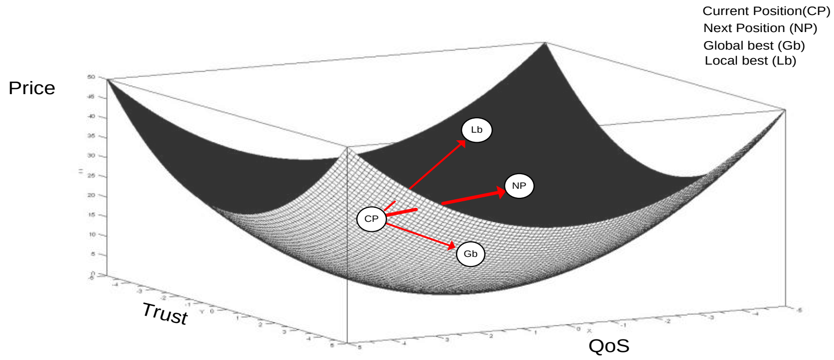


Figure 16: Particle's move in N-dimensional space

Hence the next position xi(t+1) depends from the velocity vi(t), which is equal to

$$v_i(t) = c_1 \times (P_{li} - x_i(t)) + c_2 \times (P_{gi} - x_i(t)) \qquad (17)$$

where $x_i(t) = BuyerOffer_i(t)$ and $c_1$, $c_2$ are random generated values.

# 8. EXPERIMENTS

In this thesis, we run a large number of negotiations for different values of basic parameters. At first, we define the metrics used for the evaluation of our model and accordingly we analytically present our results.

## 8.1 Performance Metrics

The performance metrics defined for our model are the following:

- ***The agreement ratio (AG)***: The AG parameter indicates the number of negotiations that end with an agreement (successful negotiation) out of R negotiations. The greater the AG is the greater the profit of the players becomes. In a successful negotiation, the buyer reaches her goals (she finally buys the specific product) while the seller gains some profit from the purchase action. Thus, the final aim of our experiments is to have AG → 1. We formulate the discussed metric as the following equation indicates:

$$AG = \frac{|SN|}{R} \quad (18)$$

  In the above equation, |SN| indicates the number of the successful negotiations. It stands |SN| ≤ R. It should be noted that a successful negotiation is held in time $t^* \leq \min(T_b, T_s)$ where $T_b$ and $T_s$ are the buyer (the threads used from a specific buyer have the same deadline $T_b$) and the seller deadline respectively.

- ***Average Buyer Utility (ABU)***: Let us define, the final utility that a buyer gains from a negotiation. This utility can be depicted by the variable $U_F$ and is defined as follows:

$$U_F = \max(U_i) \quad (19)$$

  where $U_i$ is defined by the equation (1) and calculates the utility of every thread. The buyer makes the most profitable agreement and, thus, she gains the maximum utility taken from all the agreements. Thus, the parameter ABU is defined by the following equation:

$$ABU = \frac{\sum_{k=1}^{|SN|} U_{F_k}}{|SN|} \quad (20)$$

When ABU → 1 means that the agreements are very profitable for the buyer side. In such cases, the utility for every product that the buyer is going to buy is close to the optimal (value equal to 1). We should not forget that $U_i$ is calculated as a weighted sum of the issues values and, thus, these values are optimal.

- ***Average Seller utility (ASU)***: In our experiments, we try to reveal the efficiency of the proposed model for the seller side as well. For this reason, we define the ASU metric that depicts the performance of our model based on the utility earned by the seller. We should mention that the discussed utility is calculated by using the issues values defined in the agreement. In the seller side, we define that every issue has the same proportional weight. For example, if we have four issues, each one has a weight equal to 0.25. Based on the above discussion, we define the utility of the seller as follows:

$$ASU = \frac{\sum_{k=1}^{|SN|} U_{S_k}}{|SN|} \qquad (21)$$

- ***Average Rounds (AR)***: AR is defined as the number of rounds needed to reach an agreement. The number of rounds to the agreement AR indicates the proportion of steps that entities need in order to reach an agreement out of the full horizon $T=\min(T_b,T_s)$. We consider that an agreement will be true in a time step $t^* \leq \min(T_b,T_s)$. Thus,

$$AR = \frac{t^*}{\min(T_b, T_s)} \qquad (22)$$

If AR → 0 means that the negotiation ends at first rounds while a value of AR →1 means that negotiation ends in a round close to the deadlines expiration.

- ***Number of successful thread ($P_t$)***: If we depict with H the number of threads in a specific buyer and with |H| the number of the successful threads (threads that complete the negotiation with agreement), we define the average number of successful threads as

$$SH = \frac{|H|}{H} \qquad (23)$$

Following this rationale we define the parameter $P_t$ as

$$P_t = \frac{\sum_{k=1}^{R} SH_k}{R} \quad (24)$$

When $P_t \rightarrow 1$ means that the majority of threads conclude the negotiation with an agreement and, thus, the buyer has many opportunities to select the most profitable agreement.

- Fairness (F) : Let us consider a specific negotiation between a buyer thread and a seller. We focus on the product price and try to reveal the fairness of our model concerning the discussed issue. The buyer has a specific valuation (V) about the product and the seller has a specific cost (c). It is proved [56], [57] that a price equal to $\frac{V+c}{2}$ gives the theoretic maximum utility for both entities. The fairness of the model can be defined based on the following equation:

$$F = 2 \cdot \frac{|p^* - \frac{V+c}{2}|}{V-c} \quad (25)$$

where $p^*$ is the product price defined in the set of agreement. When $p_A \rightarrow \frac{V+c}{2}$ then $F \rightarrow 0$ and $p^*$ is profitable for both. If $p^* \rightarrow V$ or $p^* \rightarrow c$ then the $F \rightarrow 1$. In order to have a fair agreement for both we should take $F \rightarrow 0$.

## 8.2 Performance Assessment

### 8.2.1 General Definition of parameters

At the beginning of the negotiation, the buyer ($thread_i$) and the sellers define the basic parameter values. These parameters are divided (i) in parameters that characterize the negotiation (negotiation's parameters) and (ii) parameters that define the issues.

The negotiation parameters are the following:

- ***V***: Price upper limit.
- ***$N_T$***: Number of threads for each buyer, which negotiate concurrently with the sellers.
- ***I:*** Number of issues that are negotiated between every thread and sellers.

The issues parameters, which are all defined randomly in the constructor of each thread, are:

- **[min, max]**: Issue value interval.
- ***Value****:* the first value of each issue in the interval [min,max].
- ***Proportionate***: true or false.
- ***Weight***: the initial value is equal to $\frac{1}{N}$, where N is the number of issues.

In our experiments, we examine three scenarios changing the negotiation parameters. In the first one, we change the buyer's valuation for a specific number of threads and issues. In the second, the variable that is changed is the number of threads with fixed valuation and number of issues. Finally in the third set of experiments, the number of issues is the variable which is studied.

### 8.2.2 Weights Evaluation

Before we begin with the discussion of different set of experiments, we would like to present snapshots of the three methods for the redefinition of weights (Heuristic, Simplex and VFA). To produce the certain snapshots, we have defined the basic negotiation parameters as following:

1. V=100 units
2. $N_T$=100
3. I=4 (1st case) and I=16 (2nd case)

The snapshots have been created by a random thread (t), with $0 \le t \le N_T$, at a randomly selected round (r), $0 \le r \le T_b$. The estimated weight values for 4 issues are depicted by Table 9 and Figure 17 and the correspondingly weight values for 16 issues by Table 10 and Figure 18.

