Utility Hyper-graphs for Complex Decision Making

by

Rafik HADFI

THESIS

Presented to the

Department of Computer Science and Engineering,

in Partial Fulfillment of the Requirements for the Doctoral Degree

under the Guidance of Professor Takayuki ITO

Nagoya Institute of Technology

March 2015

Abstract

Preferences are fundamental for the analysis of human choices, as well as for the construction of artificial agents that can make decisions in a rational and autonomous manner. Consequently, utility models are becoming of increasing importance in many areas such as Multi-agent Systems, Decision Making, Social Choice, Constraint Satisfaction, Game Theory, and so forth.

Realistic decision making problems are characterized by high-cognitive load, which has a huge impact on the utility spaces of the agents. Such utility spaces are known to be complex, nonlinear, and involve a large number of attributes, or issues. These problems are challenging from two standpoints: the standpoint of the individual agent, whenever she is exploring her own preferences; and from the standpoint of a group of agents trying to build a consensus. Reaching an agreement under such constraints becomes more difficult as the search space and the complexity of the problem grow.

We propose a utility model that is suitable for complex decision making problems. The starting point is adopting a hyper-graphical representation for the utility model of an agent. This allows a modular decomposition of the issues and the underlying constraints by mapping the utility space into a Utility Hyper-graph (UH). Exploring the utility space reduces then to a message passing mechanism along the hyper-edges by means of utility propagation. Adopting such representation paradigm allows us to rigorously show how complexity arises in complex domains. To this end, we assess the complexity arising in cognitive graphical models using the concept of entropy. Being able to assess complexity allows us to improve the message passing algorithm by adopting a low-complexity propagation scheme. The model is evaluated using a parametric form of hyper-graphs, or Random Utility Hyper-graphs (RUH). Hence, we show how the new propagation scheme can optimally handle complex utility spaces while outperforming previous sampling approaches. Similarly, our proposed approach to quantify complexity allows us to establish the interplay between complexity, entropy and uncertainty in complex domains.

Acknowledgements

Writing this thesis would not have been possible without the support of the wonderful people around me.

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Takayuki Ito, for his guidance, patience, and constant support throughout the past five years; and for accepting me at his laboratory and giving me the opportunity to start a new challenging life in such an excellent environment. I could not wish for a better or friendlier advisor!

I also would like to express my gratitude to the Ministry of Education, Culture, Sports, Science, and Technology (MEXT) for awarding me the Japanese Government Monbukagakusho Scholarship. My heartfelt gratitude goes to Junko Sugiyama, Shun Okuhara, Takanobu Otsuka, Akihiko Nagai and Tomomichi Hayakawa for their kindness and unconditional help. I thank my fellow labmates Shunsuke Tanaka, Tsuguya Ueta, Yutaro Fujii, Mikoto Okumura, Deyue Deng, Raiye Hailu, Tatsunosuke Tsuboi and many others for their support and the wonderful time spent together. I would like to thank Associate Professors Naoki Fukuta, Ryo Kanamori and Katsuhide Fujita for their kindness and presence. I would like to express my warm thanks to Shantanu Chakraborty for his friendship and the wonderful time we spent together. My sincere thanks go to Tim Baarslag for his insights and discussions. I am particularly grateful to Professor Mark Klein for the insightful discussions (even if it was for a short period of time). Parts of this thesis would have not been written without him introducing me to his work. I also thank my friends from Nagoya University: Christian, Peter and Paul for their friendship and the fun times we have spent together. My deepest thanks to my dearest friends Ghassen and Wijaya for their presence.

Finally, I want to express my deepest gratitude towards my parents and my brothers for their unfaltering love and support. They have always been a tremendous source of strength and inspiration for me.

Last but not least, I would like to express my heart-felt thanks to my lovely Joanna. She has always been by my side, encouraging, helping and caring, through the good and bad times.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement and Goals

The assessment of preferences is a fundamental task for the analysis of human choice behavior and the design of rational and autonomous agents. This problem is found in many areas of Artificial Intelligence such as Multi-agent Systems [5], Game Theory [98], Decision Making [61], Social Choice [56, 7], Constraint Satisfaction [27], and so forth. As the problem domain grows, this assessment task becomes harder in the sense that even the representation of the available choices becomes intractable. This is typically found in cognitive tasks with high-information load [16], where a decision maker builds a preference model that involves a large number of attributes and relationships. It is therefore important to find the right representation as to tackle the complexity problem found in real-world decision problems.

This complexity is also found whenever we have a group of agents interacting in a strategic manner, for instance, in a bilateral or multilateral negotiation. In such situations, agents usually do not share their preferences as to avoid exploitation. It is therefore common that an agent tries to model the opponent's behavior in order to predict her behavior. This could allow both agents to find a mutually satisfactory outcome, measured in terms of social welfare. In this purely strategic context, it would be interesting to model the opponents' behavior using a particular mathematical model that reflects her economic type and her moves.

It is within this perspective that we propose to answer the following questions:

1. How to define complexity as it arises in decision making, as opposed to computational complexity? We seek a definition that is inherent to the description of the problem itself, *i.e.*, how the agent builds her choice model and uses it to make decisions.

2. Is there an interplay between complexity and uncertainty? In other words, how does the complexity of a given problem domain affect the agent's uncertainty whenever she is reasoning about the best outcomes?

3. How to find one representation of complex utility spaces that takes into consideration all the existing utility functions' shapes? And for several economic types, is there a way to represent them using one unique form?

4. How can we assess complexity in a way that correlates with the notion of utilitarian optimality? In other words, how can complexity be tackled in the way that maximizes the utility of the agent?

5. How to perform efficient search and optimization in complex utility spaces? By "efficient", we mean more efficient than the existing sampling-based approaches.

6. How to decompose a complex utility space and exploit its structure, especially when the problem is hard and involves a strategic encounter? Exploiting the problem structure could yield better strategies whenever the agents share their similarities during an encounter. This similarity can for example manifest itself in the constraints.

To address these problems, we set a number of goals to achieve all along this thesis. The goals are described as following:

1. Analyze and quantify complexity as it arises in challenging decision making problems. That is, problems that are subject to:

    (a) High-information Load, due to the large knowledge domains and the large number of attributes (issues) to be dealt with.

    (b) High uncertainty, as a result of (a). This is due to the increasing number of alternatives that the agent is facing, which yields more uncertainty about her own choices and the moves of the opponent.

The targeted decision making problems are mainly those specific to preferences elicitation. However, we take the case of complex automated negotiation as an instance of complex decision making problems that we are interested in investigating using our model. The quantification of uncertainty will be established using the information theoretic notion of entropy, for it is the adequate tool to reason about probabilistic settings and uncertainty.

2. Propose the adequate model for complex utility spaces modeling. Particularly, we are interested in a representation that could shed some light on how issue-interdependence affects the complexity of a given utility space.

3. Lowering the complexity of a preferences elicitation process using low-cost search in a complex utility space. This is done by using the recent findings in cognitive sciences and psychology as to understand "Fast and Frugal Heuristics" and their functioning in decision making under bounded rationality [45, 44, 43, 16].

4. Proposing a unified form for constraint-based utility functions. Being able to unify all the well known utility forms with their intrinsic economic characteristics (risk aversion, indifference, discounting, etc.) could help in the study of random utility hyper-graphs. Similarly, it allows a more principled opponent modeling approach that is mainly inspired from Mixture Models in Machine Learning. Additionally, it could help a bounded rational agent in learning her own preferences whenever she is dealing with large domains.

To reach these goals, we make usage of several analytic tools from multiple domains, summarized in Figure 1.1.

1.2 Thesis Structure

The abovementioned goals will be investigated all along the thesis according to the general structure shown in Figure 1.2. First, we propose an overview of the related works, within the general scope of decision making and preferences elicitation. Additionally, we survey several

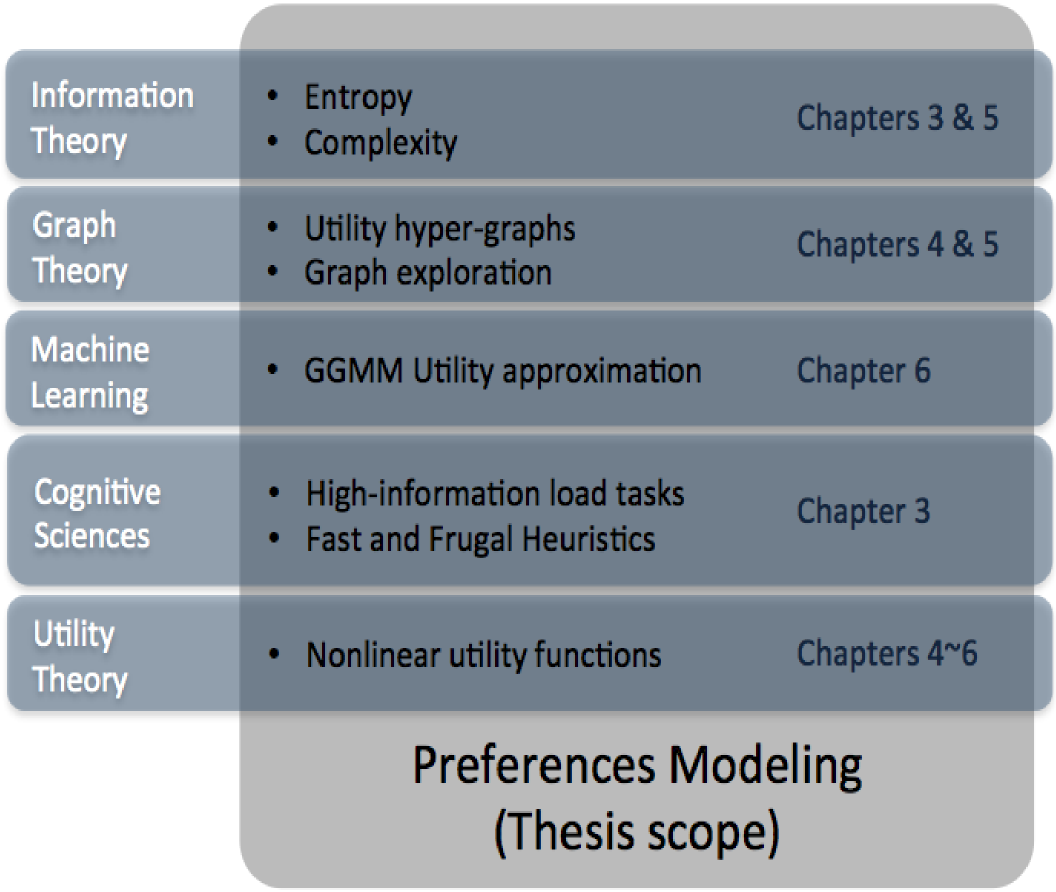


Figure 1.1: Thesis scope and domains

utility representations as well as their usages, and show the differences with our work (Chapter 2). Second, we provide a study of complexity and show how it arises in cognitive tasks under high-information load (Chapter 3). Third, we present the main component of the thesis, that is, a preference representation based on utility hyper-graphs (Chapter 4). Fourth, we use our complexity study to elaborate a low-complexity search mechanism that can be used in utility hyper-graphs' exploration (Chapter 5). Fifth, we provide an application of the utility hyper-graph model in complex automated negotiation (consensus building) as well as for the approximation of constraint-based utility functions (Chapter 6). Sixth, we propose an application of the complexity study to preferences elicitation and collaborative design (Chapter 7). Finally, we draw the conclusions of this thesis and highlight the future works (Chapter 8).

1.3 Contributions

The main contributions of this thesis are summarized as following:

1. Quantifying cognitive complexity using information entropy (Chapter 3)

   We bridge the gap between complexity as perceived in cognitive sciences, *i.e.*, as a cognitive load in cortical areas, occurring during high-information load tasks; and complexity as it is perceived in decision making, occurring under endogenous bounded rationality. We are particularly interested in measuring complexity whenever the decision maker is adopting a graphical representation as a choice model

2. Complexity, entropy and uncertainty in high-information load tasks (Chapter 3)

   We study the evolution of cognitive processes with high-information load. This is done by drawing the interplay between the structural complexity of the cognitive graph of the decision problem, its entropy, and how the uncertainty of the whole process evolves.

3. Utility hyper-graphs for complex preferences representation (Chapter 4)

   We propose a novel representation for nonlinear utility spaces as to tackle the complexity problem. It has the merit of being modular and parametric, which allows search strategies evaluation as well as any graph-theoretic analysis. We also propose efficient heuristics for optimal contracts search based on message passing in the utility hyper-graph.

4. Quantifying the complexity of nonlinear utility spaces. (Chapter 5)

   We propose an efficient method to assess the complexity of a nonlinear utility space using its induced utility hyper-graph. We use the information theoretic concept of entropy applied to the constraints' degree distribution.

5. Low-complexity search in utility hyper-graphs. (Chapter 5)

   Based on the complexity measure, we provide several search strategies and identify the optimal strategy that minimizes the search cost. The optimal search strategy allows a low-complexity traversal of the utility hyper-graph while preserving the contracts optimality.

6. Using utility hyper-graphs in multilateral negotiation. (Chapter 6)

   We provide an evaluation of the hyper-graphic representation in a multilateral mediated negotiation setting. We show that under high complexity, the collective social welfare could be greater than the sum of the individual expected best utilities.

7. Unifying constraint-based utility functions. (Chapter 6)

   We propose a formula that unifies all known constraint-based (Cubic, Bell, Conic, etc.) utility functions. The new representation leads us to a parametric model that could be used for opponent modeling in complex nonlinear negotiations.

8. Asymptotic maximum entropy utility principle. (Chapter 7)

   We propose a Maximum Entropy principle for preferences elicitation for utility functions with a countably infinite number of outcomes. Herein, we study the evolution of preferences (defined as a merging process) defined over an infinite number of outcomes, perceived as a way to describe the high predictive uncertainty that an agent is facing in complex problems.

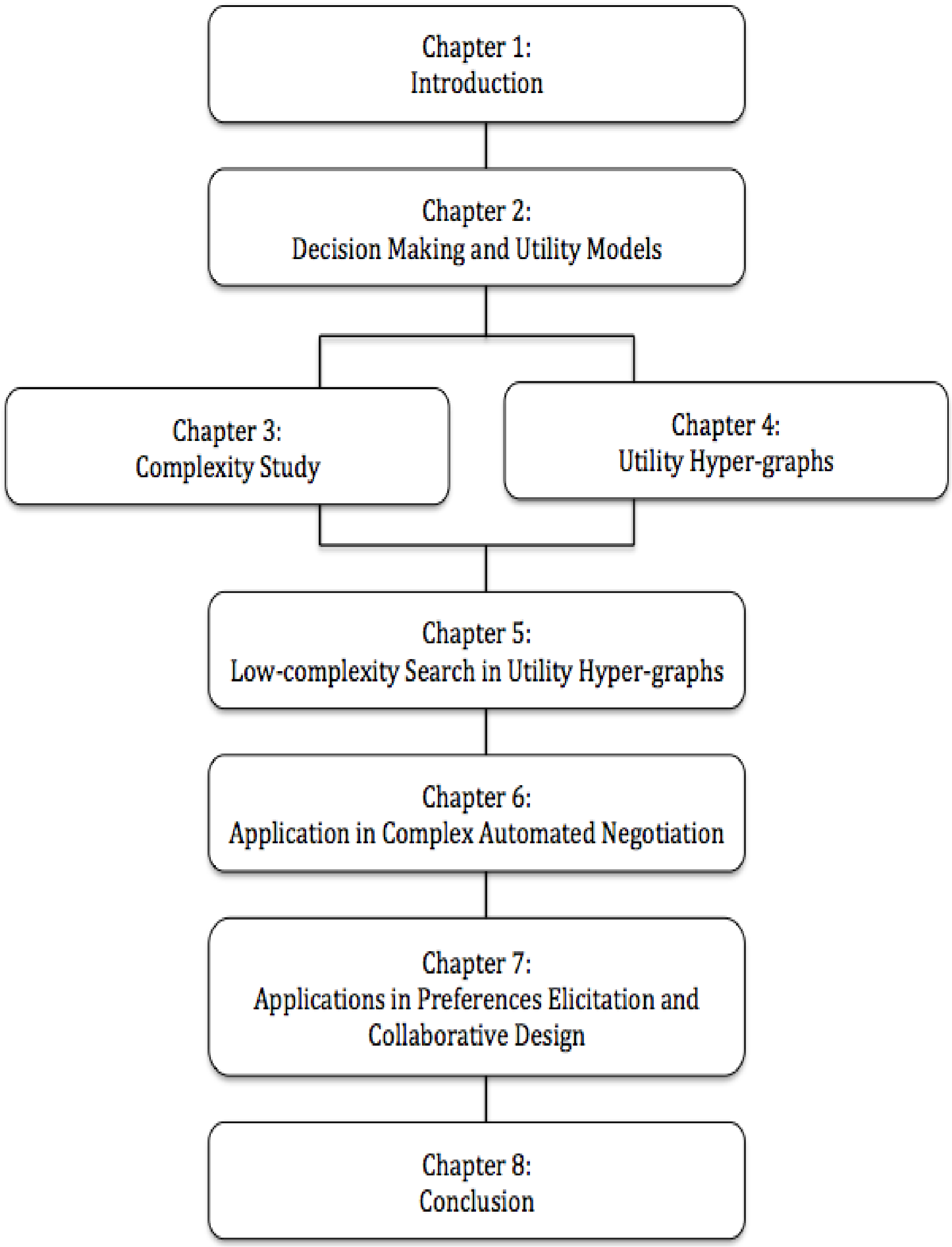


Figure 1.2: Thesis Structure

Chapter 2

Decision Making and Utility Models

2.1 Introduction

One of the main problems in decision making is the process of utility elicitation or estimation. That is, how the decision maker can define a utility function that involves his goals and is consistent with the available information. Several techniques of utility representation and assessment have been proposed. However, depending on the problem's domain size, some of these techniques are less efficient than others. Our motivation is to propose the type of analytical framework that is suitable for domains that are characterized by high-information load.

2.2 Decision Making

2.2.1 Decision with Multiple Attributes and Objectives

Decision making is mainly the cognitive process of assessing a situation, by selecting a choice among several alternative choices. An important issue in decision making is the availability of knowledge. In fact, although realistic situations involve incomplete information, decisions must be made. The most common way to model these decision problems is based on the premise of multiple attributes (or criteria, goals) and alternatives. Thus, the branch of Decision Making called Multi-Attribute Decision Making [62], which basically deals with decision problems under the presence of a number of decision criteria. Given the fact that decisions must often, perhaps usually, be made in the face of lacking or imperfect information, one solution is to try to reconcile these uncertainties with the available data. Expected Utility Theory and Risk Analysis could provide some guidelines to tackle this problem and attempt to make decisions.

2.2.2 Decision Making under Uncertainty

In most decision problems, a decision maker does not know the consequences of his choices nor the probability of each consequence under each choice. For instance, in a given game, a player will have to reason about his own moves, as well as the other player's moves. Thus, the strategic profiles induced from the consequences. In such situation, the decision maker holds subjective beliefs about the unknown aspects of the problem and attempts to use these beliefs. Such subjective beliefs could be represented by a probability distribution that assigns a likelihood to each event. Under such assumption, the decision making problem is said to be under uncertainty.

2.2.3 Logical Deduction and Decision

The act of making decisions involves the task of "analysis", in the sense that the whole problem is divided into smaller components [3]. This analytical thinking is in fact based on a logical step-by-step reasoning characterized by two aspects: *deduction*, and *induction*.

On the one hand, deductive reasoning is the process of deducing conclusions based on previously known or assumed facts, and tends to infer from the general to the particular; on the other hand, inductive reasoning arrives at a conclusion based on a set of observations, which makes it an invalid method of proof. Consequently, the task that the decision maker is facing will be more similar to a deduction problem, although optimality is the ultimate goal, rather than provability.

The top-down logic of the deductive reasoning becomes computationally intractable when applied to complex real-world problems, particularly when uncertainty is involved [91]. Therefore, we cannot use explicit deductive reasoning based on either inference rules or mental models in the face of complex problems. These complex real-world problems are usually knowledge-based and are thus dealt with by the induction of schemas or by connectionist mechanisms which are known to be very efficient in the human brain [26]. Hence, most of real-world reasoning will be essentially inductive and implicit, rather than deductive and explicit.

Herein, we adopt an induction-based mechanism by bridging the gap between modularity and connectionism in one single framework. Our induction-based mechanisms will operate on particular topologies and will make usage of heuristic-like principles [45, 44, 43] whenever we are searching for the optimal choices.

2.2.4 Bounded Rationality

One of the central problems in artificial intelligence is that agents cannot perform unbounded computations, especially when they are deployed in complex environments. This stems from the failure of pure rationality as a descriptive model of human behavior. In fact, the limits on rationality are due to the human cognitive architecture (internal constraints) as well as the external constraints defined by the environment. Bounded rationality comes in hand as to oppose the assumption of comprehensive rationality found in economic and decision theory models of choice [86, 25, 57, 89]. Despite the fact that it assumes that actors are goal-oriented, it tends to take into account the cognitive limitations of the agent in attempting to achieve those goals. Practically, rather than making assumptions about decision making and modeling the implications mathematically, it adopts an explicitly behavioral attitude. Herein, we focus on the type of mental representation that could possibly model the preferences of a bounded rational agent.

2.2.5 Cognition with High-information Load

Real-world problems are known to have high-information load in the sense that they are informationally demanding [94]. In fact, decision making depends on knowledge-intensive methods of extracting information from our world as to perform future actions.

Despite this complexity, humans are impeccable at handling these high-information load problems in a quick and frugal way [16]. The reason could in fact reside in the modular ways we tend to respond to these problems. Therefore, it is important to account for modularity whenever we want to reason about agents dealing with decision making problems with high-information load.

Generally, any cognitive system is either *modular* or *non-modular* [31]. *Modular* systems are perceived as "localized psychological faculties" [31] ensuring a domain-specific processing of *independent* modules. They are vertical, computationally autonomous and non-interactive. Most importantly, they are known for the encapsulation of information, *i.e.*, these processing systems are isolated, content-specific, with independent subsystems that do not make use of other aspects of cognition. On the other hand, *non-modular* systems are in principle connectionist, horizontal, characterizing cognitive processes that are not content specific but rather interactive, whereby interactions take place in neural networks. Specifically, connectionism refers to the nonlinear dynamic processing [9, 49] that is taking place. The idea behind connectionism is that some aspects are not the result of learning but are in fact genetically inherited. For instance, the architecture of a particular cognitive map could be genetically inherited along with the concepts and associations. Particularly, whenever it is instantiated as a neural network, the connections' weights are already learnt.

In our proposed model, we think that preserving such modularity in the way an agent represents her choices (and preferences), could tackle complexity in the same way humans deal with high-information load problems. While bearing in mind these aspects of human cognition, we will attempt to build a modular representation of an agent's preferences model.

2.3 Preferences Representation

2.3.1 Utility Theory

Utility Theory is the dominant approach to quantify the interests of a decision maker, and providing his preferences across a set of available options [5, 62]. Moreover, the theory provides a way to understand how these preferences change when the decision maker is facing uncertainty with regards to the different outcomes that could be yielded.