Table 9: Evaluation of 4 issues

| Name | Heuristic | Simplex | AHP |
|---|---|---|---|
| **Issue1/ Price** | 0,3 | 0,98 | 0,59 |
| **Issue2/ Trust** | 0,4 | 0,01 | 0,23 |
| **Issue3** | 0,1 | 0,00001 | 0,14 |
| **Issue4** | 0,2 | 0,01 | 0,05 |

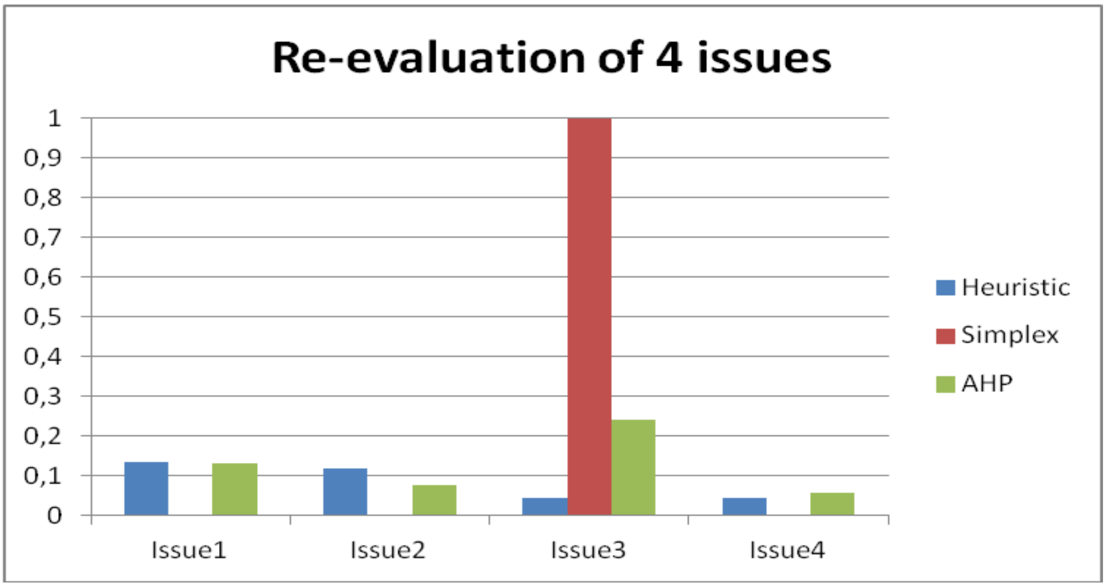


Figure 17: Evaluation of 4 issues

Table 10: Evaluation of 16 issues

| Name | Heuristic | Simplex | AHP |
|---|---|---|---|
| **Issue1/ Price** | 0.13333 | 0.00001 | 0.13048 |
| **Issue2/ Trust** | 0.12000 | 0.00001 | 0.07646 |
| **Issue3** | 0.04444 | 0.99985 | 0.24057 |
| **Issue4** | 0.04444 | 0.00001 | 0.05788 |
| **Issue5** | 0.06000 | 0.00001 | 0.05162 |
| **Issue6** | 0.06000 | 0.00001 | 0.03681 |
| **Issue7** | 0.04444 | 0.00001 | 0.10582 |

| | | | |
|---|---|---|---|
| **Issue8** | 0.06000 | 0.00001 | 0.02908 |
| **Issue9** | 0.06000 | 0.00001 | 0.07330 |
| **Issue10** | 0.06000 | 0.00001 | 0.10616 |
| **Issue11** | 0.04444 | 0.00001 | 0.01787 |
| **Issue12** | 0.04444 | 0.00001 | 0.01458 |
| **Issue13** | 0.06000 | 0.00001 | 0.01191 |
| **Issue14** | 0.04444 | 0.00001 | 0.00971 |
| **Issue15** | 0.06000 | 0.00001 | 0.00787 |
| **Issue16** | 0.06000 | 0.00001 | 0.02987 |

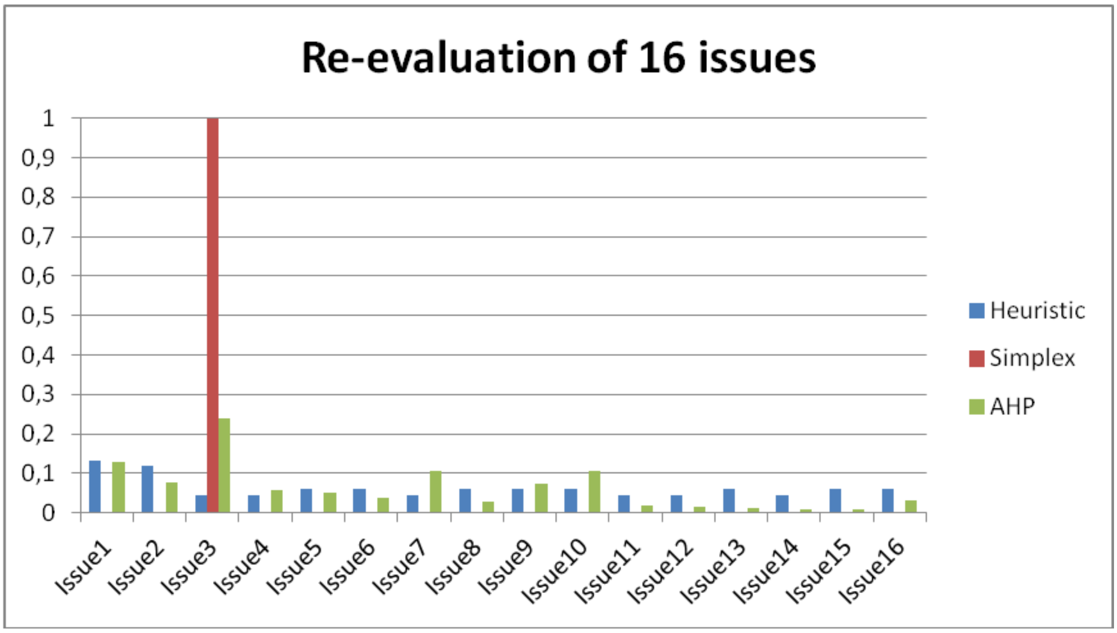


Figure 18: Evaluation of 16 issues

We can notice that AHP and Heuristic methods distribute evenly the weights to the different issues. However, Simplex method selects a specific issue, which is the output of the algorithm, in which the new estimated weight →1. Simplex in other words try to fully optimize a single issue. Even if AHP gives high value to the same issue, that Simplex selects, AHP distributes the weights more evenly.

### 8.2.3 Description of 1st Set of experiments

In the 1st set of experiments, we present comparison results for a large number (450.000) of negotiations between buyer and seller. Let us define MU as Monetary Unit. We run a number (300) of negotiations for $N_T$ = 50, I = 4 and V in the interval [10, 300]

MUs (we increase the V value by 10 MUs at every set of experiments). Seller's cost is randomly selected in the interval [10, 50].

In Figure 19, we depict the AG metric described by equation 18. We can notice that Simplex and AHP methods have AG → 1.0. Heuristic method is close to the AHP and Simplex methods while PSO reaches an average AG equal to 0.9. Simplex's AG values vary in the interval [0.93, 1.0] and this the reason why we consider it as the method with the most successful negotiations.

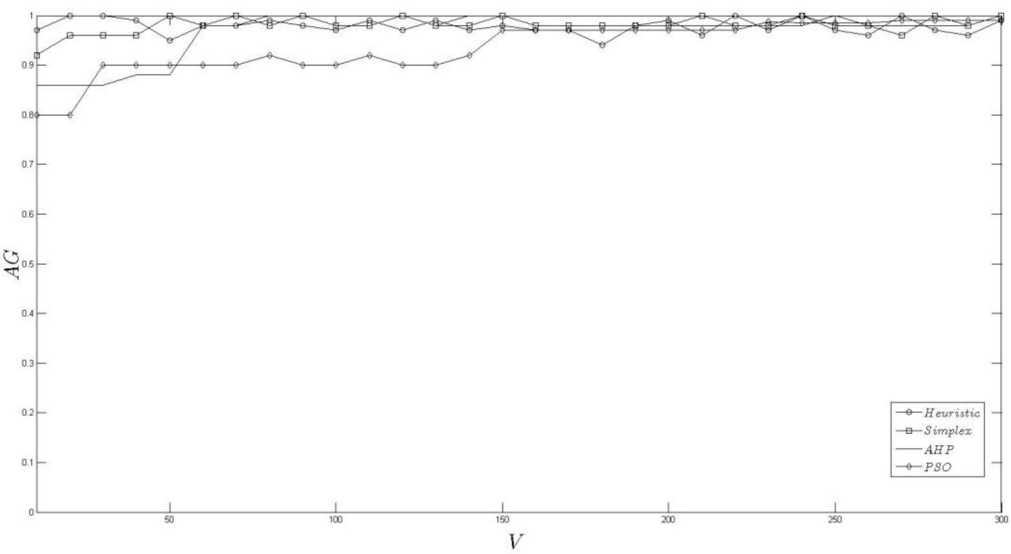


Figure 19: AG in 1st set of experiments

Figure 20 shows the ABU gained by the purchase in relation to the increase of V. Heuristic method has the lowest values. Simplex reaches the maximum ABU. AHP and PSO start from a small value of ABU, which increases gradually. Especially, the PSO starts from a minimum close to 0.45 when V=10 and reaches the Simplex's Value of ABU when V=300. PSO's particles, as long as the distance between buyer's valuation and seller's cost increases, have more space to move and finally to converge to an optimum value.

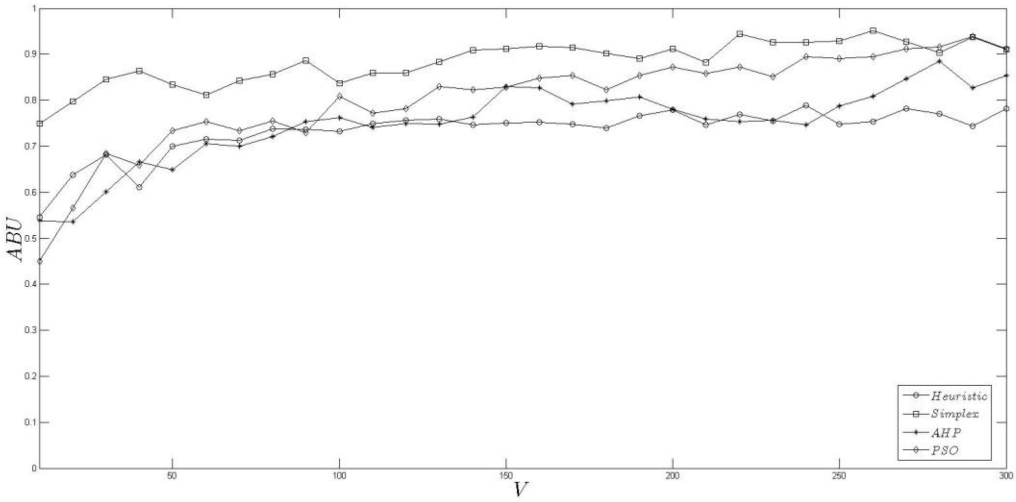


Figure 20: ABU in 1st set of experiments

ASU metric is presented in Figure 21. It can be observed that AHP and Simplex methods have constant values ranging near to 0.3. The heuristic method gradually increases the prices between the range 0.3 and 0.5. Notably the AHP method, like ABU, starts from a minimum and gradually reaches a maximum close to 0.5. The AHP describes realistic the negotiation mechanism from the seller's perspective, where the seller's utility should be low, when the difference (d) between price and cost is small, and increases as the d increases.

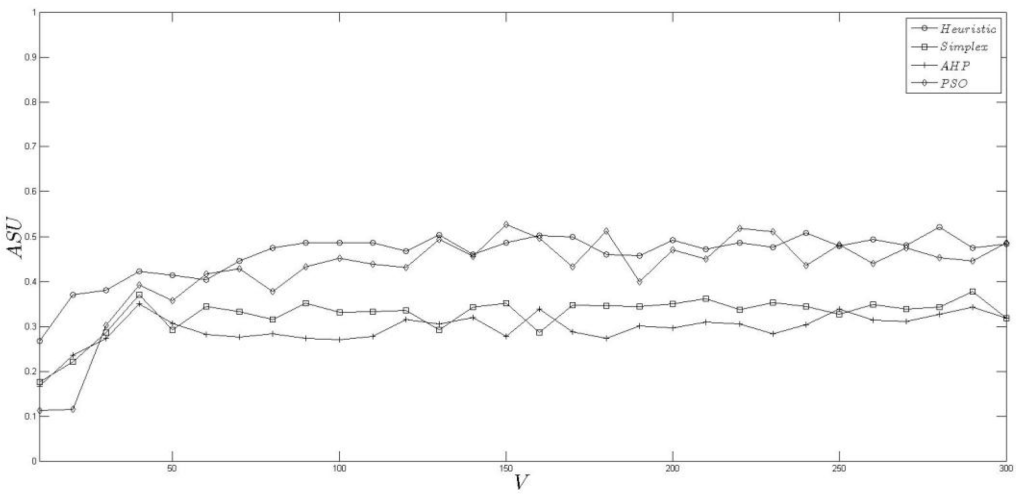


Figure 21: ASU in 1st set of experiments

Figure 22 depicts the average rounds (AR) required to reach an agreement. We can notice that all methods require several rounds to reach an agreement in the interval V [10, 50]. The above stand because V is close to the seller's cost. Once V≥ 50, all the methods have a sharp decrease of AR and converge to a specific value and can be evaluated in ascending order, in which Heuristic method requires the fewest rounds followed by Simplex, AHP and PSO.

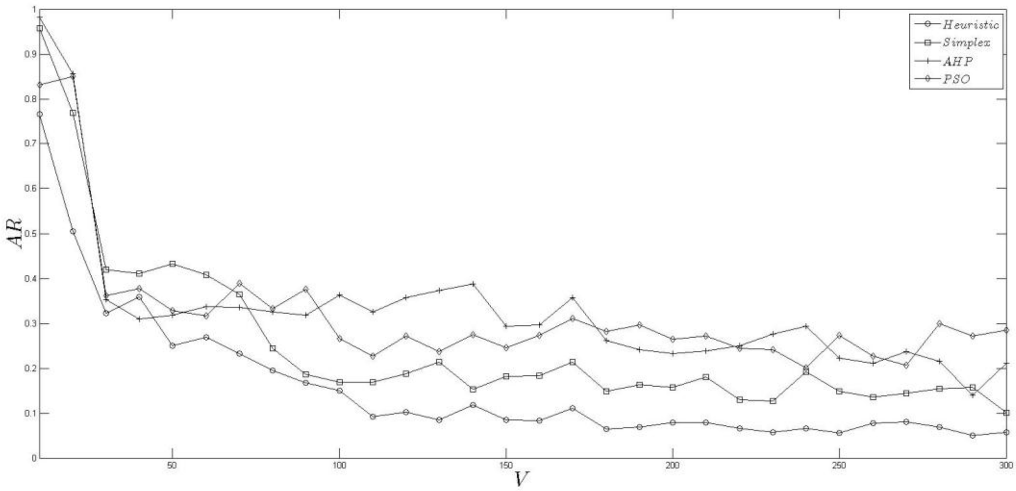


Figure 22: AR in 1st set of experiments

Regarding the proportion of the successful threads ($P_t$) presented in Figure 23, we can notice that the highest proportion of $P_t$ corresponds to Simplex method followed by AHP and Heuristic method. In the case of PSO, the values, even if they are increasing, do not overcome the threshold of 0.8. Pt values are independent over the threshold for V=50 for all methods.

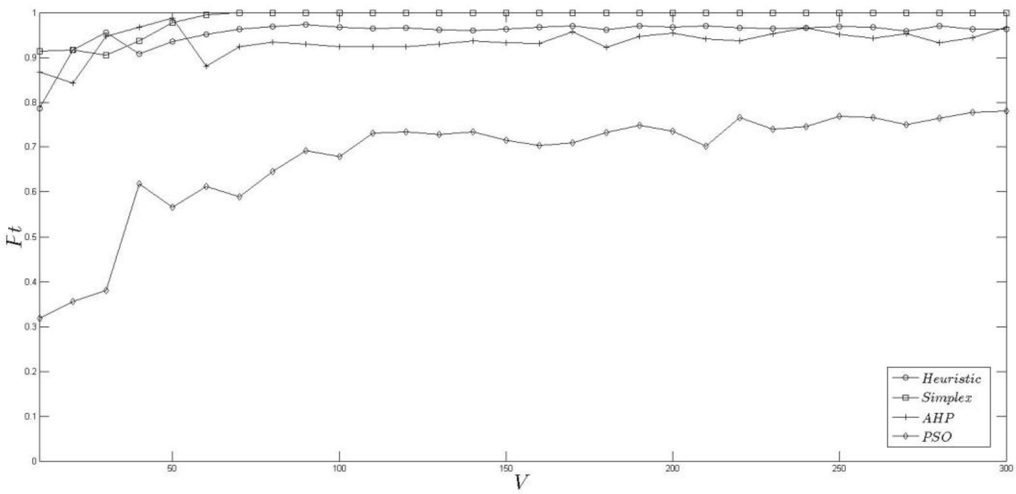


Figure 23: Pt in 1st set of experiments

Finally, we present the fairness of each method according to the increase of V. We can assess the methods based on the assumption that a method is fair if and only if F → 0. Hence it is depicted by Figure 24, that the fairest method is the Heuristic followed by AHP, Simplex and PSO (least fair). The three methods (Heuristic, Simplex and AHP) use the weights configuration function and “force” the seller to propose a better offer based on their preferences. So these methods, while they lean on buyer’s side, are characterized as “not fair” methods by default.