Herein, we are interested in a particular type of utility representations, that are inherent to complex decision making problems. In this case, utility representations are known to be defined over a large number of attributes, usually nonlinear, non-monotonic, with an elicitation process that is subject to uncertainty and high-information load.

2.3.2 Multi-attribute Utility Theory

In the case of decisions involving multiple attributes, the main approach is Multi-attribute utility theory (MAUT), which has emerged as an important component of modern Decision Making and Operational Research. As an optimization technique, it has been used in a wide range of areas such as economics, management, social sciences and engineering. In real life, the evaluation of multiple and interdependent attributes tends to make the decision problems more complex, especially when uncertainty is involved. Used with multi-agent systems, MAUT is an efficient way to design decision making mechanisms where autonomous agents interact to achieve their objectives. The agents will attempt to reach an agreement and satisfy their contradictory demands through a bargaining process. In real life situations, agents have to take into consideration multiple attributes simultaneously during their negotiation, for instance, the quality, quantity, delivery time, etc. [73]. There have been several works in the context of multi-attribute utility theory as well as its usage in negotiation, commerce and social interactions. Different approaches and methods were proposed to analyze multi-attribute utilities for contracts construction. For instance, [96] presented the notion of convex dependence between the attributes as a way to decompose utility functions with several attributes. [88] proposed an approach based on utility graphs for negotiation with multiple binary attributes.

Herein, we focus on this type of multi-issue utility functions for they are inherent to realistic decision making problem. Additionally, such preferences' representation could easily be mapped into the modular architecture mentioned in section 2.2.5, by adopting a graphical model as it was done, for instance, in [13, 83].

2.3.3 Probability and Utility Theories

Throughout the provided models of decision analysis, one interesting analogy reflects the similarities between probabilities and utilities. The idea of treating utility functions as probability (density) functions was firstly introduced by [92] through the notion of utility distribution, in which utilities have the structure of probabilities. Most importantly, a symmetric structure that includes both probability distributions and utility distributions was developed.

Herein, we adopt the same assumption as to represent a utility function. Most importantly, and in a larger sense, we adopt the same probabilistic approach when reasoning about the ways a decision maker reasons about his choices. In fact, we assume that cognition could be described as a statistical inference in the sense that the decision maker bases his judgments about the world in terms of probabilities [99]. This view is similar to the idea that cognition acts as some form of Bayesian inference [72]. Particularly, the individual constructs a range of possible choices based on some prior knowledge, where each choice is sampled from that knowledge distribution. Moving from cognition to decision making, we will adopt the same formalism when thinking about the agent choices, whether it is for the representation of the preferences or whenever the agent is reasoning about the opponent's preferences.

2.4 Existing Utility Models

The major problem in decision making is the unavailability of knowledge and, paradoxically, the abundance of knowledge (high-information load implies higher uncertainty). In fact, although realistic situations involve these two aspects, decisions must be made. The most common way to model decision problems is based on the premise of multiple attributes and alternatives. Additionally, it is common that the decision criteria are defined in terms of constraints, whereby a constraint specifies the requirements and the preferences of the decision maker with respect to a number of interdependent attributes.

In the next section, we survey the existing utility models and highlight the differences they have with our proposed representation.

2.4.1 Survey and Assessment

Our goal is to rethink the way preferences could be represented. Adopting the adequate representation gives a solid ground to tackle the complexity that arises in real-world problems. This complexity is clearly projected in the decision makers' preferences, or utility space. In this case, nonlinearity arises in the way the attributes interact as to define the overall utility function.

We address this problem by adopting a representation that allows a modular decomposition of the attributes and the constraints, given the intuition that constraint-based utility spaces are nonlinear with respect to the issues, but linear with respect to the constraints. This allows us to rigorously map the utility space into an issue-constraint hyper-graph. Exploring the utility space reduces then to a message passing mechanism along the hyper-edges by means of utility propagation. On a higher level, the utility propagation scheme acts like a welfare propagation scheme that could transcend the agent whenever a mediator is used. However, our main focus is the agent and her utility maximization, regardless from her environment.

Adopting a graphical representation while reasoning about utilities is not new in the multi-attribute decision making literature. In fact, [13] proposed a model inspired from Bayesian and Markov models, through a probabilistic analogy while representing multi-attribute utilities. In another work by [83], a similar concept was introduced by the notion of Expected Utility Networks which includes both utilities and probabilities. [14] proposed a model which takes into consideration the uncertainties over the utility functions by considering a person's utility function as a random variable, with a density function over the possible outcomes.

The idea of utility graphs could potentially help decomposing highly nonlinear utility functions into sub-utilities of clusters of interrelated items, as in [14] or [6]. Similarly, [88] used utility graphs for preferences elicitation and negotiation over binary-valued issues. [78] adopts a weighted undirected graph representation of the constraint-based utility space. Particularly, a message passing algorithm is used to find the highest utility bids by finding the set of unconnected nodes which maximizes the sum of the nodes' weights. However, restricting the graph and the message passing process to constraints' nodes does not allow the representation to be descriptive enough to exploit any potential hierarchical structure of the utility space through a quantitative evaluation of the interdependencies between both issues and constraints. In [39], issues' interdependencies are captured by means of similar undirected weighted graphs where a node represents an issue. This representation is restricted to binary interdependencies while real negotiation scenarios involve "bundles" of interdependent issues under one or multiple constraints. In our model, we do not restrict the interdependency to lower-order constraints but we allow $p-$ary interdependencies to be defined as an hyper-edge connecting p issues. An hyper-edge is an adequate way to assess the level of interdependency between issues.

We note that our representation borrows the graphical aspect found in CP-Nets [12] without being limited to qualitative preferences. This, while preserving the conditional dependence and the independence statements through the induced dependence graph of a utility hyper-graph. Additionally, our proposed exploration heuristics could significantly reduce the search effort when executing queries for preferential comparisons, namely, the ordering and the dominance queries.

Adopting such graphical representation with its underlying utility propagation mechanism comes from the intuition that elicitation, after all, is a cognitive process that involves concepts and associations, performed by supposedly bounded rational agents. And while bearing in mind the fact that cognitive processes perform some form of Bayesian inference [72], we chose to adopt a graphical representation that serves more as an adequate framework for any preference-based space.

One advantage of using this representation is its scalability in the sense that the problem becomes harder for a large number of issues and constraints. But if we can decompose the utility space, we can exploit it more efficiently. Another way to look at this "connectionist" representation is that it can be clustered in ways that can isolate interdependent components, thus, allowing them to be treated separately and independently form the rest of the problem. This "Ceteris Paribus" approach is useful because it can allow the agents to reach an agreement with respect to subsets of the problem. Once the agents reach a sub-agreement, they can focus on the parts of the problem that require more negotiation.

Another important motivation behind our hyper-graph representation is that it allows a layered, hierarchical view of any given decision making problem. Given such architecture, it is possible to recursively reason over the different layers of the problem according to a top-down approach. Even the idea of issue could be abstracted to include an encapsulation of sub-issues, located in sub-utility spaces and represented by cliques in the hyper-graph. Consequently, search processes can help identify optimal contracts for improvement at each level. This combination of separating the system into layers, then using utility propagation to focus attention and search within a constrained region can be very powerful in the bidding process. A similar idea of recursion in the exploration of utility space was introduced by [79] although it is region-oriented and does not adopt a graphical representation of the utility space.

The main novelty our work is the efficiency of the new representation when optimizing nonlinear utilities. To the best of our knowledge, our work makes the first attempt to tackle the complexity of such utility spaces using an efficient search heuristic that works and outperforms the previously used sampling-based meta-heuristics. Particularly, the novelty is that we exploit the problem structure (as hyper-graph) as well as randomization. Such performance is required when facing the scaling issues inherent to complex decision making. We experimentally evaluated our model using parametrized and random nonlinear utility spaces, showing that it can handle large and complex spaces by finding the optimal contracts.

We note that similar constraint-based settings were defined as a Distributed Constraint Optimization Problem (DCOP), particularly, in [81, 63]. However, we assume that the variables and the $n-$ary constraints could be shared amongst the agents through mediation. Additionally, we focus on the utilitarian aspect of the constraints in the sense that we seek a unified form that might be used to model agents having different economic types. We basically focus on the microscopic aspect (*i.e.*, the agent) as opposed to the multi-agent and distributed aspects found in DCOP. Another analogous example is the decentralized coordination approach found in [27, 28]. We use the same message passing method based on the max-sum algorithm. Instead of using pruning as to reduce the computation that agents must perform when using the max-sum algorithm, we use low-complexity search strategies when performing the message passing in the factor graph. This allows the agent to find the optimal contract points with a low computation and message passing overhead.

Other than the search and optimization aspects of our proposal, we provide a mathematical model that offers a better understanding of complex utility spaces from the economic perspective. In this model, an agent's utility space is built as a composition of sub-utility functions with different economic profiles. This reflects the idea that a decision maker uses different economic types when reasoning about different issues from different domains. Being able to analyze this compositional aspect could improve the quality of the outcomes of any strategic encounter between agents having complex utility spaces.

Particularly, adopting an hyper-graphic representation inspired from factor graphs [70, 76] allowed us to establish an analogy between probabilistic Mixture Models as they are used in Machine Learning, and a purely utilitarian Mixture Model for constraint-based utility approximation. In fact, our proposed Generalized Gaussian Mixture Model (GGMM) could approximate all constraint-based utility functions using one unified form. This form can be used for opponent modeling as well as for the generation of parameterized utility spaces.

Chapter 3

Complexity Study

3.1 Complex Decision Making

We first address a number of questions that will help define complexity as in arises in our decision making settings.

What do we mean by "Complexity"?

We focus on decision making problems that exhibit endogenous bounded rationality [21]. Complexity is due to the search cost that the decision maker is facing whenever he tries to know the entire structure of the problem. The cost of the search is inferred from revealed preferences through choices. We are interested in exploiting the relationship between bounded rationality arising from the search costs and the actual complexity of the problem, independent of the decision maker, and translated as structural complexity. The structural complexity will be assessed in an abstract graphical model representing the decision makers' mind.

Graphical formalism, why?

Adopting a graphical representation allows us to implement the modular aspects mentioned in section 2.2.5 as well as the connectionist view of the problems in hand. This accounts for the high-information load aspect of the problems we are addressing.

How to measure complexity?

Our way of quantifying and measuring complexity is built on the existing interplay between complexity as it arises in graphical models and complexity in its information theoretic formulation through the notion of entropy.

In the next section, we establish the interplay by assuming that cognition (decision making in particular) is a game played between an agent and nature.

3.2 Cognition as a Game of Complexity

3.2.1 Introduction

One of the major characteristics of real-world cognition is the high-information load one is facing, in the sense that the given tasks are informationally demanding and computationally intractable. However, humans are capable of handling these high-cognitive load problems despite the complexity and the large number of possibilities [45]. In the context of several cognitive problems, for instance the *Frame Problem* [59], the *Infinite Regress* problem [11], as well as in the case of *Bounded Rationality* [57], we argue that the limits of human cognition is due to the infinite number of concepts that are made available. Under such circumstances, people are considered to be poor optimizers and ought to use other search strategies [44].

Herein, we pose the problem of modeling an agent's cognitive model subject to bounded rationality and high-cognitive load. Cognition is considered as a game of complexity between nature and the agent, both operating on a cognitive graph. While the first player tries to maximize the complexity by maximizing the entropy, the second player attempts to minimize it. We adopt a view of cognition that is both connectionist and topological. Precisely, we chose to consider the whole topology of the problem, rather than studying a particular topological feature with its finer details (attributes, utilities, probabilities) and the underlying interactions (conditioning, dependence). For instance, [21] adopted another approach based on the extensive form with its decision tree representation as a way to model complexity in endogenous bounded rationality.

The reason why we chose to analyze complexity by considering the topological representation is due to the enormous information content. In fact, the content of the required probability or utility distributions may make computational representations infeasible [25] and Bayesian inference intractable (NP-hard) [72]. Particularly, our topological choice relates to the organization of the agent's mental concepts as a graph. Cognition is therefore performed through a mechanism of activation of concepts, represented by the nodes of the graph [16]. The game of complexity is defined over this graphical structure, with strategies that represent potential probabilistic representations of the agents cognitive space. Precisely, a strategy could be seen as a probability distribution over the degrees of the nodes. Thus the complexity arises implicitly from the difficulty that the agent is facing when exploring this cognitive graph. Although entropy-based measures can perfectly capture the complexity of a graph as it was established in [82, 20], we are mostly interested in the game theoretical rationale behind the usage of such measures. In fact, we assume that the cognitive complexity relates to the strategic interaction of two players, nature and the agent. The agent executes a task using her cognitive model while trying to minimize complexity, and nature attempts to make this cognition difficult by increasing the complexity. We claim that this situation is similar to the information theoretical game studied in [97], whereby the agent (the physicist) is attempting to describe, or code the outcomes of an alphabet given the observations of a physical system. After establishing the game theoretical equilibrium of the cognitive process, we show that for the agent, moving from less pertinent concepts towards pertinent concepts minimizes the complexity of the overall cognitive problem. Consequently, we provide the family of minimax strategies that minimize the complexity of the agent's cognition. In fact, adopting an exploration scheme that follows a power-law distribution tends to lower the cognitive complexity than if the agent adopts a random distribution. This corroborates the general idea that adopting pre-established, or orthodox ideas takes less effort (cognition) than intellectually engaging in the exploration of new ideas, namely "Thinking outside the box".

3.2.2 The Cognitive Model

Herein, we provide the basics of our graphical cognitive model. We define the concepts, associations and how cognition is being conducted as a process.

The Concepts

Providing a theory of concepts is beyond the scope of this section. However, we rely on *Barsalou*'s theory of concepts in his view of conceptual cognition, namely the perceptual symbol systems (PSS) view of cognition [8]. Our reliance on this view is justified to the extent that it is related to the *file model* of cognition. According to the file model, concepts refer to *files* that contain information, such as representations and beliefs about the concepts per se, or more precisely about the entities in the concepts extensions. We adopt a similar representational system, in the sense that a concept is a general type, category or class, which can undergo any number of conceptualization with different representations. For instance, a flower can be conceptualized in several ways with regard to its shape, color, texture, smell, etc. Thus, different representations will be activated depending on the context and goals of the agent. Once these concepts are established in the cognitive graph of the agent, knowledge is accumulated and added to the system. This process is reinforced with time, through new experiences and belief update.

The Cognitive Associations

The associations between the concepts are used to make inferences on the basis of the availability of beliefs related to the concepts and their representations. We can think about it from the perspective of the *file model*, as a file system where each file is labeled and additionally contains notes about the concept. The number of files and their underlying notes can be infinite, although this number could be bounded if the agent has to consider only few concepts in her perceptual or cognitive frame. This does not mean, however, that the agent cannot accumulate a large number of concepts, especially when the cognitive task is performed over a long period of time. Associations are to be understood as the type of assessments that assign *causal relations* between activated concepts when the agent is facing a problem. It could be thought of as the connections between concepts triggered by the usage of natural language. For instance, thinking about the concept of a plane might activate other concepts like the crew, pilots or luggage [16]. This type of association is the one we are about to use according to a connectionist representation. That is, an association is perceived as an edge that connects two concepts in the cognitive graph. Most importantly, the existence or the non-existence of an edge is probabilistically defined based on the strategies of the players.

Cognition Process

Formally, we adopt a graph representation for concepts and associations, and assume that the cognitive process describes how the agent operates on her cognitive graph in order to make decision, inferences, elicitation or more generally cognition. While bearing in mind the fact that cognitive processes perform some form of Bayesian inference [72], we chose to adopt a graphical representation that serves more as an abstraction of any potential quantitative instantiation. For instance, it could be instantiated as a Bayesian network operating on concepts seen as random variables, a CP-net operating on attributes and utility functions, etc. The representation we adopt acts more like an undirected Qualitative Probabilistic Network [101] where an association is defined by one of the influence relations $\{+, -, 0\}$ and the non-association by $\{?\}$ (ambiguity) as in [71]. The choice of the undirected edges is due to the fact that we are only interested in the topology and the complexity of the conceptual graph rather then the pertinence of its concepts. Since the cognitive process is evolving over time, the graph will be indexed by t, as in (3.1).

$$G_t = (V_t, E_t) \tag{3.1}$$

where at t, V_t represents the concepts, and E_t represents the associations between the elements of V_t. The graph could be concretized as an *Erdős-Rényi* Random Graph [84], characterized by the degree distribution $P_{II}(t)$ at t, and defined over n concepts. We define a cognitive process as in Definition 1.

Definition 1. *The cognitive process $\mathscr{P}$, in (3.2), is a stochastic process that operates on a graph of concepts G_t, and where the concepts and the associations are added dynamically by player II according to the distribution $P_{II}(t)$.*

$$\mathscr{P} = \{P_{II}(t) \mid \ t \in T\}. \tag{3.2}$$

In fact, even though P_{II} describes the degree distribution of G, we do not use it directly for the evaluation of the pertinence of the concepts. Instead, we consider a function $e : V \to \mathbb{R}$ that reflects the importance or energy $z_{II}(i)$ of concept i by assigning the number of nodes connected to it (number of possible associations). Since we are dealing with distributions, the energy of each node is normalized *w.r.t* the overall energies as in (3.3). We can eventually enrich (3.3) to include the amount of information/uncertainty $f(i)$ of concept i. Herein, we simply take f as deg, that is, $f(i)$ is the degree of node i. Herein, (3.3) can also be considered as a Boltzmann (Softmax) distribution.

$$e(i) = \frac{f(i)}{\sum_j f(j)} \tag{3.3}$$

Using (3.3) we define another distribution Z_{II}, as in (3.4).

$$Z_{II} = \{ \ e(i) \ \}_{i=1}^{n} \tag{3.4}$$
$$e(i) = z_{II}(i)$$

The sequence Z_{II} could also be seen as the normalized degree sequence of the concepts. At each t, the agent executes a number of actions by adding edges between a number of nodes. We say that the action $\vec{a}$ adds one edge between $x, y \in V$ according to P_{II} in the sense that the newly obtained graph has degree distribution P_{II}. Action $\vec{a}$ will therefore alter the graph G as well as the sequence Z_{II}. We are interested in the choice of P_{II} that has to be chosen by the agent in order to add an edge. More precisely, we want to find the distributions that minimize the entropy of the sequence Z_{II}.

3.2.3 The Game of Complexity

In the following, we formulate the previous cognition process according to a game theoretic approach. The ultimate goal is to analyze the equilibrium of this game of complexity from the entropic standpoint, and provide the optimal strategies.

Game Formulation

The game is a zero-sum (normal form) game between nature (player I) and the agent (player II). Nature tries to make the agent's cognition difficult by increasing its complexity while the agent tries to decrease the complexity of the cognitive process $\mathscr{P}$, defined over the cognitive graph $G(V, E)$. The cognitive game of complexity $\mathscr{G}$ is thus defined as in (3.5).

$$\mathscr{G} = (N,\ G,\ S,\ \Phi) \tag{3.5}$$

$$N = \{I, II\} \tag{3.6}$$

$$G = G(V, E) \tag{3.7}$$

$$S = S_I \times S_{II} \tag{3.8}$$

$$\Phi : S \to \mathbb{R} \tag{3.9}$$

We adopt the same objective function Φ, as in [97], defined as the measure of complexity between nature's strategy $Z_I \in S_I$ and the agent's strategy $Z_{II} \in S_{II}$ as in (3.10).

$$\Phi(Z_I \| Z_{II}) = -\sum_{i=1}^{n} z_I(i) \log(z_{II}(i)) \tag{3.10}$$

(3.10) reflects the difficulty encountered by the agent in exploring the cognitive graph when she adopts the distribution P_{II} (and its normalized degree sequence Z_{II}) and when nature adopts P_I (and its normalized degree sequence Z_I). Now, we define the $\Phi-$entropy S_Φ as the minimum complexity (3.11).

$$S_\Phi(Z_I) = \inf_{Z_{II} \in S_{II}} \Phi(Z_I \| Z_{II}) \tag{3.11}$$

Thus the entropy is defined as the minimal complexity.

The strategies

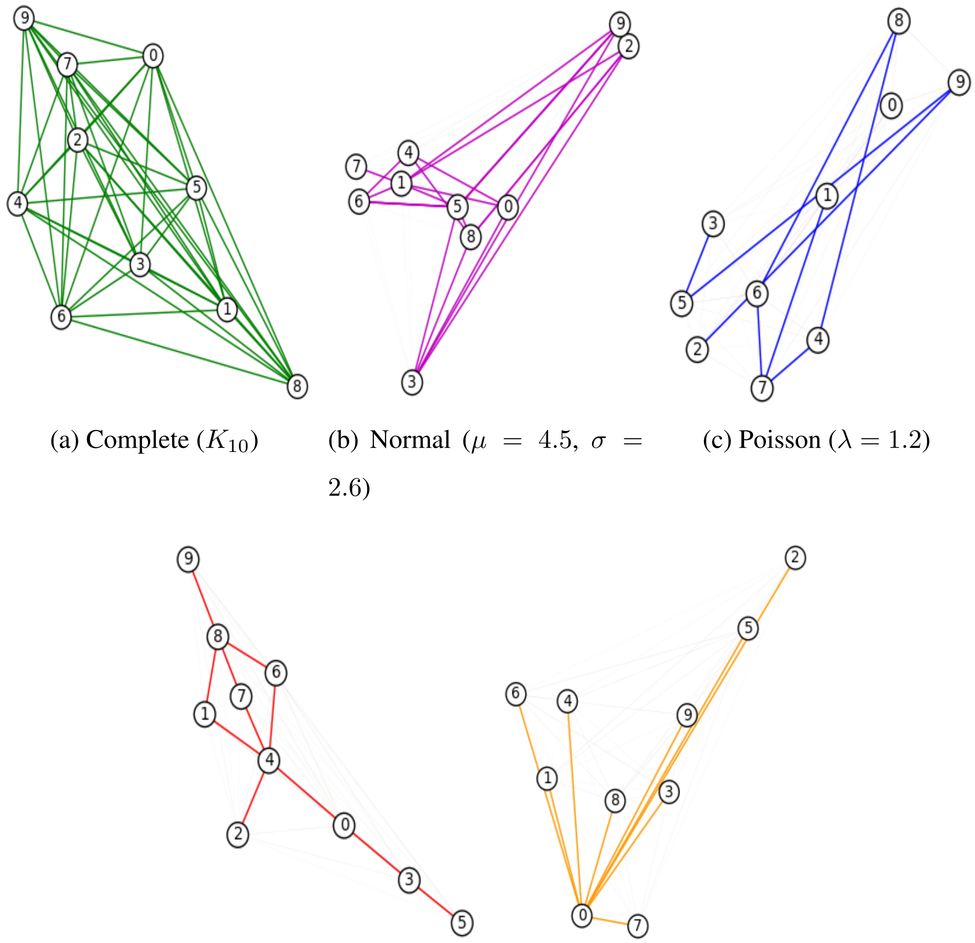
As we have mentioned previously, agent's strategy $Z \in S_{II}$ is a normalized degree sequence (3.4), but the initial form of Z is defined as the degree distribution P. This is due to the fact that we are dealing with a graph G and that its topology is better defined by P. Herein, we propose concrete examples of strategies as well as on how their complexities are found, thus, how players' optimal strategies are established.