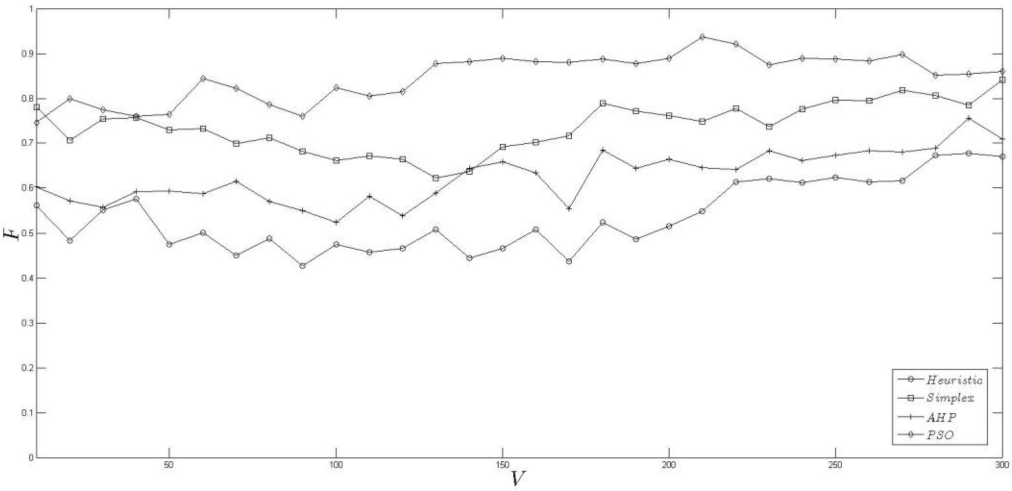


Figure 24: F in 1st set of experiments

### 8.2.4 Description of 2nd Set of experiments

In the 2nd set of experiment we are trying to study the changes in the metrics (AG, ABU, ASU, AR, $P_t$, F) while we are increasing the number of issues. The static input in the experiments are the V (V = 100) and the number of threads ($N_T$ = 50). Issues are calculated based on the equation $I=2^k$, where k=2,…,5. The metrics for each experiment are calculated as the output of 300 negotiations.

Generally speaking, the greater the number of issues the more complicated is our problem. The new issues raise in other words the overhead for the threads in the negotiation. When the utility is based on a greater number of issues, its final value equivalently decreases as a whole. Simplex and AHP algorithms need to create an array, which is proportionate to the number of issues. The bigger is the array the bigger is the complexity of AHP and Simplex and the lower is their performance. Contrary to AHP and Simplex, Heuristic can handle a big number of features because of the low algorithm’s complexity. Moreover, the new issues with their information provide to PSO enough space for particles to move and to converge.

The metric AG shown in Figure 25 presents the behavior of four methods compared with the increased number of issues. Although for I=4, the figure is similar to the 1st scenario, in which the Simplex achieves the best prices, the observations change with the increase of the number of issues. It is observed that all the methods' values gradually increase. Especially, the PSO→1 for a number of issues greater than 16. PSO having more information achieves convergence of all particles to the best possible agreement. AHP and Simplex due to algorithm's overhead and the generated table achieve the lower values in comparison to the others. The Heuristic method, because it (1) allocates the equivalent weights in both categories (changing and not-changing issues) and (2) has low complexity, increases the value of AG gradually.

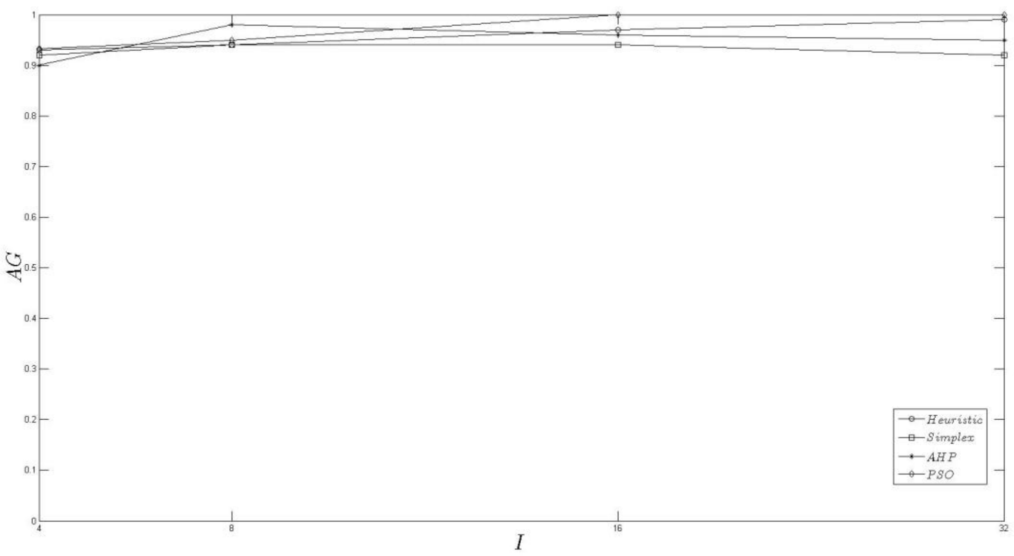


Figure 25: AG in 2nd set of experiments

Figure 26 and Figure 27 show a similar behavior of metrics ABU and ASU. All four methods as the number of issues increases, the utility gained by the buyers and sellers decreases. This reduction, which is not very sharp, is caused by the new issues, which are added in the equation (2) and multiplied with their weights. In both figures, the methods (AHP and Simplex) correspond to lower ABU and ASU values than the two others because of the large functions' overhead. PSO achieves the highest average utility, something expected as explained above.

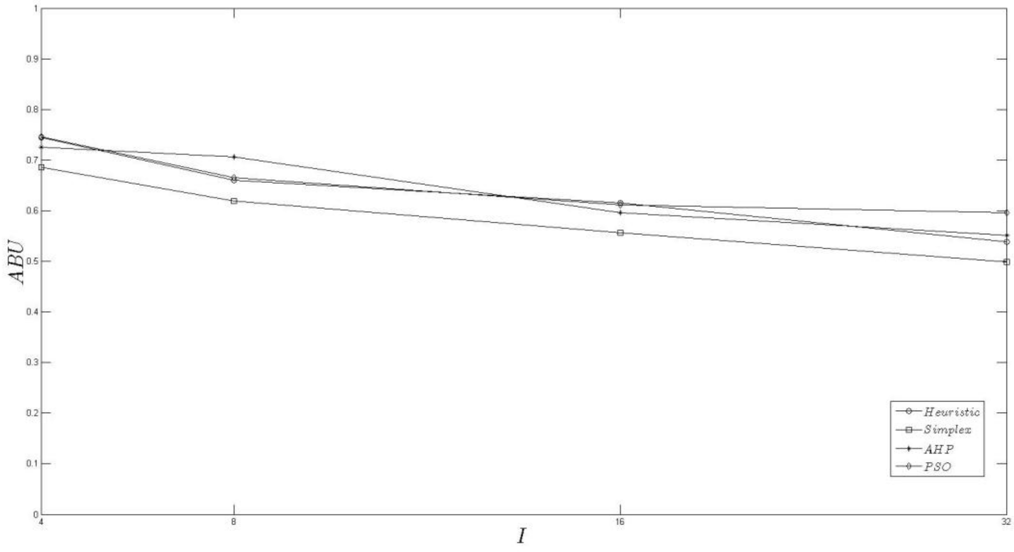


Figure 26: ABU in 2nd set of experiments

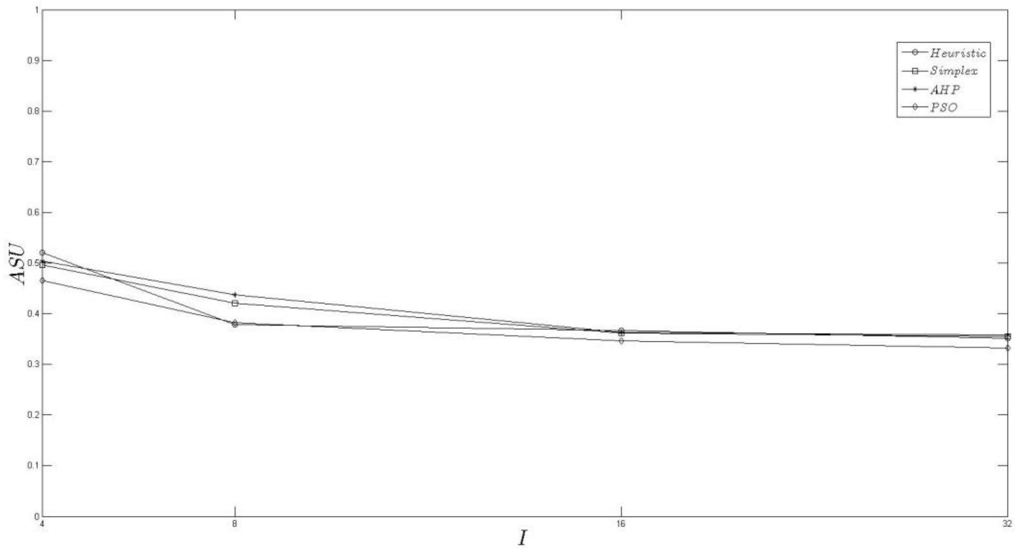


Figure 27: ASU in 2nd set of experiments

In Figure 28, for I = 4 and V = 100 the AR values of all methods are similar to the values in 1st scenario. While the values of I rise, we can observe that the AR values for the Heuristic, Simplex and AHP algorithms are not affected. Unlike the previous methods, PSO reduces the rounds needed to reach an agreement. This reduction is due to the fact that the particles have more space to move and to be deployed.

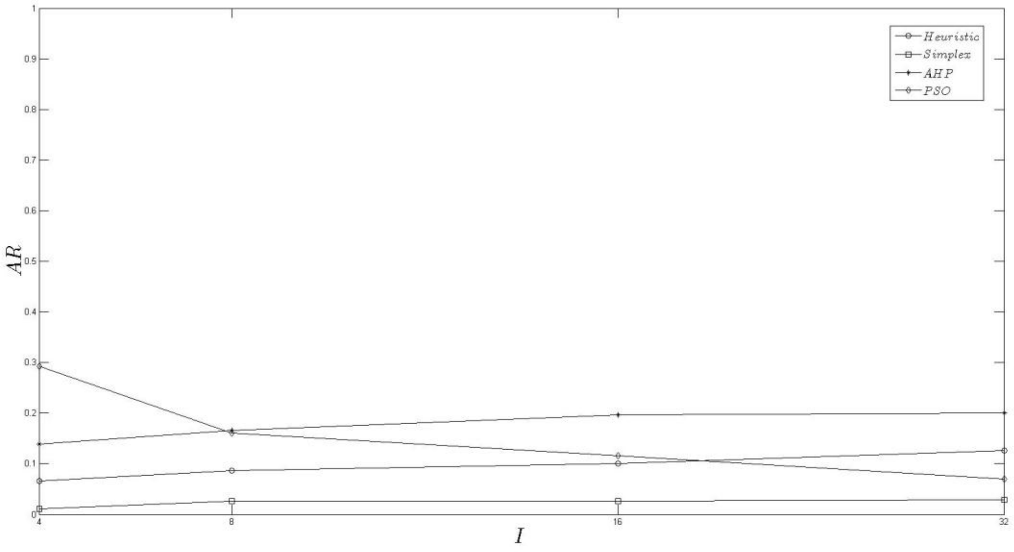


Figure 28: AR in 2nd set of experiments

In addition, another metric that is not affected by the increase in the number of issues (I) is $P_t$. We can conclude, from the Figure 29, that the threads managing to reach an agreement are not affected by the number of issues which are negotiated.

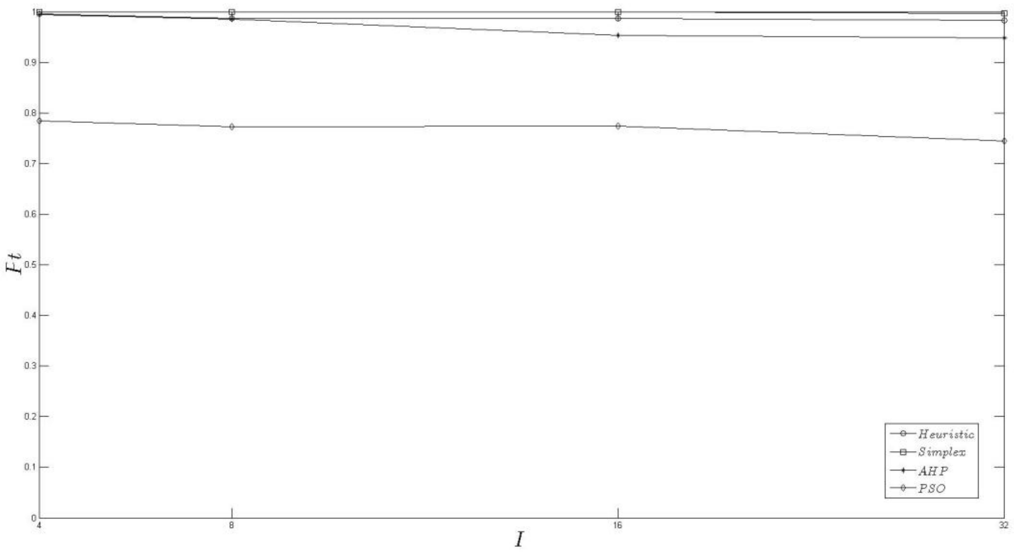


Figure 29: Pt in 2nd set of experiments

Metric F is the last considered in the 2nd scenario. In Figure 30 we can make two observations. The first is that the values of our methods over the threshold I = 8 is very close to 0.5. The values are considerably reduced in relation to the F values corresponding to the 1st scenario. The second observation is that for I ≥ 8 the values are not further improved. F is independent according to the number of issues when the I ≥ 8.

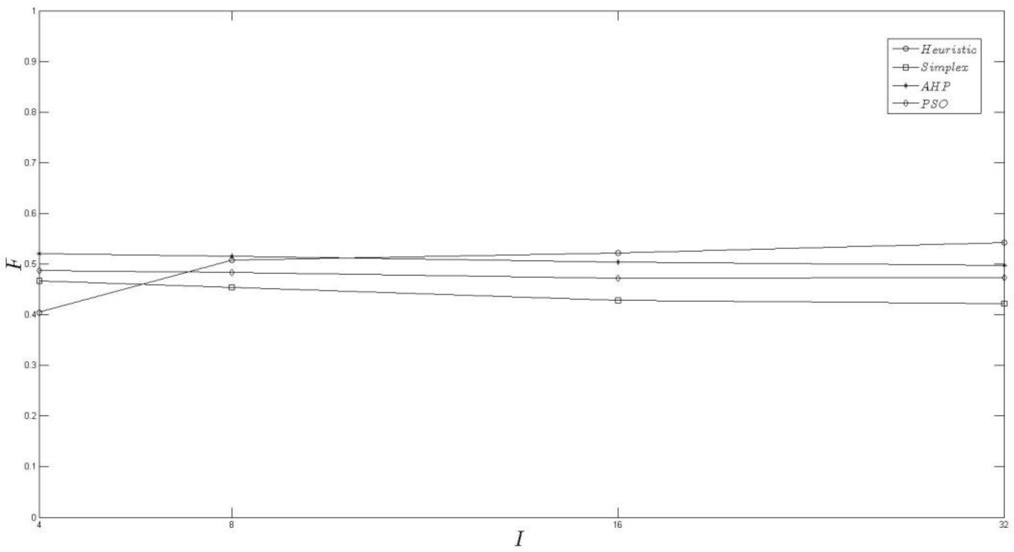


Figure 30: F in 2nd set of experiments

### 8.2.5 Description of 3rd Set of experiments

In the third set of experiments, we would like to study how the number of threads which represent the same user, affecting the performance of the discussed methodologies. In our 3rd scenario, we assume that the V = 100, I = 4 and $N_T$ increases from 5 to 50. For every value of $N_T$ in the interval [5, 50] we run 500 simulations. It is reasonable that the greater the number of threads is the higher the possibility to close an agreement and, thus, trigger the process of reconfiguration of weights.

The AG as metric is affected sufficiently by the number of the $N_T$. Although in the previous scenario, the average of AG values was close to 0.9, for $N_T$ = 5, AG values are close to 0.5. PSO method is the most influenced by the fluctuation of $N_T$, which has the minimum value of all methods close to 0.4. Similar to PSO, Heuristic and Simplex start from AG =0.5 and rise until the AG reaches 0.9. AHP is the only method that is independent from the changes in $N_T$.