A strategy is a particular way to traverse or explore a graph by moving from a node to another. For instance, Figure 3.1 shows 5 different ways that can be used to explore a graph composed of 10 nodes. Each exploration strategy corresponds to a particular degree distribution that can be mapped onto the graph. In this example, we distinguish the complete (or uniform) distribution (Figure 3.1(a)), the normal distribution (Figure 3.1(b)), the poisson distribution (Figure 3.1(c)), the power-law distribution (Figure 3.1(d)), and the star distribution (Figure 3.1(e)).

$$P \begin{cases} P_1 = Uniform(n-1) \\ P_2 = Normal(\mu \in [2,4], \sigma \in [0,2]) \\ P_3 = Poisson(\lambda \in [2,6]) \\ P_4 = Power - law(\alpha = 2.3) \\ P_5 = Star \end{cases} \tag{3.12}$$

In Figure 3.2(a), we provide another example of 5 degree distributions $P_{i\in[1,5]}$, defined in (3.12), and mapped on a graph of 100 nodes. The distribution-color mapping is $(P_1 : green, P_2 : purple, P_3 : blue, P_4 : red, P_5 : orange)$. We recall that the strategies $Z_{i\in[1,5]}$ are built from the degree distributions $P_{i\in[1,5]}$.

If we take the corresponding normalized degree sequences $Z_{i\in[1,5]}$ and their ordered entropies $H(Z_{i\in[1,5]})$, we get Figure 3.2(b). The uniform strategy yields a complete graph (K_{n-1}) with the highest complexity, *i.e.*, highest entropy. On the contrary, the star (S_{n-1}) strategy has the lowest complexity, which is due to the fact that all concepts are connected to one single pertinent concept acting like a hub. If we increase the number of hubs to few hubs, we get the power-law strategy which is ranked right above the star strategy. The normal and poisson strategies fluctuate between



(a) Complete (K_{10}) (b) Normal ($\mu = 4.5$, $\sigma = 2.6$) (c) Poisson ($\lambda = 1.2$)

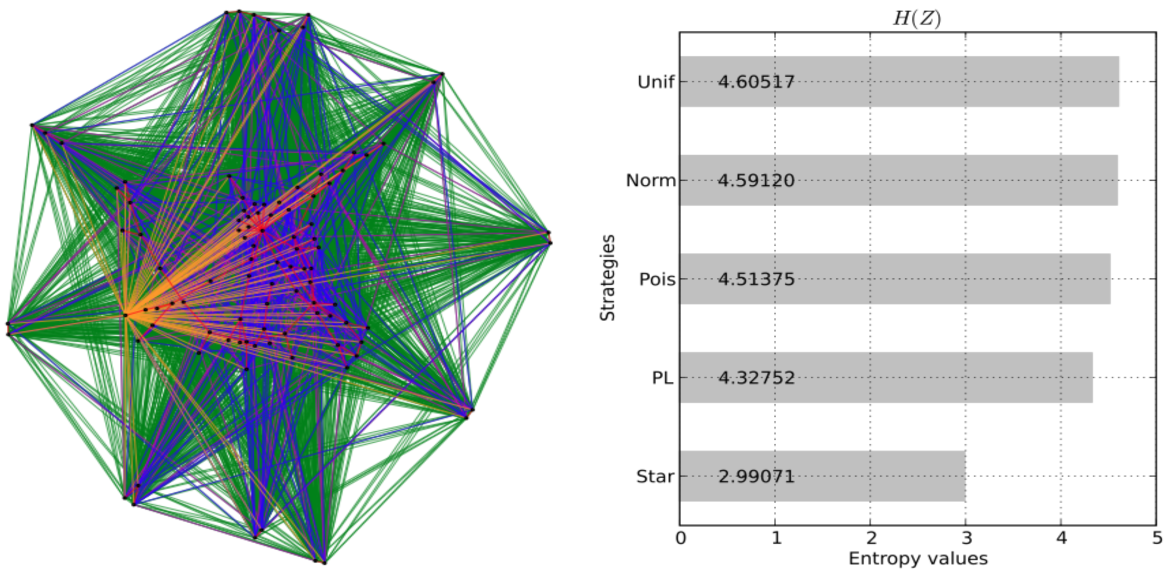
(d) Power-law ($\alpha = 2.3$) (e) Star

Figure 3.1: Example of 5 exploration strategies over 10 concepts

P_1 and P_4 in terms of complexity, relatively to the values of μ, σ and λ. The uniform distribution P_1 and the star distribution P_5 act as complexity bounds. Now, if we consider the zero-sum game $\mathscr{G}$ from the perspective of the *minimax-maximin* viewpoint, both players will behave in opposite ways. On the one hand, nature tries to achieve high complexity. Therefore, its strategy $Z_I^* \in S_I$ is optimal if (3.13) holds.

$$S_\Phi(Z_I^*) = \sup_{Z_I \in S_I} S_\Phi(Z_I) = S_\Phi^{max} \tag{3.13}$$

where S_Φ^{max} is the maximal entropy. On the other hand, the agent tries to achieve low complexity with Φ as her objective function. Therefore, she associates a risk R_Φ to any strategy $Z_{II} \in S_{II}$



(a) The cognitive graph

(b) Strategies' entropies

Figure 3.2: Cognitive Graph and Strategies' Entropies

as in (3.14).

$$R_\Phi(Z_{II}) = \sup_{Z_I \in S_I} \Phi(Z_I \| Z_{II}) \tag{3.14}$$

The agent's optimal strategy Z^*_{II} is the minimum risk value R^{min}_Φ as in (3.15).

$$R^{min}_\Phi = \inf_{Z_{II} \in S_{II}} R_\Phi(Z_{II}) = R_\Phi(Z^*_{II}) \tag{3.15}$$

Given the two optimal strategies (3.13) and (3.14) we propose to find the equilibrium of $\mathscr{G}$ in the next section.

The equilibrium

Once the strategies are defined, the equilibrium is given by considering both thermodynamical and game theoretical equilibriums and their consistency with the maximum entropy thinking. If the *minimax-maximin* inequality (3.16) holds, then the game is in equilibrium and the common value is the value of the game. Hence, the maximum entropy value equals the minimum risk equals the minimum guaranteed complexity.

$$S_{\Phi}^{max} \leq R_{\Phi}^{min} \tag{3.16}$$

The common value is the complexity that the agent is getting, with its underlying graphical representation. For instance, let us consider the case where the available strategies of the players are either uniform or power-law. The basic way to represent this situation is to consider the random payoff matrix in Table 3.1 where the entries are computed from the generated strategies $Z_I,\ Z_{II} \in \{Power-law(\alpha),\ Uniform(n-1)\}$.

		Z_{II} **Power-law**	Z_{II} Uniform
Z_I	Power-law	$\Phi(Z_I \| Z_{II})$	$\Phi(Z_I \| Z_{II})$
Z_I	**Uniform**	$\Phi(Z_I \| Z_{II})$	$\Phi(Z_I \| Z_{II})$

Table 3.1: Random payoff matrix

In this setting, nature plays the strategy that yields the highest possible complexity for the agent, the uniform strategy. On the other hand, the agent will chose the strategy that guaranties the minimal possible complexity while nature is trying to cause her damage by providing the most complex configuration. Thus the pure-strategy Nash equilibrium is $(Uniform,\ Power-law)$. When more than two strategies are available, the concept of mixed strategies becomes more compelling than the concept of pure strategies. Then, it is unlikely that the given random zero-sum game has solutions in pure strategies. That is, assuming that the shared strategies' support is of dimension $m \gg 2$, the probability that the game has saddle point involving pure strategies is $\lim_{m\to\infty} (m!)^2/(2m-1)! = 0$ [46]. Now, if players adopt randomized strategies and choose to pick the one with the highest probability, the agent's best response will remain the same while nature's strategy will be defined according to the λ, μ, σ parameters and their effect on maximizing the entropy of the corresponding distributions.

The equilibrium point $(Uniform,\ Power-law)$ reflects the agent's minimal amount of complexity necessary for cognition. The underlying game theoretic optimality is consistent with the optimal dendritic wiring described by the power-law and the tree-like shapes found in neural circuits [19]. This relates as well to the self-organized criticality (SOC) in dynamical systems whereby the system's macroscopic behavior displays a spatial scale-invariance characteristic of the critical point. Thus, the emergence of the power-law distribution when the system is in a self-organized critical state [103].

3.2.4 Optimal Cognition Strategy

In the following section, we propose to find the strategy that minimizes the cognition complexity for the agent, by finding the right action(s). Moreover, the optimal strategy will be used to justify the optimality of the power-law strategy provided in the previous section. We start by considering $G(V, E)$, the cognitive graph, V_-, the subset of low degree nodes and V_+, the subset of high degree nodes (3.17). We take ϵ as a small natural number.

$$\begin{aligned} V_- &= \{v \in V : \ |deg(v) - (n-1)| \gg \epsilon\} \\ V_+ &= \{v \in V : \ |deg(v) - (n-1)| < \epsilon\} \end{aligned} \tag{3.17}$$

Consider the process of creating a new copy of the graph $G(V, E)$ by adding one edge between $x, y \in V$, namely the action $\vec{a} : (x \sim y)$. Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be the result of such process with the new sets E_1, E_2 generated by $\vec{a}_1$ and $\vec{a}_2$, executed on $s, a_- \in V_-$ and $a_+ \in V_+$, as in (3.18).

$$E_1 \leftarrow E \cup (s, a_-) \quad \text{result of} \quad \vec{a}_1 : (s \sim a_-) \tag{3.18a}$$

$$E_2 \leftarrow E \cup (s, a_+) \quad \text{result of} \quad \vec{a}_2 : (s \sim a_+) \tag{3.18b}$$

Now consider Z, Z_1, Z_2, as the normalized degree sequences (3.4) relative to G, G_1 and G_2. The entropy of Z conditioned on $\vec{a}_1$ and $\vec{a}_2$ gives us the entropies of Z_1 and Z_2 as in (3.19).

$$H(Z|s \sim a_-) = H(Z_1) \tag{3.19a}$$

$$H(Z|s \sim a_+) = H(Z_2) \tag{3.19b}$$

In the following, we establish that $H(Z_1) > H(Z_2)$, and that $\vec{a}_2$ dominates $\vec{a}_1$.

Theorem 1. (Choice complexity)

Let $G = (V, E)$ be a cognitive graph with n nodes. Let,

1. *Z be the normalized degree sequence of G defined in (3.4)*
2. *V_- and V_+ be the subsets defined in (3.17).*
3. *s, a_-, a_+ be the different nodes defined as : $s, a_- \in V_-$, $a_+ \in V_+$ and $(s, a_-), (s, a_+), (a_-, a_+) \notin E$.*

If $Z|x \sim y$ is the normalized degree sequence of G', image of G by the process of drawing an edge between x and y, then, the complexity arising from drawing an edge from s to a_- is higher than if we draw an edge from s to a_+.

We shall write: $H(Z|s \sim a_-) > H(Z|s \sim a_+)$.

The proof is provided in A.2.

We can interpret Theorem 1 as follows: "In a cognitive graph, moving toward pertinent concepts minimizes the complexity". Intuitively, it could be seen in the human tendency to quickly embrace pre-established ideas and belief systems (V_+) rather than engaging in the type of reflection or investigation that could be tedious, complex and time consuming. Especially, when some ideas have not yet been investigated (V_-). Pre-established ideas act like the accumulated knowledge or memory of the system, which creates high degree concepts. A game theoretical and utilitarian way to interpret Theorem 1 is given in Corollary 1.

Corollary 1. (GT interpretation of Theorem 1)

Let $\mathscr{G} = (N,\ G,\ S,\ \Phi)$ be a cognitive game of complexity and Z the normalized degree sequence of G. For player i, let $\vec{a}_1 : (s \sim a_-)$ and $\vec{a}_2 : (s \sim a_+)$. If $H(Z|\vec{a}_1) > H(Z|\vec{a}_2)$, then $\vec{a}_2$ dominates $\vec{a}_1$ and $u_i(\vec{a}_2) > u_i(\vec{a}_1)$, with u_i defined as player i's subjective utility function.

Furthermore, it is possible to reinterpret Theorem 1 in the light of the strategies in 3.2. This is done in Corollary 2.

Corollary 2. (Topology of the convergence strategies)

Let $\mathscr{G} = (N,\ G,\ S,\ \Phi)$ be a cognitive game of complexity and Z the normalized degree sequence of G. For player i, let $\vec{a}_1 : (s \sim a_-)$ and $\vec{a}_2 : (s \sim a_+)$. Consider the 2 limits $\mathscr{Z}_1$ and $\mathscr{Z}_2$ of the sequence Z_t conditioned on $\vec{a}_1$ and $\vec{a}_2$, as in (3.20).

$$\mathscr{Z}_i = \lim_{t\to\infty} Z_t \mid \vec{a}_i \quad \forall i \in \{1, 2\} \tag{3.20}$$

Then, $\mathscr{Z}_1$ describes a randomly distributed graph, while $\mathscr{Z}_2$ corresponds to a power-law distributed graph.

We will describe in the experiments an instance of Corollary 2 whereby nature uses $\vec{a}_1$ and the agent uses $\vec{a}_2$. The resulting cognitive graph will contain both the random and the power-law strategies.

3.2.5 Experiments

Experiment Design

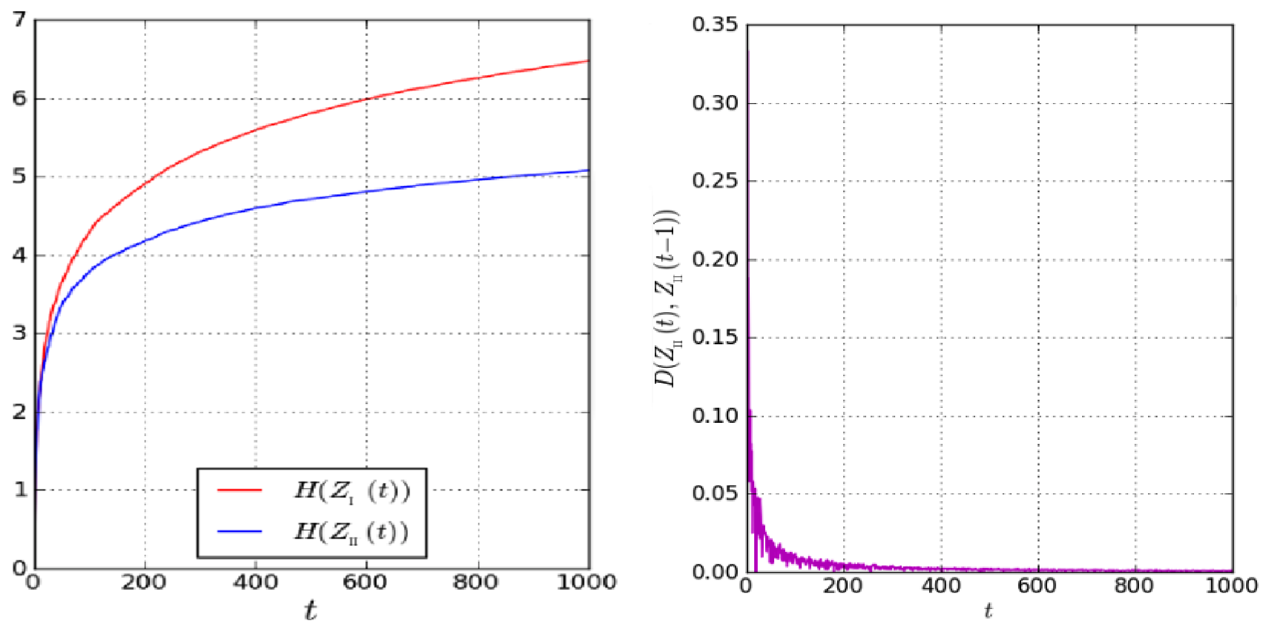
Consider the process $\mathscr{P}'$ whereby (3.1) is built by adding one node at each time step $t \in [1, T]$. Furthermore, each one of the players will add m_t edges according to a specific distribution: $P_I(t)$ for nature and $P_{II}(t)$ for the agent. In the following, we take $m_t = 5\ \forall t$ and $T = 10^3$. $P_I(t)$ and $P_{II}(t)$ are altered and extended at each time step according to a specific attachment method. The edges in $P_{II}(t)$ are added according to a preferential attachment. That is, we start by selecting a random node a, then another node b is selected from $\alpha\%$ of the high degree nodes (V_+). The edges in $P_I(t)$ are added by random selection of nodes without any preferential attachment. This process is repeated m_t times. At each round t, we propose to measure $H(Z_{II}(t))$ and $H(Z_I(t))$, the entropies (complexities) of the sequences $Z_{II}(t)$ and $Z_I(t)$. We are interested in the convergence of the sequence $Z_{II}(t)$ mirroring the asymptotic behavior of $P_{II}(t)$ and the corresponding graph. To this end, we consider $D(Z_{II}(t),\ Z_{II}(t-1))$ where the measure D is defined as in (3.21).

$$D(p,\ q) = \frac{1}{2}\sum_{i=1}^{n} \mid p_i - q_i \mid \tag{3.21}$$

Results and Discussions

Building G_t with two different attachment methods yields two different distributions $P_{II}(t)$ and $P_I(t)$. Herein, it is possible to relate the current experiment to the previous sections, by noticing that the agent's and nature's attachment methods correspond respectively to $\vec{a}_1 : (s \sim a_-)$ and $\vec{a}_2 : (s \sim a_+)$ defined in Corollary 2.

The behavior of $H(Z_{II}(t))$ and $H(Z_I(t))$ is in Figure 3.3(a). Given the choices of the strategies, and as $t \to T$, the complexity of $Z_{II}(t)$ falls below the complexity of $Z_I(t)$. The agent's representation of G could be looked upon as a process evolving over time and described by the sequence $\{Z_{II}(t) | t \in [1, T]\}$.



(a) $H(Z_{I,II}(t))$

(b) Convergence of $Z_{II}(t)$

Figure 3.3: Entropies and Convergence

As it is shown in Figure 3.3(b), the sequence approaches one unique representation as the graph grows with t. This describes the fact that the pertinent concepts are reinforced with time and that they start acting like memory hubs, which is inherent to scale-free graphs. However, in our context, this means that in the limiting strategy $\mathscr{Z}$ only a few concepts will be pertinent to the agent. Most importantly, the convergence of the sequence towards a unique topological representation (3.22) shows that we are certain about reaching this limit, even though the entropy (uncertainty) of the process is increasing logarithmically as in Figure 3.3(a).

$$\lim_{t\to\infty} D(Z_{II}(t),\ Z_{II}(t-1)) = 0 \ \Rightarrow\ \lim_{t\to\infty}\{Z_{II}(t)\} = \mathscr{Z} \tag{3.22}$$

This allows us to reinterpret the process with regard to uncertainty, additionally to cognitive complexity.

In fact, this uncertainty is about the way we assign the importance, or energy (3.3) to the concepts. Under uncertainty (or lack of knowledge), we do not know which concept is more important, therefore the assignment will follow a maximal entropy distribution. This is the situation of high complexity (chaos, disorder, randomness) discussed above as the goal of nature. The highest uncertainty value $\log(n)$ is achieved in the case of a uniform strategy (complete graph K_{n-1}). On the other hand, and under certainty, the entropy of the importance assignments will be minimized. This reflects the situation of *safe* certainty and order that the agent is seeking. For instance, under a star topology (S_{n-1}), the uncertainty is the lowest with a value of $\frac{1}{2}\log 4(n-1)$ since the total energy is concentrated in one concept [1]. Given any strategy Z, its complexity (respectively uncertainty) is bounded between the lowest complexity of the star topology and the highest complexity of the complete topology (3.23).

$$\frac{1}{2}\log 4(n-1) \leq H(Z) \leq \log(n) \tag{3.23}$$

For instance, the distributions $P_{2,3,4}$ used in section 3.2.3., all fall within this range. Under bounded rationality, non-complex problems [21] could be explained according to our topological rationale of complexity in the sense that the optimal strategy $(Uniform,\ Power-law)$ falls within the range (3.23). Most importantly, such problems tree structures are characterized by a low treewidth, which suffices in solving the Most Simple Explanation (MSE) problem with a low margin of error [71]. This is corroborated by the fact that scale-free graphs have a lower treewidth compared to the poisson or complete graphs [41].

[1] If the hub is v_s then $e(v_s) = n-1$ and $e(v_i) = 1\ \forall i \neq s$

3.2.6 Conclusion

We have modeled cognition as a game of complexity between nature and an agent in situation of bounded rationality. The complexity as well as the game theoretical equilibrium are defined in terms of the entropy. We prove that adopting a power-law distribution while exploring the cognitive graph tends to minimize the cognition cost. Furthermore, we analyze the interplay between cognitive complexity and uncertainty.

As an important research issue to be further investigated, we think about the case of directed cognitive graphs, by considering the direction of the concept activation as in [16]. This will allow us to fully characterize the graphical structures that are likely to trigger "Fast and Frugal" heuristics [45]. Our work could also relate to [72] in the sense that we could identify the cognitive topologies involved in the 'approximation' models of human cognition. Thus, bridging the gap between Bayesian computational intractability on the one hand and the ease with which humans can make the inferences that are modeled by Bayesian models.

3.3 Complexity, Entropy and Uncertainty

3.3.1 Introduction

One of the main characteristics of real-world cognition is the high-information load one is facing, in the sense that the given tasks are informationally demanding and computationally intractable. However, humans are capable of handling these high-cognitive-load problems, despite the complexity and the large number of possibilities [43]. In the context of several cognitive problems, for instance the *Frame Problem* [59], the *Infinite Regress* problem [11], as well as in the case of *Bounded Rationality* [57], we argue that the limits of human cognition is due to the infinite number of concepts that are made available.

In this part, we attempt to prove this claim by drawing a relation between uncertainty and cognition in situations subject to an infinite number of co-activated concepts. The relation is made in the sense that uncertainty is involved in the cognitive task, with regard to the solution to be found and the evolution of the process. More precisely, we investigate the reasons of this intractability and limits, in the light of a quantitative measure of uncertainty. Through the evaluation of the entropic behavior of a cognitive process evolving over time, we provide an insight on the relations between cognitive complexity and the extent to which solutions could be found. We analytically show that under such assumption of large cognitive space, the uncertainty of the overall cognitive process will keep on increasing, even though we attain complete certainty in the final stage of the process.

To this end, we propose a cognitive model with a graphical representation inspired from the work of [16] on heuristics. In his model, the organization of the mental concepts has a graph-like structure, and where the cognition is done through a mechanism of activation or priming of the given concepts. Several viewpoints have been presented in literature addressing different aspects of the problem of cognition, and how the concept of infinity arises and affects it. For instance, the frame problem refers to a problem where a cognitive agent faces an infinite supply of potentially relevant and irrelevant information, and has to process relevant information in an adaptive and intelligent manner.