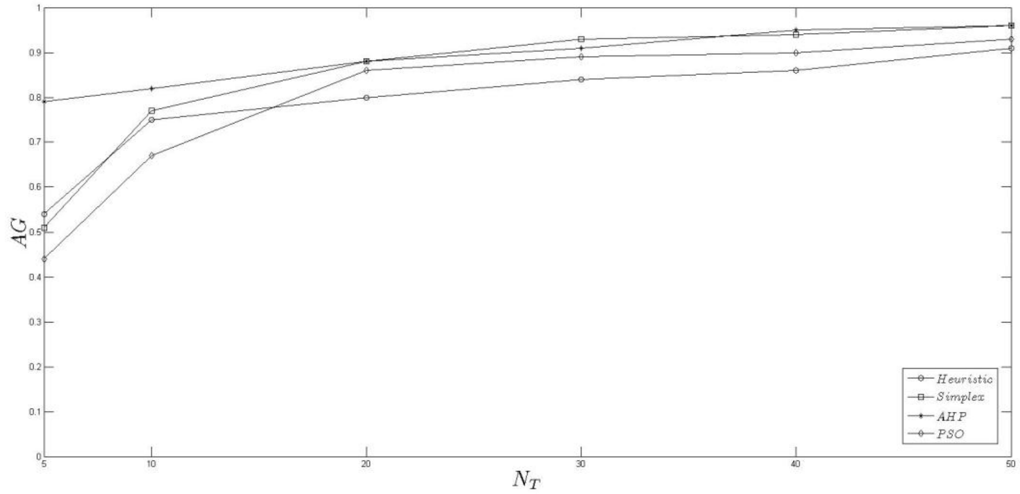


Figure 31: AG in 3rd set of experiments

Besides AHP, in all other methods an increase in $N_T$ is equivalent to increase in ABU as depicted by Figure 32. The ABU starts from a minimum and reaches a maximum for $N_T$ = 50. This increase is logical because, if more threads are involved in a negotiation, more agreements are made triggering the algorithms. With more threads, the possibility is higher for a $thread_i$ to close an agreement and to trigger the reconfiguration function for the rest of the threads. Similarly, the performance of PSO depends on the number of particles that take place in negotiations justifying the minimum ABU = 0.25 for $N_T$ = 5.

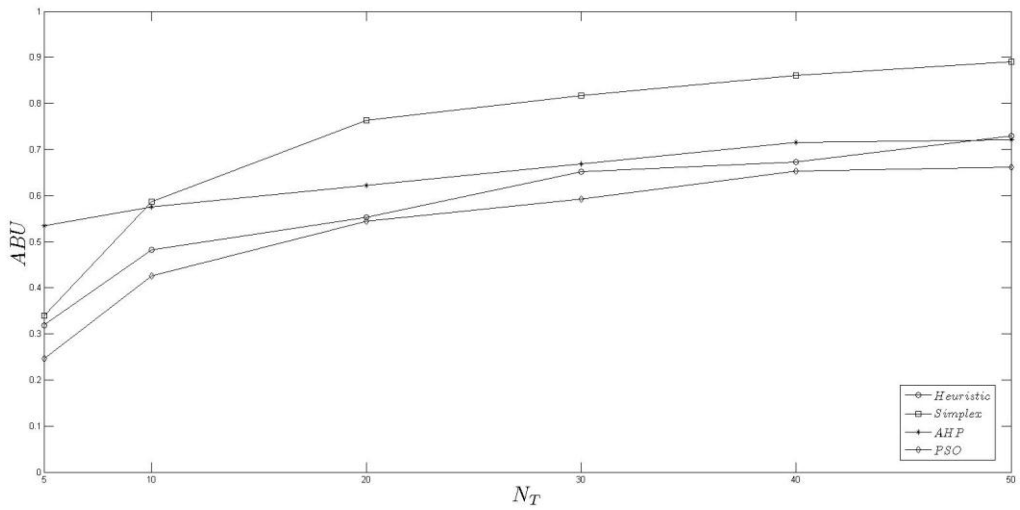


Figure 32: ABU in 3rd set of experiments

The ASU is not affected by the number of threads, because there is not defined in seller a function for weights reconfiguration. A smooth slope in Figure 33 is justified by the definition of the ASU itself. The final selected agreement is based on the buyer's utility, so the buyer's strategies affect slightly the utility of the user.

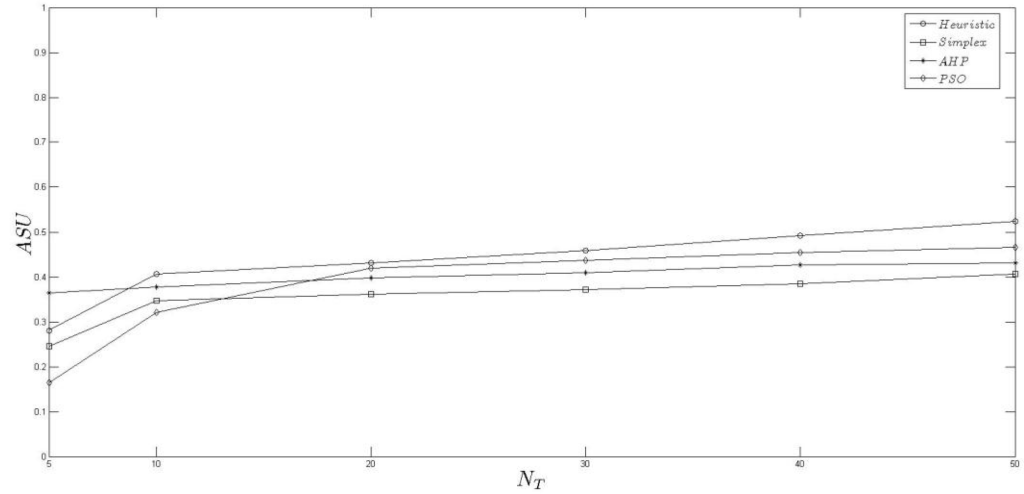


Figure 33: ASU in 3rd set of experiments

In Figure 35, AR for Heuristic, Simplex and AHP is independent by the change of $N_T$. The only method that is slightly affected is PSO because fewer particles need more time to converge in space.

Regarding $P_t$, Figure 35 shows that in all methods this metric is independent by the number of $N_T$. PSO has the lowest average values of $P_t$ compared to the other three methods. This happens because the particles try to converge to the optimal agreement consuming the negotiation time or rejecting the counter offers of the relevant sellers.

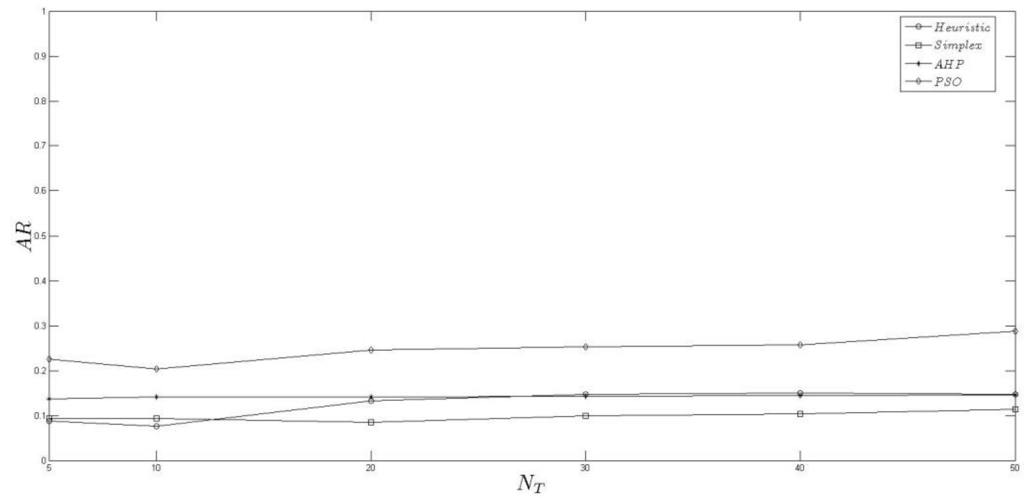


Figure 34: AR in 3rd set of experiments

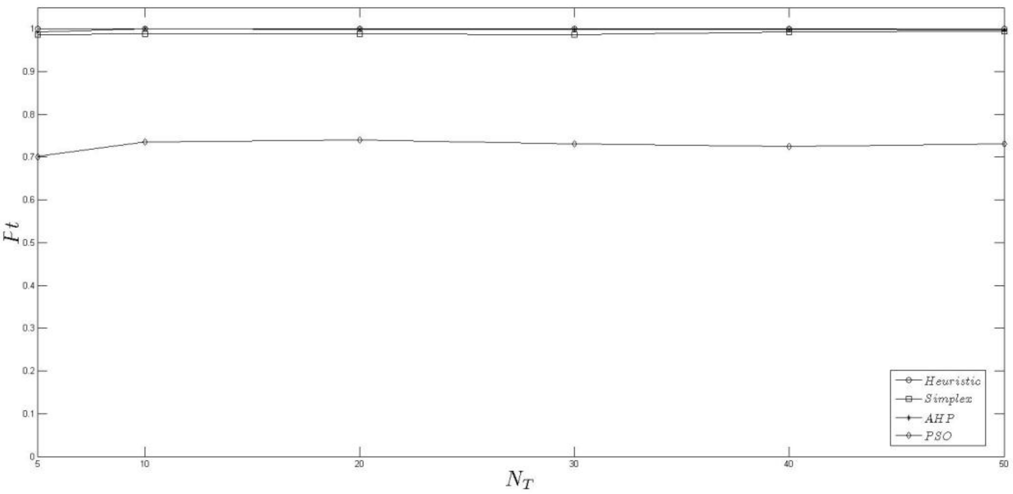


Figure 35: $P_t$ in 3rd set of experiments

Finally, in Figure 36, we can observe that the fairest method, e.g. with the average lowest value of F, is the AHP. In the rest methods, as we explained above with more threads participating in negotiations, the greater is the rate to trigger the weights configuration function. Hence, when the redefinition function starts, the seller is pressed to offer a better agreement and the negotiation tilts in the part of buyer resulting to F ≥ 0.5.