The problem of finding the relevant information while ignoring everything else is the main scope of the frame problem. The issue of framing is an ubiquitous problem in real life, and could be found in rationality, politics, ethics, especially in situations subject to a high-cognitive load [94]. Another related situation is the epistemological problem of the infinite regress problem [11] which generally rises in any situation where a statement has to be justified. Herein, we relate to the same problem, and more precisely on how knowledge is added over time and how it affects the uncertainty of the overall cognitive process. By mirroring the notion of infinite regress with the process of conceptualization and association, we can think about the cases where the agent is neither constrained by time nor stopped by the undecidability of a problem. In this case, he will have to face an infinite number of concepts, which makes the problem intractable, and eventually with complex associations. It is with regard to these limits of human cognition that we evaluate the entropic behavior of a cognitive process evolving over time. Hence, we provide an insight on the relations between cognitive complexity and the extent to which solutions could be found. We prove that the larger the cognitive space is, and the more uncertain the evolution of the process can be, despite the certainty of the final outcome.

3.3.2 The Cognitive Model

Concepts

As in the last part, we rely on *Barsalou*'s theory of concepts in his view of conceptual cognition, namely the perceptual symbol systems (PSS) view of cognition [8]. Our reliance on this view is justified to the extent that it is related to the *file model* of cognition. According to the file model, concepts refer to *files* that contain information, such as representations and beliefs about the concept per se, or more precisely about the entities in the concepts extensions. We adopt a similar representational system, in the sense that a concept is a general type, category or class, which can undergo any number of conceptualization with different representations. For instance, a flower can be conceptualized in several ways with regard to its shape, color, texture, smell, etc. Thus, different representations will be activated depending on the context and goals of the agent. Once these concepts are established in the cognitive space (or memory) of the agent, knowledge is accumulated and added to the system. This process is reinforced through new experiences and beliefs update.

Associations

The associations between the concepts will be used to make inferences on the basis of the availability of beliefs related to the concepts and their representations. We can think about it from the perspective of the file model, as a file system where each file is labeled and additionally contains notes about the concept. The number of files and their underlying notes can be infinite, although this number could be bounded if the agent has to consider only few concepts in his perceptual or cognitive frame. This does not mean, however, that the agent cannot accumulate a large number of concepts, especially when the given cognitive task is performed over a long period of time. The association are to be understood as the type of assessments that assign causal relations between activated concepts when the agent is facing a problem. It could be though of as the connections between concepts triggered by the usage of natural language. For instance, thinking about the concept of a plane, might activate other concepts like the crew, pilots or luggage [16]. This type of associations is the one we are about to model, based on a graphical representation, as it is the most intuitive and trivial representation.

To cite another example, we take the situation of utility function assessment as a method for preferences elicitation [61]. In fact, the same structure of concepts and associations is mirrored, although in utility assessment, we mostly deal with attributes and dependences. Furthermore, the notion of infinite regress can be re-defined in an utilitarian way, as to the specification of the attributes needed for the assessment of a utility function and the description of the consequences. Therefore, we can state that the usage of an attribute a_1 (analytically defining $U_{a_1}(a_1)$), requires a set $\pi[a_1]$ of attributes upon which a_1 depends and hence $U_{a_1}(a_1) = U(a_1|\pi[a_1])$. Similarly the utility of the attribute a_2 requires another set $\pi[a_2]$ as to determine $U_{a_2}(a_2) = U(a_2|\pi[a_2])$,... the utility of a_n requires $\pi[a_n]$ so that $U_{a_n}(a_n) = U(a_n|\pi[a_n])$, with $n \to \infty$. Assuming that an infinite regress arises when the justifications of propositions are required, we can say that in the case of utility assessment, the proposition to be justified refers to the comprehensibility of the attribute, *.i.e*, its appropriateness on theoretical grounds [61]. When adopting this mode of attributes (concepts) exploration and inference (associations), a question arises as to what extent this process might be possible when the agent is operating over the concepts recursively and ad infinitum. Next, we attempt to characterize this situations from the certainty standpoint.

The Formal Model

Formally, we adopt a graph representation for the concepts and the associations, and assume that the cognitive process operates on this graph. However, unlike the usage of directed graphs in [16], we assume our representation is undirected, since we are only interested in the complexity of the conceptual graph rather then the pertinence of its underlying concepts. Since the cognitive process is evolving throughout time, the graph will be indexed by t, as in (3.24).

$$G_t = (V_t, E_t) \tag{3.24}$$

where at time t, the nodes V_t represent the concepts, and the edges E_t represent the associations between the concepts. Our cognitive model will be concretized as an *Erdős-Rényi* Random Graph [84], characterized by the degree distribution $P_n^{a,b}$ defined over N concepts. The parametrization values a, b and n will be justified later for treating the asymptotic case and the entropy. We note that building our graphs from a general and parametrized degree distribution has the property of making the number of edges (associations) fluctuate according to the parameters a, b and n. Thus, it will be devoted to elaborating the characteristics of the cognitive processes with regard to uncertainty and the behavior of entropy. To this end, we begin by defining a cognitive process as follows.

Definition 2. *A cognitive process P operates on a graph of concepts G_t, where concepts and associations are added and updated dynamically by the agent. P is a stochastic process defined by $P_n^{a,b}$ as in (3.25).*

$$P = \{P_n^{a,b}(t) \mid \; t \in T\}. \tag{3.25}$$

We note that during this process, the agent observers an objective world, and constructs his own model of it, namely G_t, or the object of the process P. We take the hypothesis that the agent is observing an objective, external set of an infinite number of concepts represented by an infinite random graph, *i.e. an objective cognitive space* defined as following.

Definition 3. *The objective cognitive space $\mathcal{E}$ is the graph $G = (V, E)$ of the real objective concepts available to the agent, who builds his subjective cognitive graph from it. The size of $\mathcal{E}$ is $N = |V|$, and tends to infinity.*

The graph G_t mentioned in Definition 1 is just the agent's reduced and subjective representation of $\mathcal{E}$. We also notice that the agent has a frame, or a window, within which he can identify and represent the graspable concepts. We define it as following.

Definition 4. *The agent's cognitive frame $\mathcal{F}$ is built given the concepts of $\mathcal{E}$. It is a subset of $\mathcal{E}$ and has n concepts ($n \ll N$). The agent is building $\mathcal{F}$ by adding more concepts and associations as time goes on.*

Another notion we will use is the notion of *associative potential* of an agent, defined as following.

Definition 5. *For a given cognitive task, the associative potential, namely k, is the maximal number of associations, that an agent could possibly make within his cognitive frame.*

The choice of the maximal number of associations is due to the fact that if the agent is capable of making an association between one concept and k other concepts, he can obviously build $k-1$, or $k-2$,. . . , or 1 association(s) with the first concept. For instance, if agent 1 has an associative potential k_1 and the agent 2 has an associative potential k_2 with $k_2 > k_1$, then the agent 2 will have a higher capacity to create associations, especially if $k_1 = 1$. As we will see in the next sections, the associative potential could also be defined as the rank of 1 in a deterministic degree distribution.

3.3.3 Abstract Cognition Algorithm

Herein, we provide an example on how the agent could exploit his cognitive graph in order to make decision, inferences, elicitation or more generally cognition. We propose an algorithm that describes a generic heuristic that explores the cognitive subspace $\mathcal{F}$ available to the agent. This heuristic is driven by the importance (pertinence) of each concept, illustrated by its corresponding degree. Thus, the concepts with high degrees will be considered as Pertinent Concepts [16] to the task in hand, and must be taken into consideration in the proper cognition task. Herein, we don't specify a precise cognitive task, but we define an abstract routine f that could be instantiated as a specific treatment on a number of concepts in $\mathcal{F}$. For instance, f could be instantiated as a probability function, whenever the conceptual graph is instantiated as a Bayesian network, and operating on concepts seen as random variables. It could also be a utility function operating on attributes with the corresponding outcomes [12], etc. Therefore, Algorithm 1 operates on an abstract representation of the concepts, since we are only interested in the topological aspects of the cognitive process. Initially, the agent starts by activating one concept ι, and then explores his cognitive space by sequentially activating the surrounding pertinent concepts. The number of activated pertinent concepts is characterized by the length σ of the Random Walk that starts from the concept ι. For each concept, the algorithm selects the neighbor $m \in Nei$ with the highest degree. After saving the current concept ι in Vn (with $Vn \subset \mathcal{F}$), the algorithm moves to the concept m and repeats the same task of finding the neighbor with the highest degree.

Algorithm: Abstract Cognition

Input: $G_t(V, E)$: agent's conceptual graphs, ι:initially activated concept ($\iota \in V$), σ:number of steps of the algorithm, f:instantiated routine

Output: Execution's result of the routine f

1 **begin**
2   **for** $i \leftarrow 1$ **to** σ **do**
3         $Nei \leftarrow$Neighbors(ι);
4         $Ns \leftarrow$SortByDecreasingDegree(Nei);
5   **while** *True* **do**
6   **if** *Size*(Nei) $= 0$ **then**
7   Exit;
8   **else**
9                 $m \leftarrow Ns.pop()$;
10   **if** $m \in Vn$ **then**
11   continue;
12   **else**
13   break;
14         $Vn.append(\iota)$;
15         $Activate(\iota)$;
16         $f(\iota \mid Vn \backslash \iota)$;
17         $\iota \leftarrow m$;

Algorithm 1: Abstract Cognition

The algorithm can be executed at any time t of the evolution of the cognitive process P. Furthermore, the number of neighbors $|Nei|$ of a concept ι that the agent can apprehend is limited by the associative potential k of the agent. For instance if $k = 5$, the agent can at least find 5 neighbors of ι, if they do exist. The way the routine f is executed in line 19 reminds us of the example of utility assessment or as in belief propagation in Bayesian updating.

3.3.4 Certainty Evolution

In this section we start by describing the notion of convergence relative to a degree distribution, given the parametrization defined in (3.25). Then, we define the notion of degree distribution support which will be used to characterize the entropy measure. Finally, we present the analysis of the entropy of a cognitive process by showing the behavior of entropy whenever the support of the degree distribution of the underlying process is countably infinite.

Conceptual Graph Convergence

We assume that the convergence of conceptual graphs is mirrored by the convergence of its representative degree distribution. Therefore, we start by defining the distance between two degree distributions. For this, we could rely on the total variational distance as in (3.26).

$$D_V(P_1, P_2) = \sum_j |P_{1,j} - P_{2,j}| \tag{3.26}$$

where D_V stands for the variational distance, and $P_{i,j}$ is the j^{th} element of P_i. For P_1 and P_2 having respectively different dimensions m_1 and m_2, (3.26) becomes (3.27).

$$D_V(P_1, P_2) = \sum_{j=1}^{m_1} |P_{1,j} - P_{2,j}| + \sum_{j=m_1+1}^{m_2} |P_{2,j}| \tag{3.27}$$

We can also use the *Kullback-Leibler* divergence as in (3.28).

$$D_{KL}(P_1, P_2) = \sum_j P_{1,j}\; ln(\frac{P_{1,j}}{P_{2,j}}) \tag{3.28}$$

where we adopt the convention $D_{KL}(P_1, P_2) = 0$ if $P_{2,k} = 0$ but $P_{1,k} > 0$ for some k. Moreover, based on the *Pinsker*'s inequality [100] we have (3.29).

$$\frac{1}{2}\,[D_V(P_1, P_2)]^2 \leq D_{KL}(P_1, P_2) \tag{3.29}$$

Both divergence (3.28) and the variational distance (3.26) can be used as measures of the difference between two degree distributions. However, *Pinsker*'s inequality has the important implication that for two degree distributions P_1 and P_2, if $D_{KL}(P_1, P_2)$ is small, then so is $D_V(P_1, P_2)$. Furthermore, for a sequence of degree distributions $P_{(n)}$, as $n \to \infty$, if $D_{KL}(P, P_{(n)}) \to 0$, then $D_V(P, P_{(n)}) \to 0$. In other words, the convergence in divergence is a stronger notion of convergence than the convergence in variational distance. Thus, we will use the convergence measures (3.28) as to define the continuity of the *Shannon* entropy, in the sense that we study the convergence of a sequence of degree distribution as well as their entropies. Given a sequence of degree distributions $P_{(n)}$, we can consider the sequential decision problem in which a Bayesian agent is observing an objective cognitive space E, and updating his subjective cognitive space represented by the conceptual graph G_t. The graph G_t is generated by the successive realizations of a discrete stochastic process $\{P_{(n)} | \, n \in T\}$, indexed by n, and where n varies over a time index set T. We assume that the process will converge to the limiting degree distribution P_l, as in (3.30).

$$P_{(n)} \to P_l \tag{3.30}$$

This notion of convergence reflects the idea that we expect to see the next degree distribution in $P_{(n)}$ to become better and better modeled by P_l. The yielded convergence is expressed by the limit (3.31).

$$\lim_{n \to \infty} P_{(n)} = P_l \tag{3.31}$$

(3.31) will be used to define the continuity of the *Shannon* entropy on degree distributions.

Degree Distribution Entropy

The *Shannon* entropy measures are functions mapping a probability distribution to a real value. They can be described as the measure of uncertainty about a discrete random variable X having a probability mass function p. We define it as following.

Definition 6. *The entropy H(X) of a random variable X is*

$$H(X) = -\sum_x p(x) \log p(x) \tag{3.32}$$

Herein, summation is over the support of the considered degree distributions. *Shannon* entropy measures the spread of a probability distribution. In the case where the concepts are finite, the *Shannon* entropy measures are continuous function. We propose to focus on the case where the entropy measure is applied to degree distributions with a countably infinite set of concepts. The focus on the infinite case goes hand in hand with the motivations we have stated in the introduction. That is, the understanding highly complex cognition, concretized by a countably infinite number of concepts. We are interested in studying the continuity of H with respect to the distance measures we established in the section 3.3.4. For instance, entropy is discontinuous with respect to the *Kullback-Leibler* divergence [50]. We propose to define the continuity of a function f that will be lately extended into the entropy measure H.

Definition 7. *Let π_k be the set of all possible degree distributions on N nodes, and let $P \in \pi_k$.*
$f : \pi_k \to [0, 1]$ is continuous at P if, given any $\epsilon > 0$, $\exists \delta > 0$ such that:
$\forall P' \in \pi_k : D_{KL}(P, P') < \delta \Longrightarrow |f(P') - f(P)| < \epsilon$.

If f fails to be continuous at P, then we say that f is discontinuous at P. Given the notion of convergence we defined in section 3.3.4, we can provide the following definitions of the discontinuity of the function f.

Definition 8. *Let π_k be the set of all possible degree distributions on N nodes. Let $P \in \pi_k$.*
$f : \pi_k \to [0, 1]$ is discontinuous at P if there exists a sequence of degree distributions $P_{(n)} \in \pi_k$ such that:

$$\lim_{n\to\infty} D_{KL}(P_{(n)}, P) = 0 \tag{3.33}$$

but $f(P_{(n)})$ does not converge to $f(P)$, i.e.,

$$\lim_{n\to\infty} f(P_{(n)}) \neq f(P) \tag{3.34}$$

Discontinuity

In this section, we establish the discontinuity of H at any degree distribution having a countably infinite support. Let (3.35) be a sequence of degree distributions with the real parameters $a > 1$,

$b > 0$ and $n > a$. We will use this sequence to show that H is discontinuous at $P_1 = (1, 0, 0, \ldots)$.

$$P_n^{a,b} = \{1 - (\frac{\log a}{\log n})^b, \frac{1}{n}(\frac{\log a}{\log n})^b \ldots \frac{1}{n}(\frac{\log a}{\log n})^b, 0, 0, \ldots\} \tag{3.35}$$

Based on our definition of convergence, we show that the sequence $P_n^{a,b}$ converges to $P_1 = (1, 0, 0, \ldots)$. Assuming that:

$$D_{KL}(P_1, P_n^{a,b}) = -\left(\log(1 - \left(\frac{\log a}{\log n}\right)^b\right) \tag{3.36}$$

we have $P_n^{a,b} \to P$, which is given in (3.37).

$$\lim_{n\to\infty} D(P_n^{a,b}, P_1) = 0 \tag{3.37}$$

Then, the entropy of $P_n^{a,b}$ is given by (3.38).

$$H(P_n^{a,b}) = -\left[1 - \left(\frac{\log a}{\log n}\right)^b\right] \log\left[1 - \left(\frac{\log a}{\log n}\right)^b\right] - n\left[\frac{1}{n}\left(\frac{\log a}{\log n}\right)^b\right] \log\left[\frac{1}{n}\left(\frac{\log a}{\log n}\right)^b\right] \tag{3.38}$$

In order to compute the limit of (3.38), let

$$H(P_n^{a,b}) = \underbrace{-\left[1 - \left(\frac{\log a}{\log n}\right)^b\right] \log\left[1 - \left(\frac{\log a}{\log n}\right)^b\right]}_{T_1} \underbrace{-n\left[\frac{1}{n}\left(\frac{\log a}{\log n}\right)^b\right] \log\left[\frac{1}{n}\left(\frac{\log a}{\log n}\right)^b\right]}_{T_2} \tag{3.39}$$

We have $\lim_{n\to\infty} T_1 = -1 \times \log(1) = 0$.
T_2 can be written as in (3.40).

$$T_2 = \underbrace{\left(\frac{\log a}{\log n}\right)^b \log(n)}_{T_3} + \underbrace{\left(\frac{\log n}{\log a}\right)^b \log\left(\left(\frac{\log a}{\log n}\right)^b\right)}_{T_4} \tag{3.40}$$

Using *L'Hôpital*'s rule, we can show that $\lim_{n\to\infty} T_4 = 0$, while for T_3, we have (3.41).

$$\lim_{n\to\infty} (\log(a))^b \frac{log(n)}{(log(n))^b} = \lim_{n\to\infty} (\log(a)^b)(log(n))^{1-b} \tag{3.41}$$

Hence,

$$\lim_{n\to\infty} H(P_n^{a,b}) = \begin{cases} 0 & \text{if } b > 1 \\ \log a & \text{if } b = 1 \\ \infty & \text{if } 0 < b < 1 \end{cases} \tag{3.42}$$

From (3.42), we can provide the following proposition.

Proposition 1. *Based on Definition 7., and assuming that $f = H$. If we take $a > 1$ and $0 < b \leq 1$ in (3.42), we have (3.43)*

$$\lim_{n\to\infty} H(P_n^{a,b}) = \infty \neq H(P_1) \tag{3.43}$$
$$\textit{but} \qquad \lim_{n\to\infty} P_n^{a,b} = P_1$$

Thus, we can state that the entropy H is discontinuous at the degree distribution $P_1 = (1, 0, 0, \ldots)$.

3.3.5 Discussion

We start by describing the evolution of the cognitive process P with respect to its parametric degree distribution. Then, we provide an interpretation of the uncertainty of P, assuming that the number of concepts is countably infinite. We propose to use the same general degree sequence (3.35) provided in the previous section.

At the beginning of the cognitive process (3.25), the agent can only grasp a small and limited number of concepts (nodes). Therefore, there is a low probability that a primed concept will have a high degree. This is due partially to the fact that the agent starts the cognitive process (at $t = 0$) with no prior associations between the concepts. As time goes on, the agent will start creating and adding associations between the concepts and the degrees of the nodes will start increasing. Thus, the likelihood of the existence of concepts with a high degree within the frame of the agent will increase. The reason for which we assumed that the degree distribution of the process is described by (3.35) is the possibility to define a dominant degree. In this case, one degree will occur with a probability which is higher than the probability of occurrence of the other degrees.

For instance, the probability of the degree k in the general representation (3.44) will be equal to 1 for $n \to \infty$. In this situation, the associative potential of the agent is k, and reflects the capacity of agent to make at least k associations. Similarly to what we have stated in Definition 4, the associative potential of an agent is the rank of $\left(1 - (\frac{\log a}{\log n})^b\right)$, that is, k.

$$\begin{aligned} P_n^{a,b} &= \{\tfrac{1}{n}(\tfrac{\log a}{\log n})^b, \ldots, \tfrac{1}{n}(\tfrac{\log a}{\log n})^b, \left(1 - (\tfrac{\log a}{\log n})^b\right)_k, \ldots \\ &\quad \ldots, \tfrac{1}{n}(\tfrac{\log a}{\log n})^b, \ldots, \tfrac{1}{n}(\tfrac{\log a}{\log n})^b, 0, 0, \ldots\} \end{aligned} \tag{3.44}$$

We note that the term $\frac{1}{n}(\frac{\log a}{\log n})^b$ is repeated n times in (3.44), and $|P_n^{a,b}| = N \to \infty$. From the asymptotic degree distribution (3.44) we can describe the evolution of the degrees in Table 3.2.

deg	1	2	3	4	5	6	7
$P_3^{2,1}$	.36	.21	.21	.21	0	0	0
$P_4^{2,1}$	.5	.125	.125	.125	.125	0	0
$P_5^{2,1}$	.569	.086	.086	.086	.086	.086	0
			$\vdots$	$n \to \infty$			
$P_{n,1}^{2,1}$	1	0	0	0	0	0	0
			or				
$P_{n,4}^{2,1}$	0	0	0	1	0	0	0

Table 3.2: Dominant ranks for $k = 1$ and $k = 4$

For $a = 2$ and $b = 1$, and for different choices of the associative potentials ($k = 1$ or $k = 4$), the sequence will yield different deterministic distributions. For instance, $k = 1$ generates (3.45) and $k = 3$ yields (3.46).

$$P_{n,1}^{a,b} = \{\left(1 - (\tfrac{\log a}{\log n})^b\right)_1, \tfrac{1}{n}(\tfrac{\log a}{\log n})^b, \ldots, \tfrac{1}{n}(\tfrac{\log a}{\log n})^b, 0, 0, \ldots\} \tag{3.45}$$

$$P_{n,3}^{a,b} = \{\tfrac{1}{n}(\tfrac{\log a}{\log n})^b, \tfrac{1}{n}(\tfrac{\log a}{\log n})^b, \left(1 - (\tfrac{\log a}{\log n})^b\right)_3, \tfrac{1}{n}(\tfrac{\log a}{\log n})^b, \ldots, \tfrac{1}{n}(\tfrac{\log a}{\log n})^b, 0, 0, \ldots\} \tag{3.46}$$

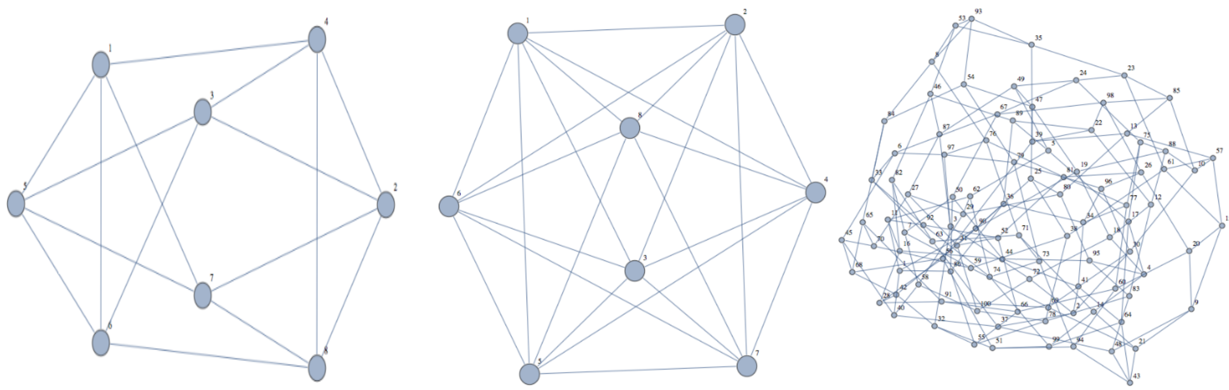


Figure 3.4: Cognitive Graphs for $(4-8)$, $(6-8)$, and $(4-100)$

The sequence (3.45) converges to a deterministic distribution $P_1 = \{1, 0, 0, \ldots\}$ (similarly, (3.46) converges to $P_3 = \{0, 0, 1, 0, 0, \ldots\}$). Consequently, the representative graph $G_{n,1}$ having a degree distribution $P_{n,1}^{a,b}$ will converge to a graph G_1 described by P_1. In case we take $k = 1$ as in (3.45), we get a Bipartite graph represented by P_1. This case is not the most efficient one in the sense that the agent cannot build more then 1 association for a given concept, which means that some relevant concepts won't be connected. As k increases, more associations will be created and added, and the more k tends to the maximal degree, *i.e.*, the number of nodes N. If $k = n+1$, then the graph has $n+1$ nodes with $n+1$ degree each, and the corresponding degree distribution is defined as in (3.47).

$$P_{n,n+1}^{a,b} = \{ \quad \frac{1}{n}(\frac{\log a}{\log n})^b, \frac{1}{n}(\frac{\log a}{\log n})^b, \left(1 - (\frac{\log a}{\log n})^b\right)_{n+1}, 0, 0, \ldots, 0, \ldots\} \tag{3.47}$$

Hence, the limiting graph is a complete graph K_{n+1} of degree $k = n+1$. Having a complete graph of concepts reflects the idea of complete and unbounded associative potential over all the available concepts within the frame. To illustrate how the evolution of the process affects the representation of the concepts, we take the examples in Figure 3.4.

Let d_n be the associative potential of the graph at state n. For instance, two different situations in Figure 3.4, $(4-8)$ and $(6-8)$ reflect that in the first case the associative potential is $d_8 = 4$ while in the second case, it is $d_8 = 6$. Now if we compare $(4-100)$ with $(4-8)$, we find that more nodes are added as time goes on ($d_{100} = 4$), although the associative potential is the same ($d_{100} = d_8 = 4$).

The convergence of the entropy of the process (3.42) over time is defined according to different values of (a, b). For instance, the entropy tends asymptotically to infinity for the values $b = 0.1$ and $a = 2$. This case illustrates the divergence of the entropy proposed in (3.43), and which seems to be different from the entropy of the limiting distribution P_1. It is obvious that the entropy of he limiting distribution is null ($H(P_1) = 0$), which asserts the certainty of the agent with regard to the final phase of the cognitive process. However, given (3.43), we can note that the certainty of the overall process cannot be deduced by considering all the successions of cognitive updates that the agent is performing. Indeed, even though we are certain that the final limiting configuration P_1 is going to be achieved by the agent, we can not deduce this certainty from observing the overall process.

3.3.6 Summary

We provided a cognitive model for concepts representation as well as their underlying associations by adopting an undirected graph structure. The provided topology reflects the complexity of the conceptual graph. Furthermore, the proposed model describes a cognitive process where an agent is updating the concepts and the associations. We evaluated the uncertainty of the overall process with respect to the entropy measure over countably infinite set of concepts. The results include an interpretation of the evolution of process. Thus, we proved that the larger the cognitive space is, and the more uncertain the evolution of the process can be, despite the certainty of the final outcome. This situation recalls the problem of infinite regress as well as the frame problem, in the sense that we incorporated the concept of infinity while reasoning about cognition.

It is possible to consider the directions of the concepts activation by adopting a directed cognitive map. This case demands a separation between pertinent concepts and referential concepts. Another issue to be addressed is to fully characterize the graphical structures that are likely to trigger heuristics in a bounded rational setting, known generally as “Fast and Frugal” heuristics

[43]. Being able to activate the most pertinent concepts and discard the irrelevant concepts could improve any search strategy.

Chapter 4

Utility Hyper-graphs

4.1 Introduction

Automated multi-agent systems could help reaching agreements among heterogenous and distributed decision makers. In fact, their applications range from coordination and cooperation [69, 55] to task allocation [67, 60], surplus division [29], and decentralized information services [68]. In practical, most of the realistic decision making problems are characterized by interdependent issues, which yields complex and nonlinear utility spaces [53]. As the search space and the complexity of the problem grow, finding optimal contracts becomes intractable for one single agent. Similarly, reaching an agreement between a group of agents becomes harder.

We propose to tackle the complexity of the utility spaces used in multi-issue decision making by rethinking the way they are represented. We claim that adopting the adequate representation gives a solid ground to tackle the scaling problem. We address this problem by adopting a representation that allows a modular decomposition of the issues-constraints given the idea that constraint-based utility spaces are nonlinear with respect to issues, but linear with respect to the constraints. This allows us to map the utility space into an issue-constraint hyper-graph with the underlying interdependencies. Exploring the utility space reduces then to a message passing mechanism along the hyper-edges by means of utility propagation.

Adopting a graphical representation while reasoning about utilities is not new in both the multi-attribute utility and the multi-issue negotiation literatures. Indeed, the idea of utility graphs could potentially help decomposing highly nonlinear utility functions into sub-utilities of clusters of inter-related items, as in [14, 6]. Similarly, [88] used utility graphs for preferences elicitation and negotiation over binary-valued issues. [78] adopts a weighted undirected graph representation of the constraint-based utility space. However, restricting the graph and the message passing process to constraints' nodes does not allow the representation to be descriptive enough to exploit any potential hierarchical structure of the utility space through a quantitative evaluation of the interdependencies between both issues and constraints. In [39], issues' interdependency are captured by means of similar undirected weighted graphs where a node represents an issue. This representation is restricted to binary interdependencies while real negotiation scenarios involve "bundles" of interdependent issues under one or more specific constraints. In our approach, we do not restrict the interdependency to lower-order constraints but we allow $p-$ary interdependencies to be defined as an hyper-edge connecting p issues. The advantage of our representation is its scalability in the sense that the problem becomes harder for a large number of issues and constraints. But if we can decompose the utility space into independent components, we can exploit it more efficiently using a message passing mechanism.

Another motivation behind the hyper-graph representation is that it allows a layered, hierarchical view of any given negotiation problem. Given such architecture, it is possible to recursively negotiate over the different layers of the problem according to a top-down approach. Even the idea of issue could be abstracted to include an encapsulation of sub-issues, located in sub-utility spaces and represented by cliques in the hyper-graph. Consequently, search processes can help identify optimal contracts for improvement at each level. It is within this perspective that we are proposing our model. The main novelty our work is the efficiency of the new representation when optimizing nonlinear utilities. To the best of our knowledge, our work makes the first attempt to tackle the complexity of such utility spaces using an efficient search heuristic that works and outperforms the previously used sampling-based meta-heuristics. Particularly, the novelty is that we exploit the problem structure (as hyper-graph) as well as randomization. Such performance is required when facing the scaling issues, inherent to complex negotiation. We experimentally evaluated our model using parametrized and random nonlinear utility spaces, showing that it can handle large and complex spaces by finding the optimal contracts while outperforming previous sampling approaches.

4.2 Nonlinear Utility Spaces

4.2.1 Formulation

We start from the formulation of nonlinear multi-issue negotiation of [52]. That is, N agents are negotiating over n issues $i_{k\in[1,n]} \in \mathbb{I}$, with $\mathbb{I} = \{i_k\}_{k=1}^{n}$, forming an $n-$dimensional utility space. The issue k, namely i_k, takes its values from a set $\mathbb{I}_k$ where $\mathbb{I}_k \subset \mathbb{Z}$. A contract $\vec{c}$ is a vector of issue values $\vec{c} \in \mathcal{I}$ with $\mathcal{I} = \times_{k=1}^{n} \mathbb{I}_k$.

An agent's utility function is defined in terms of constraints, making the utility space a constraint-based utility space. That is, a constraint $c_{j\in[1,m]}$ is a region of the total $n-$dimensional utility space. We say that the constraint c_j has value $w(c_j, \vec{c})$ for contract $\vec{c}$ if constraint c_j is satisfied by contract $\vec{c}$. That is, when the contract point $\vec{c}$ falls within the hyper-volume defined by the constraint of c_j, namely $hyp(c_j)$. The utility of an agent for a contract $\vec{c}$ is thus defined as in (4.1).

$$u(\vec{c}) = \sum_{c_{j\in[1,m]},\ \vec{c}\in hyp(c_j)} w(c_j, \vec{c}) \tag{4.1}$$

In the following, we distinguish three types of constraints: Cubic constraints, Bell constraints and Plane constraints, shown in Figure 4.1. The constraint-based utility formalism is a practical way to reason about preferences subject to restrictions. More details about constraint-based utility spaces and their usage is to be found in [78, 77, 79].

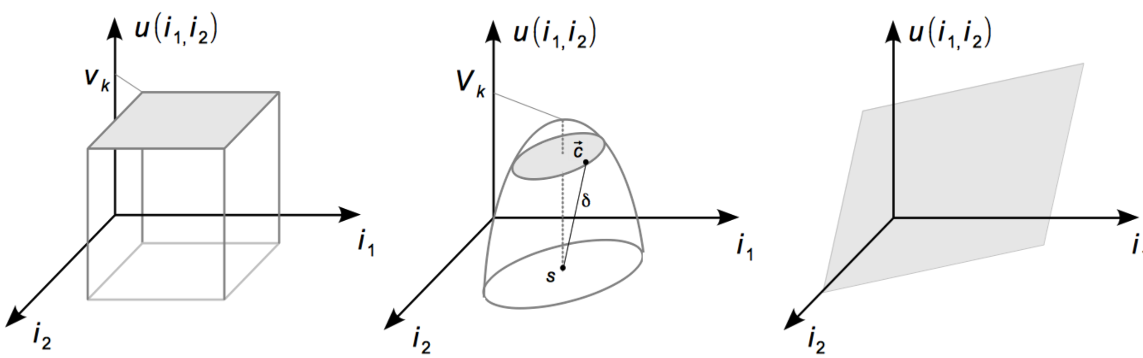


Figure 4.1: Cubic, Bell and Plane Constraints

Having a large number of constraints produces a “bumpy" nonlinear utility space with high points whenever many constraints are satisfied and lower points where few or no constraints are satisfied. Figure 4.2 shows an example of nonlinear utility space for issues i_1 and i_2 taking values in $\mathbb{I}_1 = \mathbb{I}_2 = [0, 100]$, with $m = 500$ constraints and where a constraint involves at most 2 issues.

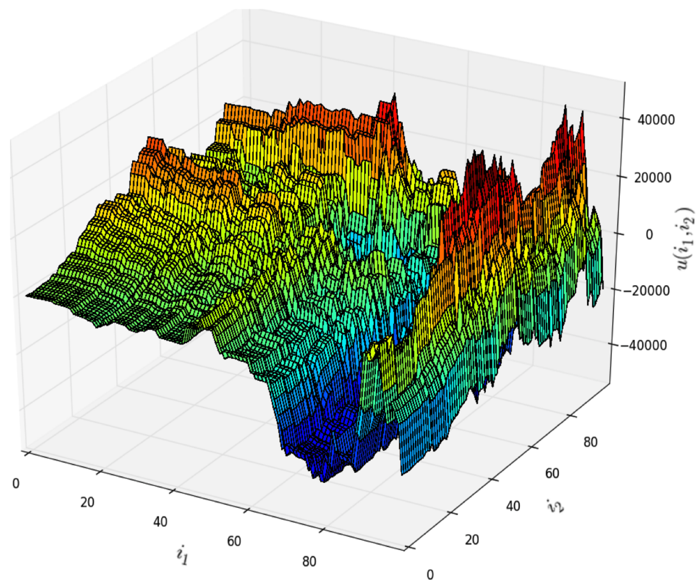


Figure 4.2: $2-$dimensional nonlinear utility space

4.2.2 New Representation

The agent’s utility function (4.1) is nonlinear in the sense that the utility does not have a linear expression against the contract [52]. This is true to the extent that the linearity is evaluated with regard to the contract $\vec{c}$. However, from the same expression (4.1) we can say that the utility is in fact linear, but in terms of the constraints $c_{j\in[1,m]}$. The utility space is therefore decomposable according to the c_j constraints. This yields a modular representation of the interactions between the issues and how they locally relate to each other. In fact, $hyp(c_j)$ reflects the idea that the underlying contracts are governed by the bounds defined by c_j once the contracts are projected according to their issues’ components.

In this case, the interdependence is not between issues but between constraints. For instance, two constraints c_1 and c_2 can have in common one issue i_k taking values respectively from an interval $\mathbb{I}_{k,c_1}$ if it is in c_1, and values in $\mathbb{I}_{k,c_2}$ if it is in c_2, with $\mathbb{I}_{k,c_1} \neq \mathbb{I}_{k,c_2}$. Finding the value that maximizes the utility of i_k while satisfying both constraints becomes harder due to fact that changing the value of i_k in c_1 changes its instance in c_2 in a cyclic manner. This gets worse with an increasing number of issues, their domains' sizes, and the non-monotonicity of the constraints.

Next, we propose to transform (4.1) into a modular, graphical representation. Since one constraint can involve one or more multiple issues, we adopt a hyper-graph representation.

4.2.3 From Utility Space To Utility Hyper-graph

We assign to each constraint $c_{j\in[1,m]}$, a factor Φ_j, with $\Phi = \{\Phi_j\}_{j=1}^{m}$. We define the hyper-graph G as $G = (\mathbb{I}, \Phi)$. Nodes in $\mathbb{I}$ define the issues and the hyper-edges in Φ are the factors (constraints). To each factor Φ_j we assign a neighbors' set $\mathcal{N}(\Phi_j) \subset \mathbb{I}$ containing the issues connected to Φ_j (involved in c_j), with $|\mathcal{N}(\Phi_j)| = \varphi_j$. In case $\varphi_j = 2\ \forall j \in [1, m]$, the problem collapses to a constraints satisfaction problem in a standard graph.

To each factor Φ_j corresponds a φ_j-dimensional matrix, $\mathcal{M}_{\Phi_j}$, where the jth dimension is the discrete interval $[a_k, b_k] = \mathbb{I}_k$, the domain of issue i_k. This matrix contains all the values that could be taken by the issues in $\mathcal{N}(\Phi_j)$. Each factor Φ_j has a function Φ_j defined as a sub-utility function of the issues in $\mathcal{N}(\Phi_j)$, as in (4.2).

$$
\begin{aligned}
\Phi_j :\ & \mathcal{N}(\Phi_j)^{\varphi_j} \to \mathbb{R} \\
& \Phi_j(i_1, \ldots, i_k, \ldots, i_{\varphi_j}) \mapsto w(c_j, \vec{c})
\end{aligned}
\tag{4.2}
$$

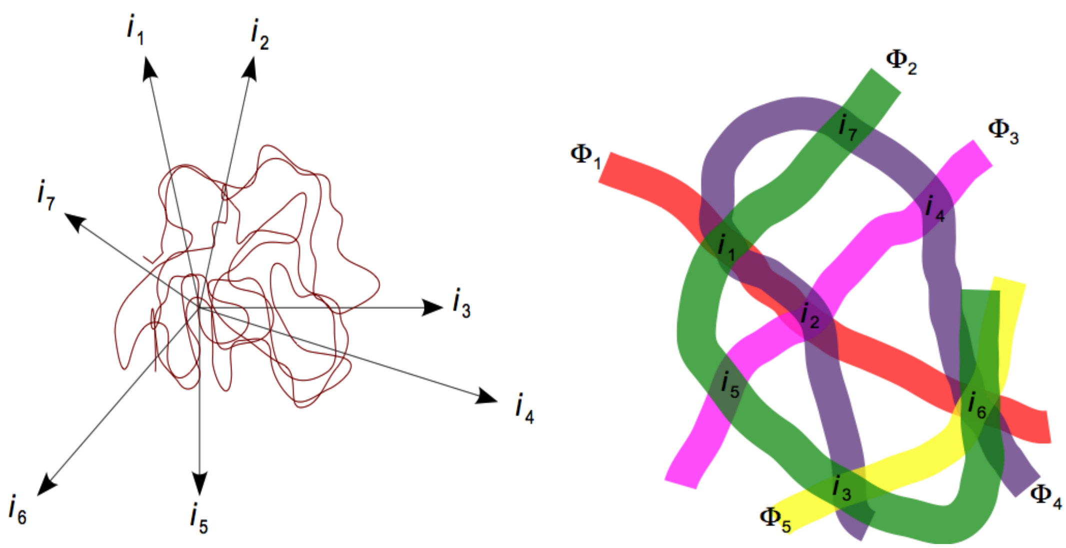
As we are dealing with discrete issues, Φ_j is defined by the matrix $\mathcal{M}_{\Phi_j}$. That is, $\Phi_j(i_1, \ldots i_k, \ldots, i_{\varphi_j})$ is simply the $(1, ..., k, ..., \varphi_j)^{th}$ entry in $\mathcal{M}_{\Phi_j}$ corresponding as well to the value $w(c_j, \vec{c})$ mentioned in (4.1). It is possible to extend the discrete case to the continuous one by allowing continuous issue-values and defining Φ_j as a continuous function.

The mapping from the cartesian representation (utility space) to the hyper-graphical representation could for instance be shown as in Figure 4.3. The utility space in Figure 4.3(a) shows no clear structure. However, its hyper-graphical counterpart in Figure 4.3(b) represents the total utility in a structured manner, as an aggregation of factors (4.3) or hyper-edges. The resulting utility formulation is therefore expressed as in (4.3).

$$u(\vec{c}) = \phi_1(i_1, i_2, i_6) + \phi_2(i_1, i_3, i_5, i_6, i_7) + \phi_3(i_2, i_4, i_5) + \phi_4(i_1, i_2, i_3, i_4, i_6, i_7) + \phi_5(i_3, i_6) \tag{4.3}$$

As shown in Figure 4.3(b), it is possible to see the interdependencies,

- as existing between the constraints through the intersecting issues. For instance, constraints Φ_2 and Φ_5 are interdependent given issues i_3 and i_6.
- as existing between the different instantiations that one issue can have, depending on where it is defined, as explained in section 4.2.2. For instance, the issue i_6 could have different domains, depending on the constraints it is defined in (Φ_1, Φ_2, Φ_4, and Φ_5).



(a) Utility Space $u(i_1, \ldots, i_7)$ (b) Utility Hyper-graph $G(I, \Phi)$

Figure 4.3: Mapping from u to G

The comparison between the two views of the problem is shown in Table 4.1.

Cartesian view of the problem	Graphical view of the problem
Constraints and issues are entangled	Constraints and issues are differentiated
Analytical	Representational, Graphical, Connectionist
No apparent structure	Well structured, topology can be used
Not intuitive, due to high dimensionally	Intuitive, representative of real-world problems

Table 4.1: Cartesian vs. Graphical Representations

Example

To provide an example of our representation, let us consider a $10-$dimensional utility space and propose to represent it in a graphical way. That is, as a hyper-graph defined as $G_{10} = (\mathbb{I}, \Phi)$ with $\mathbb{I} = \{i_k\}_{k=1}^{9}$, $\Phi = \{\Phi_j\}_{j=1}^{7}$ and shown in Figure 4.4.

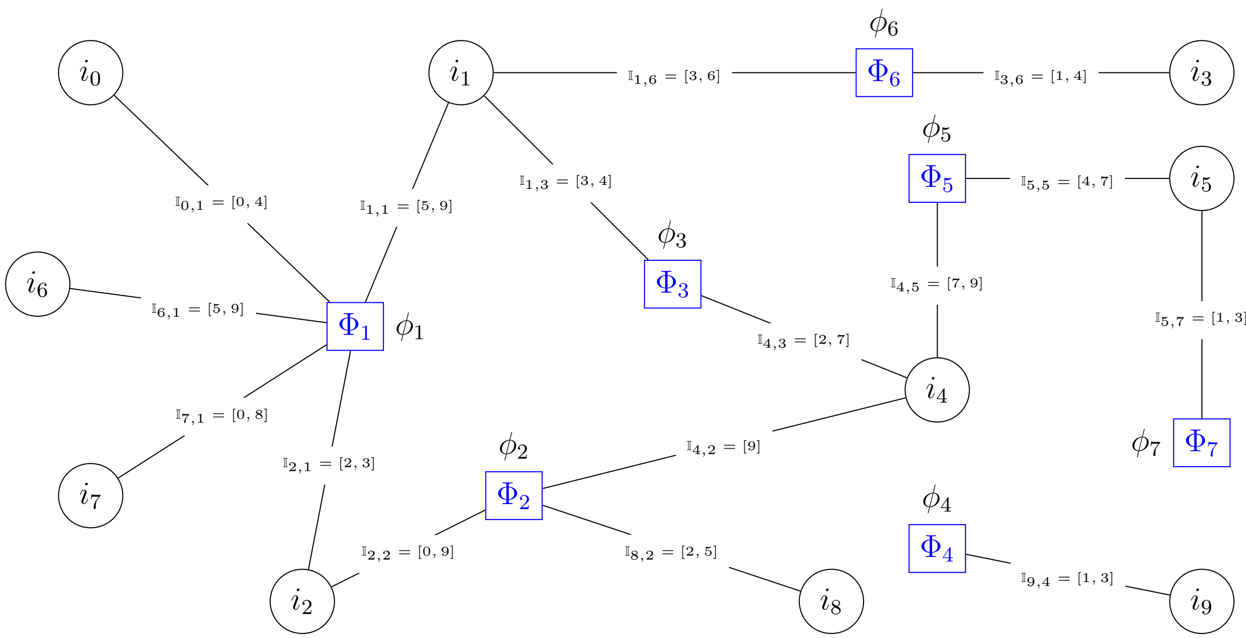


Figure 4.4: Issues-Constraints Hyper-graph

Each issue i_k has a set $\mathbb{I}_k = \bigcup_{\nu \in \mathcal{N}(k)} \mathbb{I}_{k,\nu}$ where $\mathbb{I}_{k,\nu}$ is an edge connecting i_k to its neighbor $\nu \subset \mathcal{N}(k) \in \Phi$. For example, $\mathbb{I}_1 = \bigcup_{\nu \in \{\Phi_1, \Phi_3, \Phi_6\}} \mathbb{I}_{1,\nu} = \{[5,9], [3,4], [3,6]\}$.