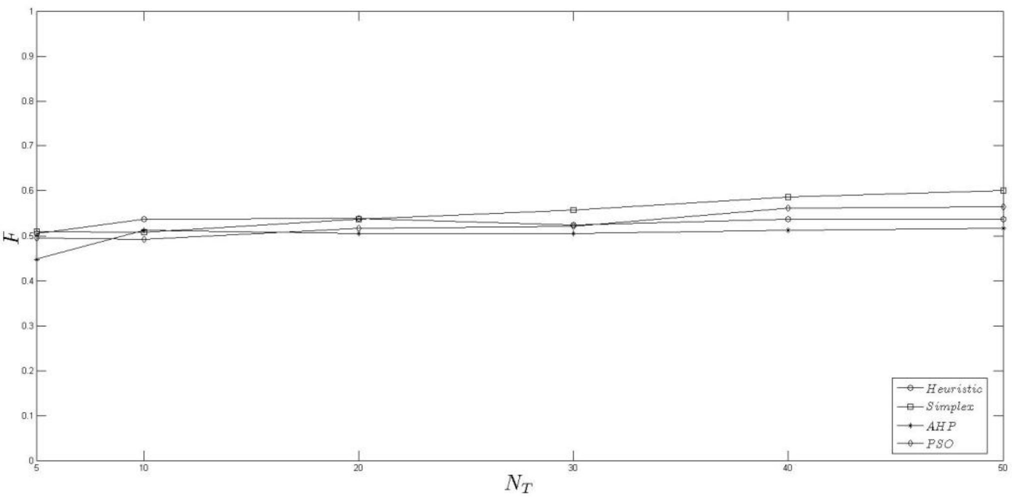


Figure 36: F in 3rd set of experiments

We would like to conclude that the performance of our methods depends on the initial values of our system. For a large number of threads and a small number of issues the Simplex method reaches the best agreement. A large number of issues combined with a large number of threads can be handled as input by the PSO algorithm. AHP can be characterized as a stable method but with good (not excellent performance). AHP can be used for a system which is robust under all circumstances.

# 9. OPEN ISSUES AND FUTURE WORK

## 9.1 Conclusions

In this thesis, we focus on the decision making algorithms that can be implemented in real life negotiations. The basic idea is to design an algorithm which can deal with one-to-many, concurrent, dynamic and with limited knowledge negotiations. The proposed solution is an automated process of dynamically and independently change of the adopted strategy of the IAs involved. Even if the current research efforts mainly deal with a subset of the discussed problem, we have designed four algorithms that can dynamically change buyer's strategy during realistic negotiations. The first three algorithms, e.g. Heuristic, Simplex and AHP methods, redefine the weights of product's issues resulting to a change in the calculation of the buyer's reward (utility function). Also, we approach our problem through moving IAs in the N-dimensional space applying the Particle Swarm Optimization algorithm (PSO). The offer made by each IA is at every round a new position in space, which depends on local and global best of rest IAs as well as the velocity of the same. Theoretical analysis and experimental results show that the average utility gained by the buyer in all methods is above 50%. Furthermore, all methods can handle multi issue negotiations and specifically PSO algorithm can handle excellent a large number of issues combined with a large number of IAs.

## 9.2 Future Work and Open Issues

We present methods for optimizing multi-issue negotiations focusing on the buyer's side. The function of weights' configuration has been added in the functionalities of the buyer. The next steps for our work is to define relevant function for dynamically change of weights for the seller's part. Also, PSO algorithm requires from particles to move in N-dimensional space. The following step for PSO algorithm is to study whether the behavior of particles will change, if the weights of issues can be dynamically defined again during the negotiations.

In addition, we have developed artificial scenarios to evaluate the proposed methods. We have tried to evaluate the behavior of the algorithms and their performance, while we have been changing our metrics. The comparison of our results with real data or with Pareto's Optimality would give us more realistic perspective between the developed methods. For example in a future realistic system, N users can define their preferences for the negotiations. The threads will return the agreements closed with the sellers and users will select one of them as the most preferable. Our proposed methods, also, will choose the optimal agreement for the user. The choices of the designed methods and the choice of the user can be compared providing us with the “closest-to-human-behavior” methodology.

# ABBREVIATIONS – ACRONYMS

| | |
|---|---|
| AHP | Analytic hierarchy process |
| AI | Artificial Intelligence |
| API | Application Program Interface |
| B2B | Business-to-Business |
| B2C | Business-to-Consumer |
| C2B | Consumer-to-Business |
| C2C | Consumer-to-Consumer |
| EANS | Electronic Automated Negotiation Systems |
| E-Commerce | Electronic Commerce |
| E-Marketplace | Electronic Marketplace |
| E-negotiations | Electronic Negotiations |
| E-Payment | Electronic Payments |
| GUI | Graphical User Interfaces |
| IA | Intelligent Software Agent |
| IM | Information Marketplace |
| InA | Interface Agent |
| ITA | Intelligent Trading Agency |
| NSS | Negotiation Support Systems |
| PSO | Particle Swarm Optimization |
| QoS | Quality of Service |
| T@T | Tete-a-Tete |
| VFA | Virtual Force Algorithm |

# BIBLIOGRAPHY


[1] Fonseca, S., M. Griss, et al. (2001). An Agent Mediated e-Commerce Environment for the Mobile Shopper, Hewlett Packard.

[2] Kraus, S., Negotiation and cooperation in multi-agent environments, Special issue on economic principles of multi-agent systems in Artificial Intelligence, Vol (94):1-2, pgs. 79-97, 1997

[3] Nash, J.F., The Bargaining Problem, Econometrica 18 (1950), pgs. 155-162 von Neumann, J. and Morgenstern, O., The Theory of Games and Economic Behavior, Princeton University Press, 1994

[4] Tsvetovatyy, M., Gini, M.: Toward a Virtual Marketplace: Architectures and Strategies. Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM'96, The Practical Application

[5] Company Ltd, Blackpool (1998) 597-613

[6] Krulwich, B.: Bargain finder agent prototype. Technical report, Anderson Consulting,1999

[7] Doorenbos, R., Etzioni, O., Weld, D.: A Scalable Comparison-Shopping Agent for the World Wide Web. Proceedings of the First International Conference on Autonomous Agents (Agents's 97). Marina Del Rey, CA (1997)

[8] Mougayar, W.: Opening Digital Markets. 2nd edn. McGraw-Hill (1998)

[9] Bailey, J. P.: Intermediation and Electronic Markets: Aggregation and Pricing in Internet Commerce. Doctor's Thesis, Computer Science Department, Uppsala University, Sweden (1998)

[10] Brazier, F., Keplicz, B. D., Jennings, N. R., Treur, J.: Formal Specification of Multi-Agent Systems: a Real-World Case. In V. Lesser (ed.), Proceedings of the First International Conference on Multi-Agent Systems, ICMAS'95, MIT Press, Cambridge, MA (1995) 2532. Extended version In International Journal of Cooperative Information Systems, M. Huhns, M. Singh, (eds.), special issue on Formal Methods in Cooperative Information Systems: Multi-Agent Systems, vol. 6 (1997) 67-94

[11] Alfred Marshall. 1920. Principles of Economics. An introductory Volume. 8th edition. London: Macmillan.

[12] Gerding, E. H., D. J. A. Somefun, et al. (2004). Multi-Attribute Bilateral Bargaining in One-toMany Setting. Amsterdam, Centre for Mathematics and Computer Science (CWI).

[13] Luo, X., J. H. Lee, et al. (2003). "Prioritised Fuzzy Constraint Satisfaction Problems: Axioms, Instantiation and Validation." Fuzzy Sets and Systems 136(2): 155-188.

[14] Kowalczyk, R. and V. Bui (2003). On Constraint based Reasoning in eNegotiation Agents. Agent Mediated Electronic Commerce III. F. Dignum and U. Cortes, Springer Verlag 31-46.

[15] Faratin, P., C. Sierra, et al. (1998). "Negotiation Decision Functions for Autonomous Agents." Robotics and Autonomous Systems 24(3-4): 159-182.

[16] Reid G. Smith "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", Readings in DAI, (ed Bond and Gasser),1985

[17] Almeida, V., Meira, W., Ribeiro, V., Ziviani, N.: A Quantitative Analysis of the Behaviour of a Large Non-Englisg E-Broker. In Proc. of the International Workshop on Advanced Issues of E-Commerce and Web-based Information Systems (WECWIS'99), Santa Clara, CA (1999)

[18] Erikson, J., Finne, N., Janson, S.: Market Space: an open agent-based market infrastructure. Master's Thesis. Computer Science Department, Uppsala University, Sweden (1997)

[19] Li, C., Giampapa, J. A., & Sycara, K. (2003). A review of research literature on bilateral negotiations. Technical Report CMU-RI-TR-03-41. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University

[20] Schleiffer, R. (2005). An intelligentagent model. European Journal of Operational Research, 166(3), 666–693.

[21] Jennings, N. and M. Wooldridge (1998). Applications of Intelligent Agents. Agent Technology: Foundations, Applications and Markets. N. Jennings and M. Wooldridge, Springer-Verlag.

[22] Bibliografy Gilbert, Aparicio, et al. The Role of Intelligent Agents in the Information Infrastructure. IBM, United States, 1995.

[23] Caglayan, A. and C. Harrison (1997). Agent Sourcebook. New York, John Wiley & Sons, Inc

[24] Gopal Sakarkar & Sachin Upadhye (2010). A Survey of Software Agent and Ontology. 2010 International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 7

[25] "E-negotiation systems: Interaction of people and technologies to resolve conflicts", Gregory E. Kersten, UNESCAP Third Annual Forum on Online Dispute Resolution, Melbourne, Australia, 5-6 July 2004

[26] Parkes, D. C., Ungar, L. H. and Foster, D. P. 1999. Accounting for Cognitive Costs in On-line Auction Design. In Noriega, P. and Sierra, C. eds., Agent mediated electronic commerce (LNAI 1571), 25--40. Springer-Verlag.

[27] Sandholm, T. 1996. Negotiation among Self-Interested Computationally Limited Agents. Ph.D. Dissertation. University of Massachusetts at Amherst, Department of Computer Science.Chapter 5.

[28] S. McConnell, M. Mez, L. Maesano and M. Witthaut. An open architecture for electronic commerce. Technical Report, Object Management Group, Cambridge, MA, 1997

[29] J.M. Tennebaum, T.S. Chowdhry and K.Hughes. eCo System: CommerceNet's architectural framework for internet commerce. Technical report, Object Management Group,Cambridge, 1997

[30] Iyad Rahwan, Ryszard Kowalczyk, Ha Hai Pham. Intelligent Agents for Automated One-to-Many e-Commerce Negotiation. Twenty-Fifth Australian Computer Science Conference,2002

[31] Paolo Torroni, Francesca Toni. Extending a Logic Based One-to-One Negotiation Framework to One-to-Many Negotiation. ESAW '01 Proceedings of the Second International Workshop on Engineering Societies in the Agents World II. Pages 105-118. Springer-Verlag London, UK. 2001

[32] Hillier F.S. - Lieberman G.J., (1984), '*Εισαγωγή στην Επιχειρησιακή Έρευνα*', Τόμος Α, Αθήνα: Εκδόσεις Παπαζήση.

[33] Φακίνος Δ. - Οικονόμου Α., (2003), *'Εισαγωγή στην Επιχειρησιακή Έρευνα- θεωρία και Ασκήσεις'*, Αθήνα: Εκδόσεις Συμμετρία.

[34] Σίσκος Γ., (1998), *'Γραμμικός Προγραμματισμός'*, Αθήνα: Εκδόσεις Νέων Τεχνολογιών.

[35] Feng Kong and Hongyan Liu, "Applying fuzzy analytic hierarchy process to evaluate success factors of e-commerce", International journal of information and systems sciences, Institute for Scientific Computing and Information,2005

[36] Saaty TL, "Axiomatic Foundations of the Analytic Hierarchy Process", Management Science, vol. 32, 7, July 1986, 841-855

[37] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. IEEE InternationaConference on Neural Networks, pp. 1942–1948 (1995)

[38] V. Chvatal, *Linear Programming*, W. H. Freeman and Company, 1983

[39] Saaty, T.L. 1990: How to make a decision: The Analytic Hierarchy Process. European Journal of Operational,Research 48: 9−26

[40] Intelligent Agents for Automated One-to-Many e-Commerce Negotiation Abstract Iyad Rahwan Department of Information Systems , University of Melbourne Ryszard Kowalczyk and Ha Hai Pham CSIRO Mathematical and Information Sciences 713 Swanston Street, Carlton, Vic 3053, Australia

[41] Fatima, S. S., Wooldridge, M. and Jennings, N. R.. “Optimal Agendas for Multi-Issue Negotiation”. Second International Conference on Autonomous Agents and Multiagent Systems (AAMAS-03), Melbourne, July 2003, pp. 129-136.

[42] Fatima, S., Wooldridge, M. and Jennings, N. R.. “Optimal Negotiation Strategies for Agents with Incomplete Information” . In Intelligent Agents VIII, Springer-Verlag LNAI, vol. 2333, pp 377-392, March 2002

[43] Fatima, S., Wooldridge, M. and Jennings, N. R.. “Optimal Negotiation Strategies for Agents with Incomplete Information” . In Intelligent Agents VIII, Springer-Verlag LNAI, vol. 2333, pp 377-392, March 2002

[44] Demonstration of a Software System for Automated Multi-Attribute Negotiation (2004), Catholijn Jonker , Lourens van der Meij , Valentin Robu , Jan Treur, the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMA) 2004

[45] Rahwan, I.; Kowalczyk, R.; and Pham, H. H. 2002. Intelligent agents for automated one-to-many e-commerce negotiation. In CRPITS ’02: Proceedings of the twenty-fifth Australasian conference on Computer science, 197–204.Australian Computer Society, Inc.

[46] Nguyen, T. D., and Jennings, N. 2004a. Coordinating multiple concurrent negotiations. In Proc. Third International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS’04), 1064–1071. New York,USA: ACM.

[47] Moss, J. E. 1985. Nested transactions: an approach to reliable distributed computing. Massachusetts Institute of Technology.

[48] Sandholm, T., and Lesser, V. 1995. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In Lesser, V., ed., Proceedings of the First International Conference on Multi-Agent Systems (ICMAS’95), 328–335. San Francisco, CA, USA: The MIT Press: Cambridge, MA, USA.

[49] Zhang, X.; Lesser, V.; and Abdallah, S. 2004. Efficient Management of Multi-Linked Negotiation Based on a Formalized Model. Autonomous Agents and Multi-Agent Systems.

[50] Anthony Chavez , Pattie Maes; Kasbah: An Agent Marketplace for Buying and Selling Goods,1996, Conference on the practical application of intelligent agents and multi-agent technology

[51] M. Locateli and U. Raber, “Packing equal circles in a square: a deterministic global optimization approach”, Discrete Applied Mathematics, vol. 122, pp. 139-166, October 2002.

[52] Howard, M. J. Matari´c and G. S. Sukhatme, “Mobile Sensor Network Deployment Using Potential Field: a distributed scalable solution to the area coverage problem”, to appear in Proc. International Conference on Distributed Autonomous Robotic Systems”, June 2002

[53] Lecky-Thompson G.W, “Corporate Software Project Management”,2005

[54] Fatima, S. S., Wooldridge, M., and Jennings, N., 'Multi Issue Negotiation Under Time Constraints', In Proceedins of the AAMAS 2002, Bologna, Italy, 2002.

[55] Oprea, M., 'An Adaptive Negotiation Model for Agent Based Electronic Commerce', Stuies in Informatics and Control, vol. 11(3), 2002, pp. 271-279.

[56] Zeng, D., and Sycara, K., 'Bayesian Learning in Negotiation', International Journal of Human Computer Studies, vol. 48 (1), 1998, pp. 125-141.

[57] Kolomvatsos, K., and Hadjiefthymiades, S., 'Buyer Behavior Adaptation Based on a Fuzzy Logic Controller and Prediction Techniques', Elsevier Fuzzy Sets and Systems, vol. 189(1), 2012, pp. 30-52.