Constraints can have different types in the sense that each type reflects a particular geometric shape. For example, constraints $\Phi_{1,2,3,4}$ could be cubic, $\Phi_{5,6}$ could be defined as planes and Φ_7 defined as a bell. Any combination is in fact possible, and depends only on the problem in hand and how it is being specified. Each constraint is assigned a sub-utility representation used to compute the utility of a contract if it satisfies the corresponding constraint by being located in the underlying hyper-volume. For example, the general utility function Φ_j, defined in (4.2), could correspond to the functional definition of each constraints, as shown in (4.4).

$$\Phi_j = \begin{cases} Plane: & \beta_j + \sum_{k=1}^{\varphi_j} \alpha_{j,k} \times v_k(i_k) & (\beta_j, \alpha_{j,k}) \in \mathbb{Z}^2 \\ Cube: & v_j & \\ Bell: & V_j & \end{cases} \tag{4.4}$$

A plane constraint will be defined using its φ_j-dimensional equation, while a cubic constraint will be assigned the value v_j in case the contract is in the cube. The computation of the utility V_j of a bell shaped constraint is performed as in (4.5) (see Figure 4.1). Herein, δ is the Euclidean distance from the center s of the bell constraint to a contract point $\vec{c}$. Distances are normalized in $[-1, 1]$.

$$V_j = \begin{cases} \beta_j\,(1 - 2\delta^2) & \text{if} \quad \delta < 0.5 \quad \beta_j \in \mathbb{Z} \\ 2\,\beta_j\,(1-\delta)^2 & \text{if} \quad \delta < 1 \quad\;\; \beta_j \in \mathbb{Z} \\ 0 & \text{else} \end{cases} \tag{4.5}$$

It is possible to extend the current constraints' to involve Cone constraints [37] or any other type of geometrical shapes.

4.3 Optimal Contracts

The exploration of the utility hyper-graph is inspired from the sum-product message passing algorithm for belief propagation [87]. However, the multiplicative algebra is changed into an additive algebra to support the utility accumulation necessary for the assessment of the contracts. The messages circulating in the hyper-graph are nothing other than the contracts we are attempting to optimize through utility maximization. Next, we develop the message passing (MP) mechanism operating on the issues and the constraints.

4.3.1 Message Passing

We consider the issues set $\mathbb{I}$ and a contract point $\vec{c} = (i_1, \ldots, i_k, \ldots, i_n) \in \mathcal{I}$. We want to find a contract $\vec{c^*}$ that maximizes the utility function defined in (4.1). Assuming that Φ_j is the local sub-utility of constraint Φ_j, we distinguish two types of messages: messages sent from issues to constraints, and messages sent from constraints to issues.

From issue i_k to constraint Φ_j

In (4.6), each message $\mu_{i_k \to \Phi_j}$ coming from i_k to Φ_j is the sum of the constraints' messages to i_k coming from constraints other than Φ_j.

$$\mu_{i_k \to \Phi_j}(i_k) = \sum_{\Phi_{j'} \in \mathcal{N}(i_k) \setminus \Phi_j} \mu_{\Phi_{j'} \to i_k}(i_k) \tag{4.6}$$

From constraint Φ_j to issue i_k

Each constraint message (4.7) is the sum of the messages coming from issues other than i_k, plus the constraint value $\Phi_j(i_1, \ldots, i_k, \ldots, i_n)$, summed over all the possible values of the issues (connected to the constraint Φ) other than the issue i_k.

$$\mu_{\Phi_j \to i_k}(i_k) = \max_{i_1} \ldots \max_{i_{k'} \neq k} \ldots \max_{i_n} \left[\Phi_j(i_1, \ldots, i_k, \ldots, i_n) + \sum_{i_{k'} \in \mathcal{N}(\Phi_j) \setminus i_k} \mu_{i_{k'} \to \Phi_j}(i_k) \right] \tag{4.7}$$

The MP mechanism starts from the leaves of the hyper-graph, *i.e.*, the issues. At $t = 0$, the content of the initial messages is defined according to (4.8), with $\phi'_j(i_k)$ being the partial evaluation of i_k in the factor Φ_j.

$$\mu_{i_k \to \Phi_j}(i_k) = 0 \tag{4.8}$$

$$\mu_{\Phi_j \to i_k}(i_k) = \phi'_j(i_k) \tag{4.9}$$

The partial evaluation $\phi'_j(i_k)$ of issue i_k in the factor Φ_j is the utility of i_k using Φ_j regardless of any other issue involved in Φ_j. For instance, for cubic and bell constraints, the evaluation is simply $v_j(i_k)$ and $V_j(i_k)$ $\forall k$ as described in (4.4). If Φ_j is a plane constraint, the partial evaluation of i_k will be $\alpha_{j,k} \times v_k(i_k)$. In this manner, the factor Φ_j will get all the evaluations ($\alpha_{j,k} \times v_k(i_k)$) from its surrounding issues in order to yield the total utility (4.4) as a sum of the partial evaluations plus the plane constant β_j.

Finally, the optimal contract $\vec{c^*}$ is found by collecting the optimal issues as in (4.10).

$$\vec{c^*} = \Bigg(\arg\max_{i_1} \sum_{\Phi_j \in \mathcal{N}(i_1)} \mu_{\Phi_j \to i_1}(i_1), \; \dots \\ \dots, \; \arg\max_{i_k} \sum_{\Phi_j \in \mathcal{N}(i_k)} \mu_{\Phi_j \to i_k}(i_k), \; \dots \\ \dots, \; \arg\max_{i_n} \sum_{\Phi_j \in \mathcal{N}(i_n)} \mu_{\Phi_j \to i_n}(i_n) \Bigg) \tag{4.10}$$

In a negotiation setting, it is more common that the agent requires a collection, or *bundle*, of the optimal contracts rather than one single optimum. In order to find such collection, we should endow (4.10) with a caching mechanism allowing each node in the hyper-graph to store the messages that have been sent to it from the other nodes. That is, the cached messages will contain the summed-up utility values of the underlying node's instance. This is performed every time the operation *max* is called in (4.7) so that we can store the settings of the adjacent utility (and contract) that led to the maximum. Once ordered, such data structure allows us to generate an ordered bundle for the bidding process. In the next section, we algorithmically provide the MP mechanism.

4.3.2 Utility propagation algorithm

Main algorithm

Algorithm 2 operates on the hyper-graph nodes by triggering the MP process. Despite the fact that we have two types of nodes (issues and constraints), it is possible to treat them abstractly using $MsgPass$. The resulting bundle is a collection of optimal contracts with utility greater or equal to the agent's reservation value rv.

Algorithm: Utility Propagation

Input: $G = (\mathbb{I}, \Phi), rv, mode, \rho$

Output: Optimal contracts (bundle)

begin
    for $i = 1 \rightarrow (\rho \times |\mathbb{I} \cup \Phi|)$ **do**
        if $mode\ is\ Synchronous$ **then**
            foreach $\nu_{src} \in \mathbb{I} \cup \Phi$ **do**
                foreach $\nu_{dest} \in \nu_{src}.Neighbors()$ **do**
                    $\nu_{src}.MsgPass(\nu_{dest})$
    $bundle \leftarrow \emptyset$
    foreach $i \in \mathbb{I}$ **do**
        $bundle[i] \leftarrow \emptyset$
        $\iota \leftarrow \cup_{j \in i.instances()}[j.min, j.max]$
        $\mu^* \leftarrow k^* \leftarrow -\infty$
        $\mu \leftarrow i.getmax()$
        foreach $k = 1 \rightarrow |\mu|$ **do**
            if $\mu^* < \mu[k]$ **then**
                $\mu^* \leftarrow \mu[k]$
                $k^* \leftarrow k$
                if $\mu^* \geq rv$ **then**
                    $bundle[i] \leftarrow bundle[i] \cup \iota[k^*]$
    return $bundle$

Algorithm 2: Main Algorithm

Issue to Constraint

The issue's message to a factor (or constraint) is the element-wise sum of all the incoming messages from other factors, as shown in Algorithm 3.

Constraint to Issue

In Algorithm 4, the factor's message to a targeted issue is done by recursively enumerating over all variables that the factor references (4.7), except the targeted issue. This needs to be performed for each value of the target variable in order to compute the message. If all issues are assigned ($\nexists i : \alpha[i] = -1$), the values of the factor and of all other incoming messages are determined,

Algorithm: MsgPass

Input: $G(\mathbb{I}, \Phi),\ \Phi_j$

Output: Updated message μ

begin
- $\mu \leftarrow [0] \times |\mathbb{I}_k|$
- **for** $\nu \in \mathcal{N}(i_k) \backslash \Phi_j$ **do**
  - $\mu \leftarrow \mu + \nu.GetMsg()$
- **return** μ

Algorithm 3: MsgPass: Issue to constraint

so that their sum term is compared to the prior maximum, as in Algorithm 5. The resulting messages, stored in $bundle$, contain the values that maximize the factors' local utility functions.

Algorithm: MsgPass

Input: $G(\mathbb{I}, \Phi),\ i_k$

Output: Updated message μ

begin
- $\alpha \leftarrow [-1] \times \varphi_j$
- $\iota \leftarrow \pi_{i_k}(\Phi_j)$
- **if** $\iota = \emptyset$ **then**
  - $\mu \leftarrow i_k.GetMsg()$
  - **for** $i = 1 \rightarrow len(\mu)$ **do**
    - $\alpha[\iota] \leftarrow i$
    - $\mu[i] \leftarrow Sum(\alpha, i_k)$
  - **return** μ

Algorithm 4: MsgPass: Constraint to Issue

Optimal issue-values

At any time of the utility propagation process, it is possible to collect the current optimal contract(s) by individual concatenation of all the optimal issue-values i_k^*, defined in (4.10). Particularly, the summation in (4.10) is performed as to only include the overlapping evaluations depending on how the issue domains are defined for different factors. For instance, Figure 4.5 shows how issue i_k has three possible evaluations depending on $\mathbb{I}_{k,1}$, $\mathbb{I}_{k,j}$ and $\mathbb{I}_{k,m}$.

The maximization objective (4.10) will attempt to find the combination(s) of $v_{1,i}$, $v_{k,i}$ and $v_{m,i}$

```
Algorithm: Sum
Input: G(𝕀, Φ), α, i_k
Output: max
begin
    ι ← {i | i ∈ [1, φ_j] ∧ α[i] = −1}
    if ι = ∅ then
        ρ ← M_{Φ_j}[α]
        for ν ∈ N(Φ_j)\i_k do
            μ ← ν.GetMsg()
            ρ ← ρ + μ[α[i]]
        return ρ
    else
        max ← −∞
        for i = 1 → dim(M_{Φ_j}(ι)) do
            α[ι] ← i
            σ ← Sum(α, i_k)
            if σ > max then
                max ← σ
        return max
```

Algorithm 5: Sum: recursive summing

that maximize the sum. An optimal combination is an optimal issue-value i_k^*.

Mechanism

The full message passing mechanism could be described as following:

- Constructing the utility hyper-graph using the available issues and constraints. Issue domains will have to be specified, and constraints' functions will have to be defined for each constraint and the issues involved in it.
- Injecting the initial values for the issues (4.8) and the constraints (4.9).
- Depending on the propagation strategy being used (synchronous or asynchronous), the algorithms picks a node (deterministically or randomly) from the issue and constraint nodes. This is the propagation source for the current round.

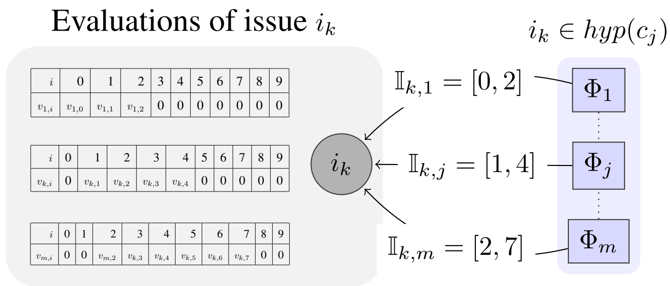


Figure 4.5: Finding the optimal issue-values

- For each chosen node (issue or constraint), the algorithm will propagate the messages to a neighboring node according to the exploration strategy being used.

- The propagation process continuous while computing the utilities for the evaluations and storing the issue-evaluations that maximize the utility. Optimal contracts will be stored in a bundle whenever their utility is greater than a specific reservation value.

- The process will eventually converge when we reach the required optimal bundle size.

4.4 Experiments

4.4.1 Settings

Before evaluating the utility propagation algorithm, we identify the criteria that could affect the complexity of the utility space and thus the probability of finding optimal contract(s). Other than n and m, we distinguish p, defined as the maximal number of issues involved in a constraint. p can be unary ($p = 1$), binary ($p = 2$), ternary ($p = 3$), or p-ary in the general case.

The parametrized generation of a utility space (or utility hyper-graph) should meet the consistency condition $p \leq n \leq m \times p$, with $n, m, p \in \mathbb{N}^+$, to avoid problems like attempting to have an $8-$ary constraints in a $5-$dimensional utility space.

4.4.2 Discussion

After the generation of the hyper-graph using Algorithm 6, the message passing routines will be evaluated and analyzed microscopically from the agent perspective.

Algorithm: ParamRandHGen

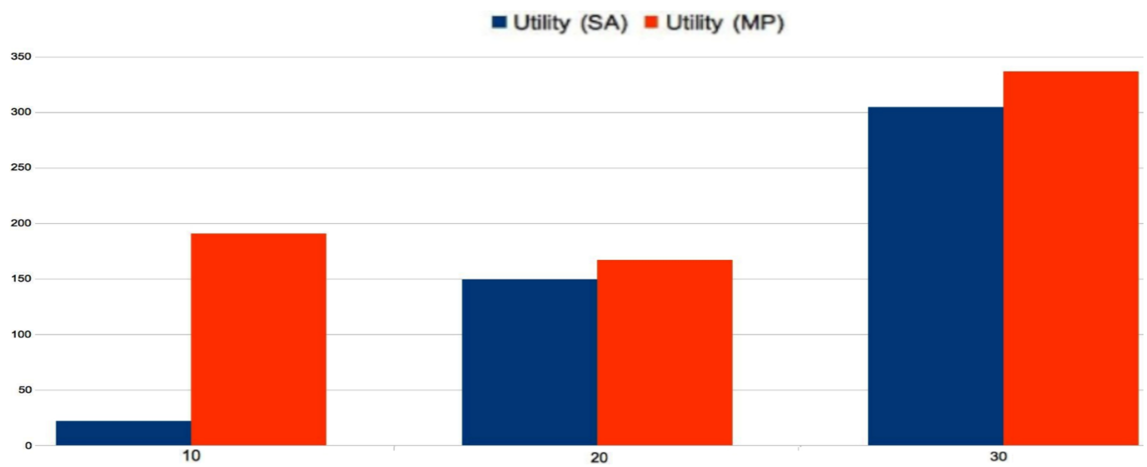
Input: n, m, p

Output: $G(\mathbb{I}, \Phi)$

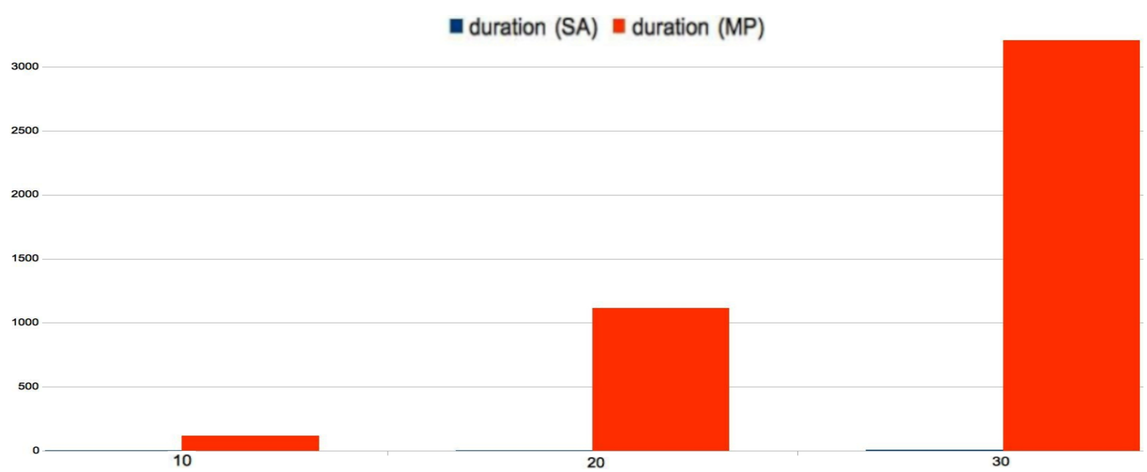
begin
- $[\beta_{min}, \beta_{max}] \leftarrow [1, 100]$ // constants
- $[\alpha_{min}, \alpha_{max}] \leftarrow [0, 1]$ // slopes
- $[b_{min}, b_{max}] \leftarrow [0, 9]$ // bounds
- $\Phi \leftarrow [\emptyset] \times m$ // init constraints set
- **for** $k = 1 \rightarrow m$ **do**
  - $\Phi[k].\theta \leftarrow rand(\{cube, plane, bell\})$
  - **if** $\Phi[k].\theta = plane$ **then**
    - $\alpha \leftarrow [0] \times n$
    - $\alpha[j] \leftarrow rand([\alpha_{min}, \alpha_{max}])\ \forall i \in [1, n]$
    - $\Phi[k].\alpha \leftarrow \alpha$
  - **if** $\Phi[k].\theta \in \{bell, cube\}$ **then**
    - // refer to (4.4) or (4.5)
  - $\Phi[k].\beta \leftarrow rand([\beta_{min}, \beta_{max}])$
  - $\mu \leftarrow rand([1, n])$, $\mathbb{I} \leftarrow \emptyset$
  - **while** $|\mathbb{I}| \neq \mu$ **do**
    - $\iota \leftarrow rand([1, p])$
    - **if** $\iota \notin \mathbb{I}$ **then**
      - $\mathbb{I} \leftarrow \mathbb{I} \cup \iota$
  - **for** $j = 1 \rightarrow \mu$ **do**
    - $\mathbb{I}[j].a \leftarrow rand([b_{min}, b_{max}])$ $\mathbb{I}[j].b \leftarrow rand([\mathbb{I}[j].a + \epsilon, b_{max}])$
  - $\Phi[k].\mathbb{I} \leftarrow \mathbb{I}$
- **return** Φ

Algorithm 6: Utility Hyper-graph Generation

We will compare the MP mechanism in terms of utility and duration to the Simulated Annealing (SA) approach in [52] for optimal contract finding. The SA optimizer will be randomly sampling from the regions that correspond to an overlap of constraints. For instance, generating a random contract satisfying c_j is performed backwardly through random generation of values from $\mathbb{I}_{j,k}$ $\forall i_k \in \mathcal{N}(\Phi_j)$. Our comparison criteria is based on the utility/duration performed on a set of profiles of the form (n, m, p), with 100 trials for each profile. Figure 4.6 illustrates the per-



(a) Utility



(b) Durations Δ_{SA} and $\Delta_{SynchMP}$ for $m \in \{10, 20, 30\}$

Figure 4.6: SynchMP vs. SA for the profile $(10, [10, 20, 30], 5)$

formance of SynchMP for $(10, [10, 20, 30], 5)$. That is, we take the case of utility spaces involving 10 issues, with constraints having cardinalities in $\{10, 20, 30\}$. Each constraint involves at most 5 issues. From Figure 4.6(a), we can see that the synchronous message passing (SynchMP) always finds the high-utility contracts, as opposed to SA. However, from Figure 4.6(b), we can see that SynchMP takes a lot of time when searching for the optimal contracts.

The deterministic aspect of the synchronous message passing algorithm (SynchMP) makes it very slow ($\Delta_{SA} << \Delta_{SynchMP}$) compared to its SA counterpart which exploits the randomization, allowing it to perform “jumps" in the search space. To avoid the enumeration over the local nodes of G, it is possible to add randomization to the way nodes are selected. To introduce an asynchronous mode, AsynchMP, we add another condition after the synchronous mode condition in Algorithm 2, as follows:

if *mode* is *Asynchronous* **then**

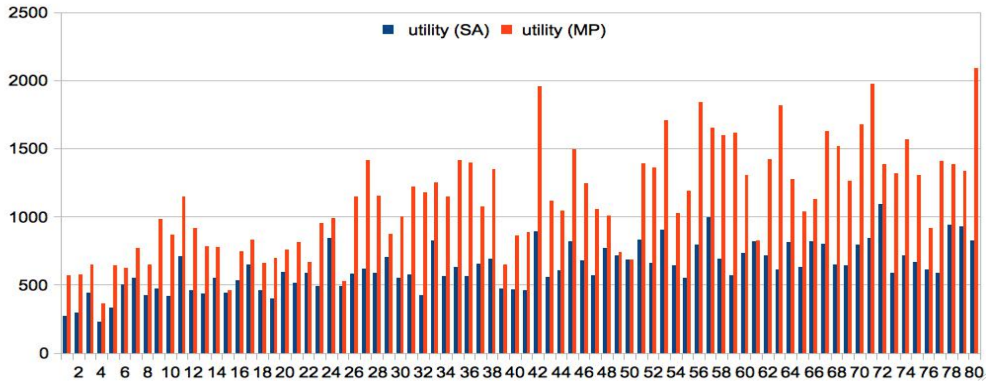
$\nu_{src},\ \nu_{dest} \leftarrow rand_2([1, |V|]),\ \nu_{dest} \neq \nu_{src}$

$\nu_{src}.MsgPass(\nu_{dest})$

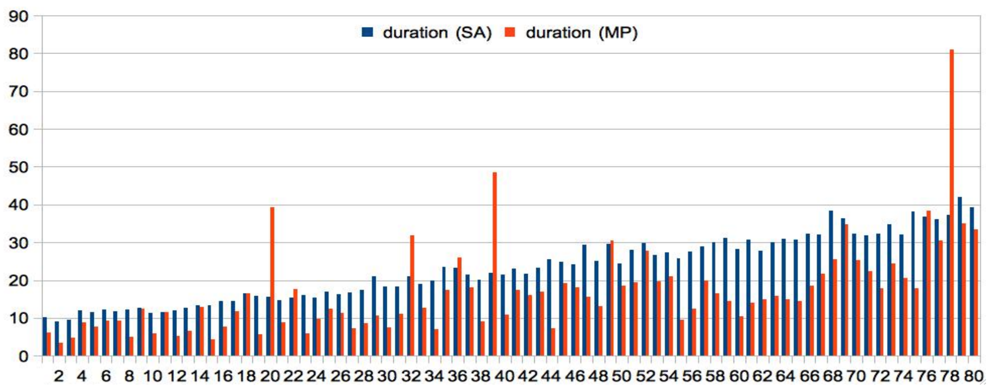
For $(40, [20, \ldots, 100], 5)$, Figure 4.7 shows the resulting difference in the performance of AsynchMP compared to SA. As shown in Figure 4.7(a), the message passing is still better than the sampling-based search for optimal contracts. In terms of duration, Figure 4.7(b) shows that the new asynchronous mechanism (AsynchMP) is on average faster than the SA mechanism, which is due to the usage of randomization when choosing the nodes. In fact, the randomization allows the algorithm to converge faster by covering the utility space regions in a short period of time. Another advantage is that the message passage mechanism has a self-organization property that cannot be found in the sampling-based approaches. This property allows the algorithm to treat different components of the hyper-graph is an independent manner according to the proximity relationships between the constraint and issue nodes. This self-organizing aspect could be improved by running several parallel instances of the algorithm, in the sense that each instance will treat each component of the hyper-graph as an independent problem.

4.5 Conclusion

A new and modular representation for nonlinear utility spaces is proposed. The exploration and search for optimal contracts is performed based on a message passing mechanism in the hyper-graph. Results show that the proposed mechanism outperforms the sampling-based optimizers.



(a) Utility



(b) Durations Δ_{SA} and $\Delta_{ASynchMP}$

Figure 4.7: AsynchMP vs. SA for the profile $(40, [20, \dots 100], 5)$

As future work, we intend to exploit the structure of the hyper-graphs for hierarchical negotiation. Additionally, we think about studying the interdependence and correlation of the issues based on the structure of the utility hyper-graph.

Chapter 5

Low-complexity Search in Utility Hyper-graphs

5.1 Introduction

The propagation, or circulation of the messages in a utility hyper-graph could be defined according to a particular strategy with respect to the hyper-graph topology. For example, the simplest way to propagate the messages could be a systematic, deterministic, synchronous way of choosing the nodes. Another mode could correspond to a randomized, non-deterministic, asynchronous way of selecting the sources and the destinations. In the next sections, we will adopt the asynchronous mode of messages transmission.

5.2 Parametric Random Utility Hyper-graph

Herein, we redefine the utility hyper-graphs using a set of parameters that could potentially affect the problem in hand. This is our proposed analytical way for the study of complex domains. The parameters could carefully be chosen as to generate random utility hyper-graphs with a particular configuration.

It is important to identify the criteria that could affect the complexity of the utility space and thus the probability of finding optimal contract(s). To this end, we start by defining the parameters that could have an impact on the complexity of the preference spaces. These parameters are also used for the generation of the scenarios.

- n: number of issues, attributes, or concepts.

- m: number of constraints, hyper-edges, or factors.

- π: a constraint-distribution. For a constraint c_j and its factor Φ_j, $\pi(\Phi_j)$ refers to the number of issues involved in Φ_j, making it unary ($\pi(\Phi_j) = 1$), binary ($\pi(\Phi_j) = 2$), ternary ($\pi(\Phi_j) = 3$), or $\pi(\Phi_j)$-ary in the general case. π is defined as in (5.1),

$$\begin{aligned} \pi \ &: \Phi \to [1, n] \\ \Phi_j &\mapsto \underbrace{\mathcal{N}(\Phi_j)}_{\pi_j},\ j \in [1, m] \end{aligned} \tag{5.1}$$

  with the resulting sequence (5.2).

$$\pi = (\pi_1, \ldots, \pi_j, \ldots, \pi_m) \tag{5.2}$$

- Constraints' weights distribution, or the way the total utility is distributed amongst the constraints. This could be done by defining a function ψ that assigns a weight to each constraint, as in (5.3).

$$\begin{aligned} \psi \ &: \Phi \to \mathbb{R} \\ \Phi_j &\mapsto w_j \end{aligned} \tag{5.3}$$

- Constraints monotonicity, usually governed by the underlying utility functions: bell-shaped, cubic, planar or conic.

- Domain sizes of the issues $\prod_{k=1}^{n} |\mathbb{I}_k|$ as well as the domains' types (discrete and/or continuous).

The most important criteria is the distribution π. In fact, it correlates with the computation time necessary for an algorithm to perform, as we will show in the next subsections. A practical way to evaluate different distributions is to define a utility space profile as a tuple (n, m, π). This parametrization will be used in the study of the complexity. It is important to note that a profile (n, m, π) must meet the consistency condition (5.4).

$$\pi_j \leq n \leq m \times \pi_j, \quad \forall j \in [1, m] \tag{5.4}$$

Such condition prevents cases like attempting to have $12-$ary constraints in a $7-$dimensional utility space.

Next, we show how to quantitatively assess complexity using the entropy measure. This will allow us to define the optimal search strategies.

5.3 Improvement using Low-complexity Search

5.3.1 Complexity Evaluation using Entropy

In the probabilistic sense, the sequence (5.2) could follow any distribution law. Furthermore, each distribution will have a complexity mirrored by the underlying utility hyper-graph, which will certainly affect the performance of any search algorithms. In the general case, this is known as the degree distribution of a graph. The complexity of a particular profile (n, m, π) is assessed using the information theoretical notion of entropy (5.5).

$$H(\pi) = -\sum_{j=1}^{m} \pi_j \log(\pi_j) \tag{5.5}$$

Entropy could in fact be used to measure the complexity of cognitive graphical models [47, 48], including any representation that uses the idea of degree distribution (5.2). Herein, entropy is meant to reflect complexity from a temporal standpoint.

As a general example, suppose that π is taking different forms $\pi_{i\in[1,5]}$, shown in (5.6).

$$\pi \begin{cases} \pi_1 = Uniform(m-1) \\ \pi_2 = Normal(\mu \in [2,4], \sigma \in [0,2]) \\ \pi_3 = Poisson(\lambda \in [2,6]) \\ \pi_4 = Power-law(\alpha = 2.3) \\ \pi_5 = Star \end{cases} \tag{5.6}$$

The idea behind the distributions is that it is possible to traverse a graph according to different distributions $\pi_{i\in[1,5]}$. Exploring the graph corresponds to a specific way of moving from one node to another as to perform any type of optimization. For example let us take a graph with

$m = 100$ nodes, represented in Figure 3.2(a), with different distributions (π_1:green, π_2:purple, π_3:blue, π_4:red and π_5:orange). For instance, traversing the graph based on a star distribution (π_5, in orange) corresponds to moving from the central node to a peripheral node, then back to the central node, etc. The exploration based on a uniform distribution gives a complete graph (π_1, in green) where we explore based on all the nodes, uniformly. The same logic applies for other distributions.

These distributions differ in their topologies, but most importantly in terms of complexity, or as we refer to it here, entropy. As shown previously in Figure 3.2(b), uniform strategy has the highest entropy and generates a complete graph (K_{m-1}). On the contrary, the star (S_{m-1}) strategy has the lowest entropy, with nodes connected to one single node acting like a hub. If we increase the number of hubs to few hubs, we get the power law strategy which is ranked right above the star strategy. Normal and poisson distributions fluctuate between π_1 and π_4 in terms of complexity, relatively to the values of μ, σ and λ. The uniform π_1 and the star π_5 act as complexity bounds.

Our usage of the entropy as a measure of the complexity is in fact another quantification of the level of interdependency between the issues and between the constraints. In this sense, how complexity arises could be assessed by the interdependency rate used in [40]. For instance, Table 5.1 illustrates how the interdependency rate changes for different strategies (Complete, Power-law, and Random). For all constraints, it increases as the number of constraints (m) grows. Additionally, we can see that on average, the complete distribution has the highest average interdependency rate. The random distribution comes second, and power-law comes third. This is also corroborated by the usage of entropy as to rank these distributions according to their complexities.

We will see in the next subsection how entropy, complexity and performance (time) relate to each other.

5.3.2 Complexity and Performance

Now, let us take a concrete example of exploration strategies and their underlying distributions, and let us evaluate the interplay between the entropy and the computation time needed by the search algorithm to find the optimal contract(s).

For instance, let us assume that we have 10 strategies $\pi_{k\in[0,9]}$ where each π_k is either a uniform distribution $\mathcal{U}$, a deterministic distribution $\mathcal{D}$ or a power-law distribution $\mathcal{PL}$. If the search algorithm is taken to be AsynchMP, then the computation time and the complexity of the underlying strategies are illustrated in Figure 5.1. The first observation is that both entropy and computation time fluctuate similarly, describing the same topology of the underlying strategy. Secondly, $\mathcal{U}$ is the most complex structure, since it possess the highest entropy and computation time as opposed to $\mathcal{D}$ and $\mathcal{PL}$.

It is possible to think about the complexity of strategy $\mathcal{U}$ from two standpoints. An analytical or cartesian view of the problem reduces the complexity to high-dimensionally [24]. From a graphical viewpoint, the distribution is perceived as representing a complete graph with one strong component having the highest number of possible connections. Both views reflect the difficulty of the search problem.

5.3.3 Optimal Strategy

Instead of the asynchronous mode of AsynchMP (randomly picking ν_{src} and ν_{dest}), we propose to use the distribution π as a prior, allowing us to optimize the message passing algorithm by taking into consideration certian topologies. For example, adopting a strategy $\pi \sim \mathcal{PL}$ allows us to focus on the hubs of the hypergrpah, *i.e.*, the factors with large numbers of issues. Let us call the new strategy AsynchMPi, which consists in performing the message passing within a set $\sigma_1 \subset (\Phi \bigcup \mathcal{I})$ of high degree nodes.

For a specific profile (n, m, π), and for two strategies AsynchMP and AsynchMPi, the idea is to see which one converges to the optimium faster, while being certain that both will find this optimum. That is, the same profile will be traversed and explored with different distributions

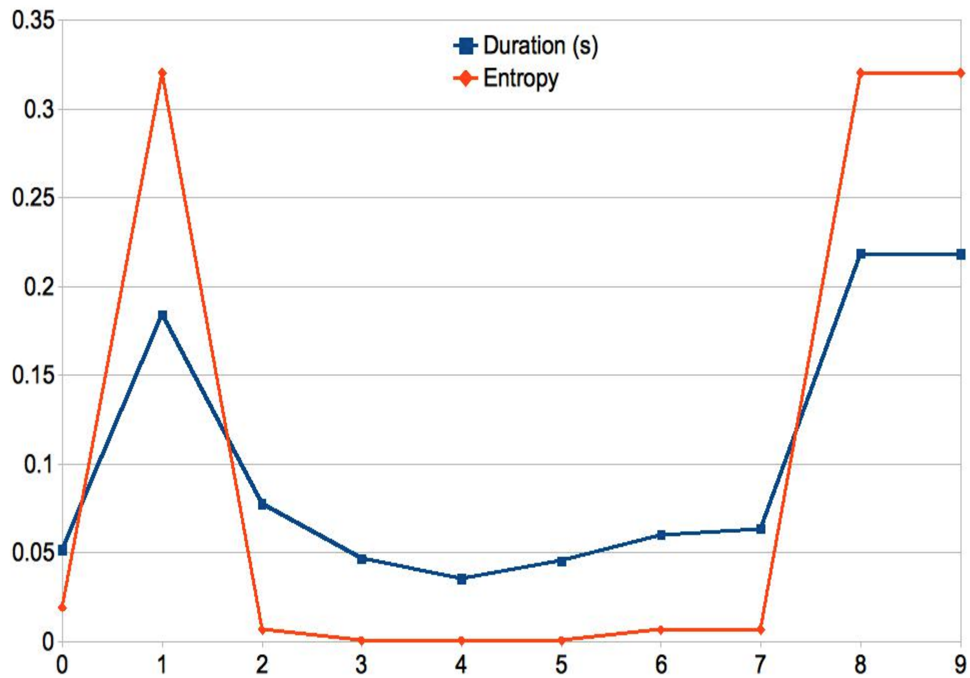


Figure 5.1: $H(\pi_j)$ and $\Delta(\pi_j)$ for $\pi_{j\in[0,9]} \in \{\mathcal{U}, \mathcal{D}, \mathcal{PL}\}$

throughout time.

In the following, we show how the set σ_1 is constructed.

1. Start by generating the sequence c_j, defined as in (5.7). The sequence c_j has the property of having the majority of its points clustered in the upper portion of the domain. It will be used to map the set of high degree nodes.

$$c_j = \begin{cases} \frac{1}{2} & \text{if } j = 0 \\ \\ \frac{1}{2} + \frac{1}{2} \times c_{j-1}^2 & \text{if } j \in [1, (n+m) \times 10] \end{cases} \tag{5.7}$$

2. Uniformly sample r points from c_j. The result is $\mathcal{U}(c_j)$

3. Generate the sequence s_π containing $\{\pi(\nu_i)\}_i$ by decreasing order (5.8). The sequence s_π approximates a power-law distribution.

$$s_\pi = \{i, \ldots, n' \mid \pi(\nu_i) \leq \cdots \leq \pi(\nu_{n'})\} \tag{5.8}$$

4. Generate the set s_σ according to Algorithm (7).

Input: Sequences $\mathcal{U}(c_j)$ and s_π

Output: Set s_σ of high connectivity nodes

begin
    $s_\sigma \leftarrow \emptyset$
    for $c \in \mathcal{U}(c_j)$ **do**
        if $c < \lceil c \rceil - \frac{1}{2}$ **then**
            $s_\sigma \leftarrow s_\sigma \cup \lceil c \rceil - 1$
        else
            $s_\sigma \leftarrow s_\sigma \cup \lceil c \rceil$
    return s_σ

Algorithm 7: Generation of high degree nodes s_σ

5.3.4 Discussion

The generation of the hyper-graph is performed using Algorithm 6. Depending on the nature of π, a particular topology will be generated. A uniform π generates a complete hyper-graph, a power-law π generates a scale-free hyper-graph and so on.

In the following we evaluate AsynchMP and AsynchMPi for 6 profiles $(100, 100, \pi_i)$, $i \in \{5, 6, 7, 8, 9, 10\}$. The profiles have decreasing complexity defined as $\pi_i(\Phi_j) \leq i \; \forall j$.

It is important to make sure that both strategies give the same expected optimal utilities, as shown in Figure 5.3(a). That is, for the different profiles specified by $i \in \{5, 6, 7, 8, 9, 10\}$ on the x axis, both strategies give the same optimal utility values, shown on the y axis. However, Figure 5.3(b) shows that restricting the message passing process to the high degree nodes (hubs) results in a drastic decrease in the computation time (in seconds) for the same profiles specified by $i \in \{5, 6, 7, 8, 9, 10\}$.

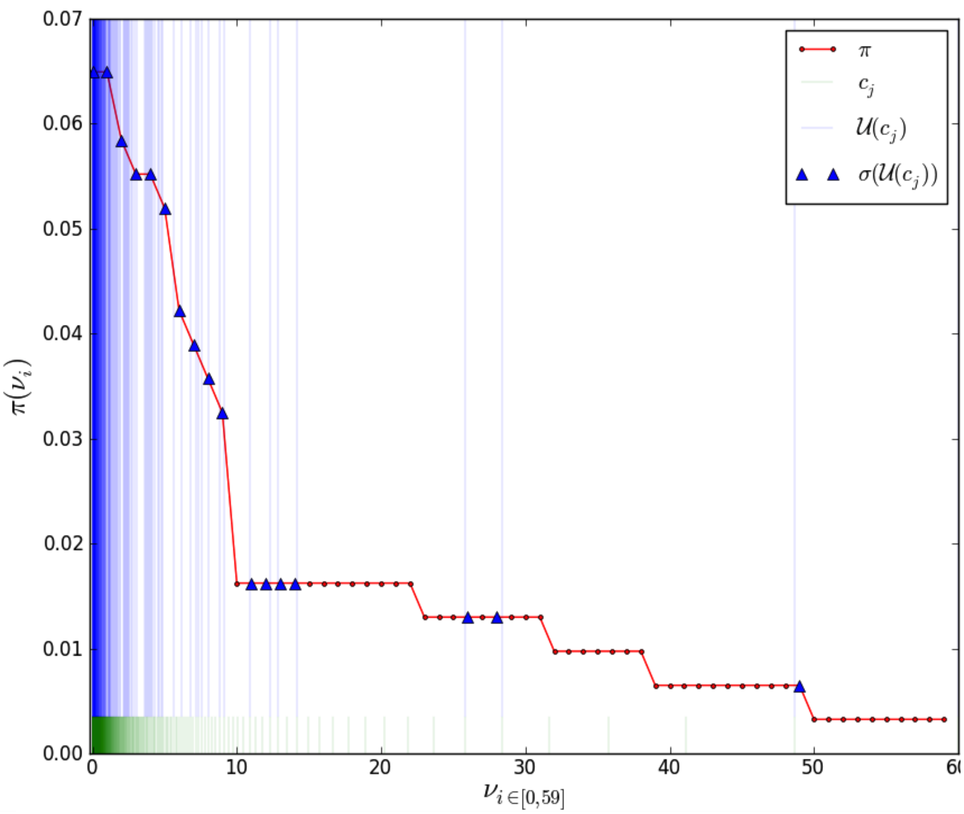


Figure 5.2: Generation of s_σ for $n, m, p, r = (10, 50, 5, 17)$

The way we can restrict the search to hubs is shown in Figure 5.2. That is, we sort the nodes $\nu_{i\in[0,59]}$ by decreasing degree ($\pi(\nu_i)$ is the degree of node ν_i) and we randomly sample from this sequence according to the process described in section 5.3.3. The resulting sequence of high degree nodes is shown as blue triangles in Figure 5.2.

We additionally observe that for small connectivity values ($\pi_1 = \pi_2$), the search process takes approximately the same amount of time despite the large number of issues and constraints $n = m = 100$. In fact, assessing the complexity of a utility space (respectively utility hypergrpah) must take into consideration the connectivity function π. In this sense, neither the dimension n nor the number of constraints m could objectively reflect this complexity. For instance, a profile $(100, 100, 1)$ is less complex than a profile $(10, 10, 2)$. In fact, $(100, 100, 1)$ is not a nonlinear utility space because $\pi(\Phi_j) = 1\ \forall j$. That is, each constraint contains one unique issue, *i.e.*, the whole utility is reduced to a sum of the partial utilities of the individual issues with, $n = m$.

Thus, the whole problem becomes linear with the utility function (5.9),

$$u(i_1, \ldots, i_j, \ldots, i_n) = \sum_{j=1}^{n} \phi_j(i_j) \tag{5.9}$$

with ϕ_j being the utility corresponding to constraint c_j. We note that the previous case is a degenerate case, and that generally, we ought to generate constraints with cardinalities greater or equal to 2.

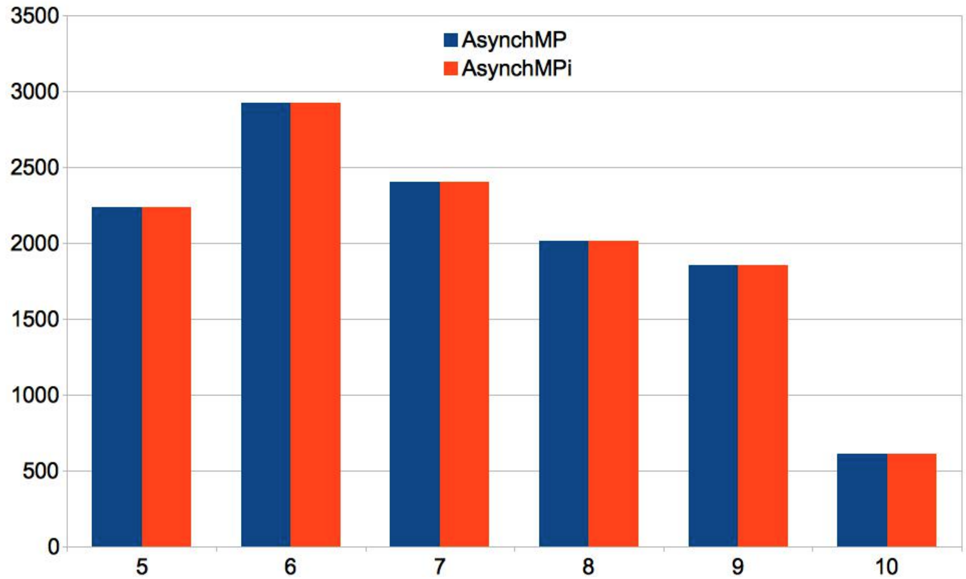
5.3.5 Summary

We introduced a new modular representation of utility spaces based on hyper-graphs. The exploration and search for optimal contracts is performed based on a message passing mechanism outperforming the sampling-based optimizers. Additionally, the model was evaluated in terms of complexity assessment showing that power-law topologies have lower complexity. Consequently, we provided an exploration strategy that searches the hyper-graph based on a power-law topology. Results show that such strategy outperforms drastically the synchronous message passing strategy.

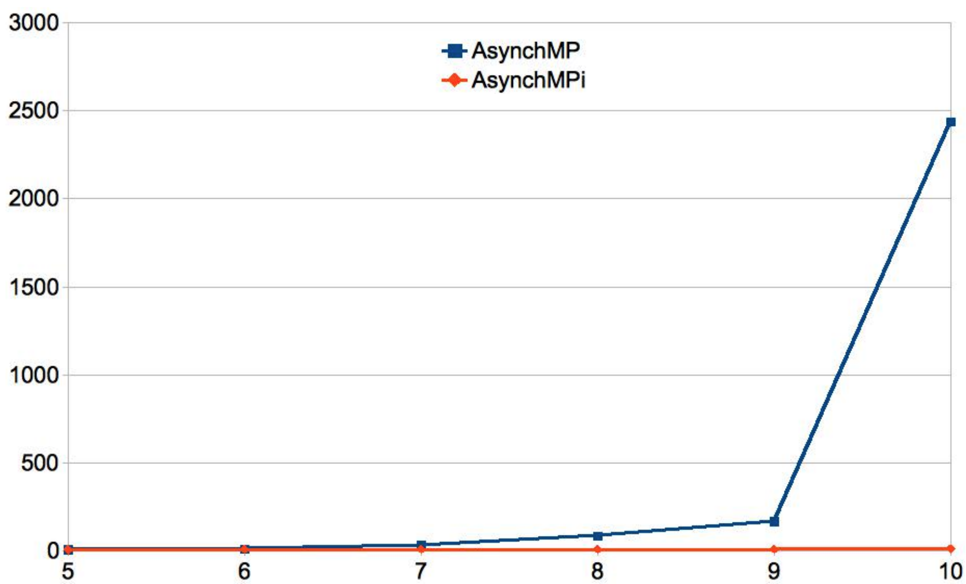
As a future work, we intend to exploit the structure of the hyper-graphs by proposing an hierarchical exploration scheme and evaluate it in a hierarchical negotiation scenario. Additionally, we intend to study the interdependence between the issues as to assess their importance and influence on the contracts optimality. Being able to assess the issues importance could in fact be used to simplify complex negotiation scenarios by focusing on the most important sub-contracts.

Table 5.1: Average interdependency rates for different profiles (n, m, π)

Strategy (π)	Number of issues (n)	Number of constraints (m)	Interdependency Rate
Complete	10	1	0.25
Complete	10	10	2.5
Complete	10	50	12.5
Complete	10	100	**25**
Complete	50	1	0.25
Complete	50	10	2.5
Complete	50	50	12.5
Complete	50	100	**25**
Complete	100	1	0.25
Complete	100	10	2.5
Complete	100	50	12.5
Complete	100	100	**25**
Power-law	10	1	0.25
Power-law	10	10	0.944
Power-law	10	50	1.688
Power-law	10	100	**2.688**
Power-law	50	1	0.25
Power-law	50	10	1.0912
Power-law	50	50	2.161
Power-law	50	100	**3.692**
Power-law	100	1	0.25
Power-law	100	10	0.901
Power-law	100	50	1.707
Power-law	100	100	**3.692**
Random	10	1	0.25
Random	10	10	0.961
Random	10	50	5.244
Random	10	100	**8.338**
Random	50	1	0.25
Random	50	10	0.842
Random	50	50	4.572
Random	50	100	**7.450**
Random	100	1	0.25
Random	100	10	0.816
Random	100	50	3.949
Random	100	100	**7.980**



(a) Utility



(b) Computation time

Figure 5.3: AsynchMP vs. AsynchMPi

Chapter 6

Applications in Complex Automated Negotiation

6.1 Introduction

In the following, we are interested in evaluating the behavior of a whole group of agents from the standpoint of social welfare [7, 90]. That is, we take the multilateral formulation of the problem where agents use our model as to represent their preferences as well as to bid in a multilateral multi-round mediated negotiation. This approach of the problem relates more to a consensus making situation [85], rather than the agreement-oriented approach found in bilateral negotiations. We show that under high complexity, the collective social welfare could be greater than the sum of the individual agents' best utilities.

6.2 Multilateral Negotiation (Consensus Building)

6.2.1 Setting

In the following we assume that N agents are bidding over n issues. Each agent i, $i \in [1, N]$, possesses an $n-$dimensional utility space. A contract, or a bid, is a vector of issue-values. The bid number 1 of agent i at round t is noted as $b^i_{t,1}$. Each agent will be sampling from her utility space using a particular algorithm. We use the same type of utility spaces described in section 4.2.1.

Let us assume that the N agents are using the AsynchMP algorithm for bidding over n issues with varying constraints (m) and cardinalities (p).

The general mediation protocol involves a mediator $\mathcal{M}$ receiving a bundle B_t^i from agent $i \in [1, N]$ at time $t \in [0, T]$. At round t, agent i's bundle, described in (6.1), contains n_t^i bids.

$$B_t^i = \{b_{t,1}^i, \dots, b_{t,k}^i, \dots, b_{t,n_t^i}^i\} \tag{6.1}$$

A bundle must respect a preference ordering $\preceq$ with regard to the agent's subjective utility, as it is shown in (6.2).

$$b_{t,1}^i \preceq \cdots \preceq b_{t,k}^i \preceq \cdots \preceq b_{t,n_t^i}^i \tag{6.2}$$

The feedback of $\mathcal{M}$ at t is a contract point $x_{\mathcal{M}}^t$ that the agents choose to accept or to ignore by providing a new bundle B_{t+1}^i. The process is repeated until reaching the deadline T with final contract $x_{\mathcal{M}}^*$.

Due to the randomized nature of the utility spaces, we will make the mediator $\mathcal{M}$ use a family $x_{\mathcal{M}}$ of sampling algorithms (6.3) that attempt to take into consideration the geometrical topology of the received bundles in order to generate candidate contracts. Herein, we are interested in evaluating the Social Welfare [4, 90] yielded by the individual usages of AsynchMP coupled with the sampling algorithms $x_{\mathcal{M}}$.

We start by defining the family of algorithms (6.3).

$$x_{\mathcal{M}} \begin{cases} x_{\mathcal{M}}^1 &= c(\{b_{t,n_t^i}^i,\ i \in [1, N]\}) \\ x_{\mathcal{M}}^2 &= c(\{\mathcal{N}(b_{t,n_t^i}^i),\ i \in [1, N]\}) \\ x_{\mathcal{M}}^3 &= c(\{c(C_t^i),\ \forall i \in [1, N]\}) \\ x_{\mathcal{M}}^4 &= c(\{f(C_t^i),\ \forall i \in [1, N]\}) \end{cases} \tag{6.3}$$

with C_t^i being the convex hull of bundle B_t^i and $\mathcal{N}(x)$ the neighbor set of x in $\mathbb{R}^n$. Functions are defined as following: c returns the centroid of a convex hull of a set of contract points; f, defined in (6.4), randomly picks a contract within a convex hull $x \subset \mathbb{R}^n$; g, defined in (6.5), returns a random point from a segment $[x, y] \in \mathbb{R}^{n \times 2}$; and h picks a random simplicial facet from a convex hull.

$$f(x) = \begin{cases} x & \text{if } x \in \mathbb{R}^n \\ g(c(x),\ (f \circ h)(x)) & \text{else} \end{cases} \tag{6.4}$$

$$g(x, y) = \alpha x + (1 - \alpha) y, \alpha \in [0, 1] \tag{6.5}$$

If the final contract is $x^*_{\mathcal{M}}$, we want to evaluate the social welfare using a set of social welfare functions (SWF). Particularly, (6.6a) as well as (6.6b) assigning high weights to low utilities and low weights to high utilities. Most importantly, we define a differential SWF in (6.6d). W_D evaluates the difference between what the mediator proposes to the agents ($x^*_{\mathcal{M}}$) and what the agents' subjective best options ($x^*_i,\ \forall i \in [1, N]$) are.

$$W_U = \sum_{i=1}^{N} u_i(x^*_{\mathcal{M}}) \tag{6.6a}$$

$$W_{\alpha U} = \sum_{i=1}^{N} \alpha_i u_i(x^*_{\mathcal{M}}) \tag{6.6b}$$

$$W_D = \sum_{i=1}^{N} u_i(x^*_{\mathcal{M}}) - \sum_{i=1}^{N} u_i(x^*_i) \tag{6.6c}$$

$$= W_U - W_{parts} \tag{6.6d}$$

6.2.2 Discussion

We take $N = 100$ agents, bidding over $n = 10$ issues with $(m, p) = (50, 5)\ \forall i \in [1, N]$, and where $\mathcal{M}$ is using $x^1_{\mathcal{M}}$, $x^2_{\mathcal{M}}$, $x^3_{\mathcal{M}}$ and $x^4_{\mathcal{M}}$. Figure 6.1 illustrates the obtained SWFs. It is interesting to notice that $W_D > 0\ \forall x^j_{\mathcal{M}}, j = 1, 3, 4$, reflecting the fact that on average, the agents get more than what they were expecting to get, with $W_U > W_{parts}$. This is due in fact to the complexity of the individual utility spaces and that one single agent cannot entirely explore her utility space to find all the high utility bids. However, using the mediation mechanism, all the agents' bids are collectively filtered by the mediator as to find optimal bids that increase the social welfare.

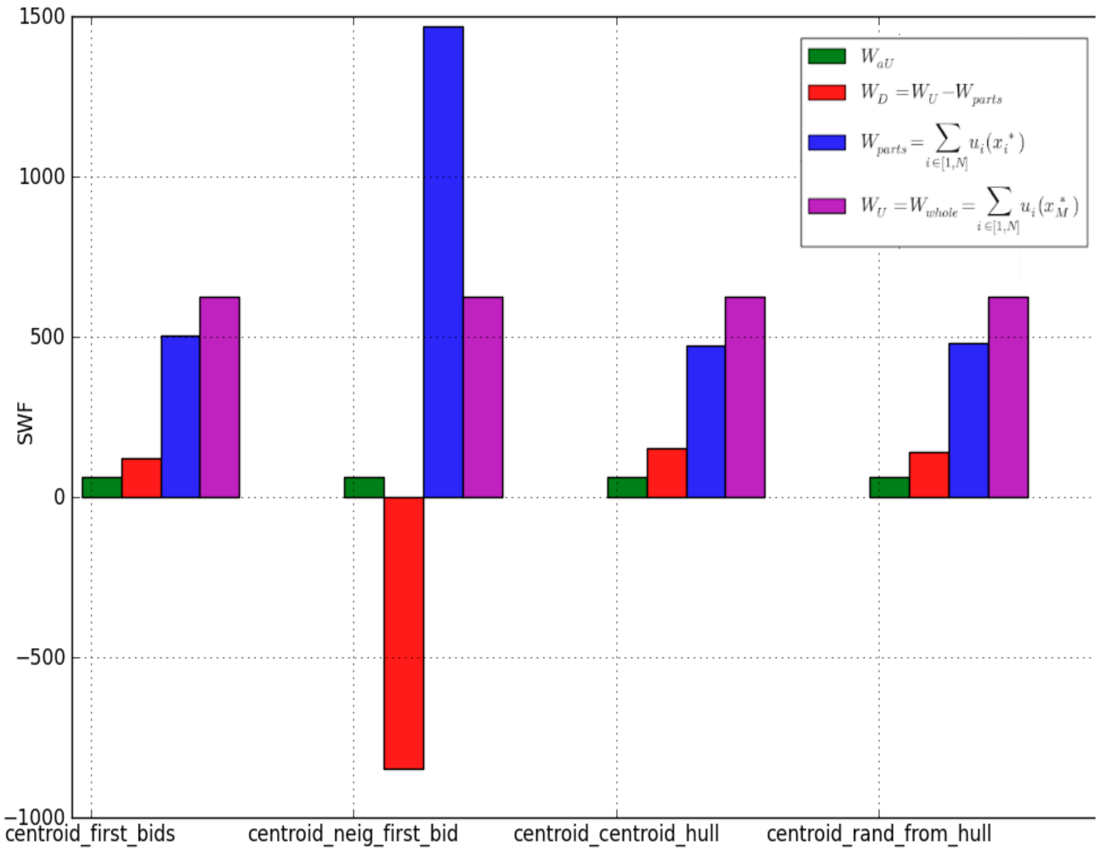


Figure 6.1: Social Welfare

In other words, the problem becomes as if the agents are searching in the same utility space and attempting to find all the Pareto-dominant bids.

In case $W_U \leq W_{parts}$, agents get less then expected on average, due to individual concessions or to bad contracts being reinforced by $\mathcal{M}$ and depending on the used algorithm ($x^2_{\mathcal{M}}$). Generally, $W_U \neq W_{parts}$ corroborates the assumption of nonlinearity by reflecting the idea that the nonlinearity of the (individual) agent's utility spaces is propagated from the issue-constraint level up to the agreement level through means of mediation (operators $x_{\mathcal{M}}$).

The convex aspects of the algorithms (6.3) and the nonlinearity of the utility spaces could in fact show the lack of structure and correlation between the different SWFs in Figure 6.1. This is to say that adopting a convex representation when searching for the optimal contracts is not appropriate whenever the utility space is highly nonlinear despite the efficiency of the local optimizers (AsynchMP).

6.3 Towards Opponent Modeling using GGMM

6.3.1 Introduction

In strategic encounters, agents usually do not share their preferences as to avoid exploitation. It is therefore common that an agent tries to model the opponent's behavior in order to predict her future offers. This could allow both agents to find a mutually satisfactory outcome, measured in terms of utility gain. In the case were we are certain that the utility space of the opponent is constraint-based, it would be interesting to find the parametrization that governs the opponent's utility model, despite the multitude of constraints' representations. This could in fact allow us to find the parameters that best fit the opponent offers and therefore her real preferences.

In this section, we provide a canonical and compact form for constraint-based utility functions that unifies a number of well known constraint-based utility functions, that is, cubic, bell and conic constraints. This leads us to new insights on how to model the preferences of opponents with constraint-based utilities.

Particularly, we propose a parametric representation that can fit any possible shape and that is defined as a Mixture of Generalized Gaussians. The new approximation is compact, unified and could reduce the complexity of the optimal contracts search as well as the generation of negotiation scenarios. Additionally, having a parametric form of this type of utility functions could in fact help in modeling the topology of the opponent's utility space by defining prior distributions on the unknown opponent's model and use Machine Learning techniques to predict the opponent moves [22]. This of course comes with the assumption that the agents' utility spaces are constraint-based, which is at least true for the non-linear negotiation domains of the ANAC competition [32, 75] in its fifth year.

6.3.2 Preliminaries

We start from the general setting of the non-linear multi-issue negotiation of [52]. That is, N agents are negotiating over n issues $i_{k\in[1,n]} \in \mathbb{I}$, with $\mathbb{I} = \{i_k\}_{k=1}^{n}$, forming an $n-$dimensional utility space. The issue k, namely i_k, takes its values from a set $\mathbb{I}_k$ where $\mathbb{I}_k \subset \mathbb{Z}$. A contract $\vec{x}$ is a vector of issue values $\vec{x} \in \mathcal{I}$ with $\mathcal{I} = \times_{k=1}^{n}\mathbb{I}_k$.

An agent's utility function is defined in terms of constraints, making the utility space a constraint-based utility space. That is, a constraint $c_{j\in[1,m]}$ is a region of the total $n-$dimensional utility space. We say that the constraint c_j has value $w(c_j, \vec{x})$ for contract $\vec{x}$ if c_j is satisfied by $\vec{x}$. That is, when the contract point $\vec{x}$ falls within the hyper-volume defined by c_j, namely $hyp(c_j)$. The utility of an agent for a contract $\vec{x}$ is thus defined as in (6.7).

$$u(\vec{x}) = \sum_{c_{j\in[1,m]},\ \vec{x}\in hyp(c_j)} w(c_j, \vec{x}) \tag{6.7}$$

In the following, we distinguish three types of constraints: Cubic constraints, Bell constraints and Conic constraints, shown previously in Figure 4.1. More details about constraint-based utility spaces and their usage could be found in [78, 77, 79].

6.3.3 Canonical Utility Representation

We start from the intuition that the Generalized Gaussian Distribution (6.8) could in fact represent a multitude of geometric shapes that could approximate the constraints' shapes we are dealing with. Precisely, the exponent ρ in (6.8) controls the asymptotic behavior of the function branches (right and left, in the one dimensional case). For the moment, we only focus on the one dimensional case before generalizing into higher dimensions.

$$g(x; \rho, \mu, \beta) = \frac{\rho}{2\beta\Gamma(\frac{1}{\rho})} e^{-(\frac{|x-\mu|}{\beta})^{\rho}} \tag{6.8}$$

Γ being the gamma function. As it is shown in Figure 6.2, choosing $\rho = 2$ gives the classical bell curve, or Gaussian distribution. If $\rho \to +\infty$ with $2|\rho$, the previously Gaussian-shaped curve will morph into square wave. Similarly, if $\rho \in [1, 2]$, we get a single-peacked function [18]

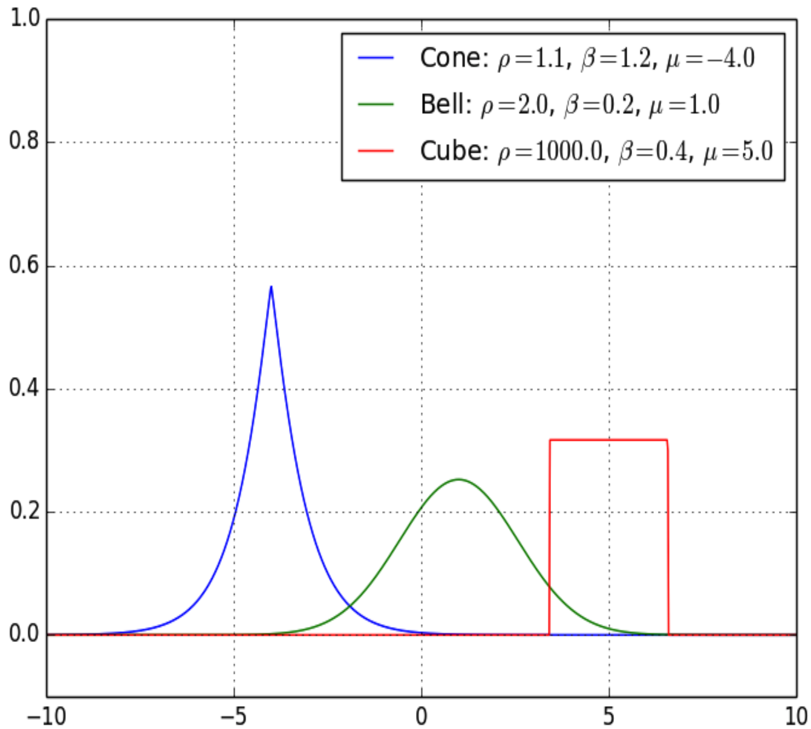


Figure 6.2: GGD for Constraints Representation

that could approximate a conic-shaped constraint in its one dimensional case. Thus, depending on the exponent ρ, it is possible to reproduce the three different structures (Cube, Cone, Bell). Particularly, let us take the angle φ defining the slope of the left branch (conversely $-\varphi$ for the right branch) as shown in Figure 6.3. The exponent ρ could in fact affects φ and thus contribute in morphing the shape of the bell. For instance, if ρ grows asymptotically, the general form (6.8) will look like a square curve with ($\varphi = \frac{\pi}{2}$) and as shown in (6.9).

$$\lim_{\substack{\rho \to +\infty \\ 2|\rho}} \varphi = \frac{\pi}{2} \tag{6.9}$$

Next, we propose the general parametric form we will be using, based on (6.8).

From (6.8) we construct a parametric form that corresponds to the partial weight term $w(c_j, \vec{x})$ in (6.7). By exchanging the parametrization $(2\rho^{-1}\beta\Gamma(\frac{1}{\rho}), \beta, \mu)$ with $(\gamma_j, \beta_j, \zeta_j)$, we re-scale the width, length, height for the n dimensions of the constraint, yielding (6.10). For example, if the contract point $\vec{x}$ is located in the hyper-volume (or concavity) defined by the n-dimensional

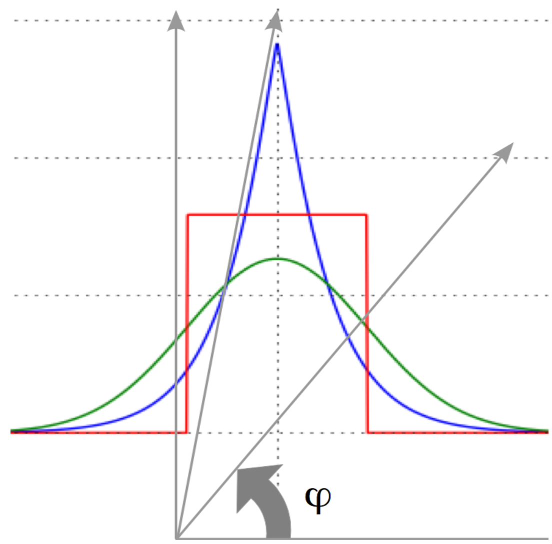


Figure 6.3: Controlling the shape using φ

function f_j, we get $f_j(\vec{x}; \pi_j) > 0$.

$$f_j(\vec{x}; \pi_j) = \gamma_j + \beta_j e^{-\sum_{i=1}^{n} |\zeta_{j,i} x_i - \mu_{j,i}|^{\rho_j}} \tag{6.10a}$$

$$\pi_j = \rho_j, \beta_j, \gamma_j, \delta_j, \mu_j, \zeta_j \tag{6.10b}$$

$$\vec{x} = (x_1, \ldots, x_i, \ldots, x_n) \tag{6.10c}$$

If we take all the m constraints into account, we get the total utility function, defined as a mixture of Generalized Gaussians (6.11), and compatible with its constraint-based counterpart (6.7).

$$u(\vec{x}) = \sum_{j=1}^{m} f_j(\vec{x}; \pi_j) \tag{6.11}$$

In the next section, we show how we can approximate the three types of constraints with regard to several dimensions.

6.3.4 Constraint's Approximation

It is important to find the right correspondence between the constraint and its fitting function f from the geometric characteristics of the constraint. Firstly, we take the two dimensional case.

To this end, we start from the general form (6.12).

$$f(x;\rho,\beta,\gamma,\delta,\mu,\zeta) = \gamma + \beta e^{\delta - |\zeta x - \mu|^{\rho}} \tag{6.12}$$

Depending on constraint shape, we mainly rely on ρ to firstly define the general shape, as in (6.13), and then we need to adjust few parameters in f in order to map the correct constraint's dimensions.

$$f(x;\rho,\beta,\gamma,\delta,\mu,\zeta) \begin{cases} 1 \leq \rho \leq 2 & \text{if Conic} \\ \rho = 2 & \text{if Bell} \\ 100 \leq \rho,\ 2|\rho & \text{if Cubic} \end{cases} \tag{6.13}$$

Let us take these cases one by one. Cubic:

In the following, we take the example of a $2-$dimensional cube, *i.e.*, a square. From (6.12) we compute the two bounds (6.14) that delimit the square according to the x axis. These specific points are in fact the inflection points provided from the derivative(s) of f. They need to be determined in order to fit the length (6.15) and height (β) of the square.

$$x_1 = \frac{1}{\zeta}(\frac{\rho-1}{\rho})^{1/\rho} - \frac{\mu}{\zeta} \tag{6.14a}$$

$$x_2 = -\frac{1}{\zeta}(\frac{\rho-1}{\rho})^{1/\rho} - \frac{\mu}{\zeta} \tag{6.14b}$$

And the length l is defined as in (6.15).

$$l = |\frac{2}{\zeta}(\frac{\rho-1}{\rho})^{1/\rho}| \tag{6.15}$$

We assume that in the example shown in Figure 6.4, the square in red is delimited by four points $[a,b,c,d]$. Given the square length $l = 19.38$, height $h = 4.19$ and its location in $\mathbb{R}^2$, we get the following parameterization: $\rho = 10^2$, $\beta = 1.5$, $\gamma = -2$, $\delta = 0$, $\mu = 0.432$ and $\zeta = 0.103$. In order to compare the accuracy of the fit we can measure the areas (6.16a) for the constraint (square $[a,b,c,d]$) and its approximation f.

$$\mathcal{A}_{square} = h \times l = da \times ab \tag{6.16a}$$

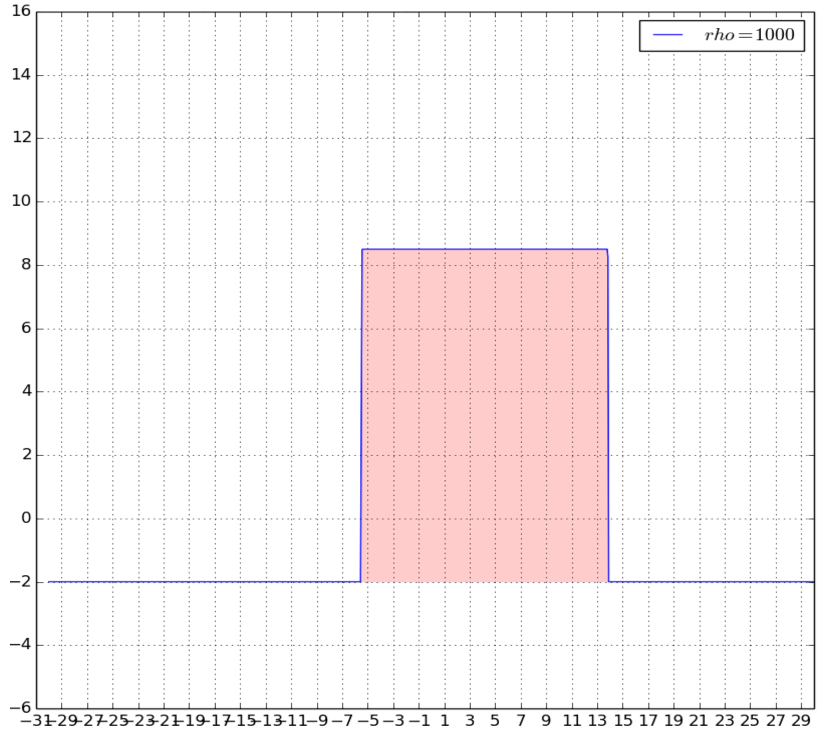


Figure 6.4: Square Approximation

$$\mathcal{A}_f = \int_{\frac{-\mu}{\zeta}-\frac{L}{2}}^{\frac{-\mu}{\zeta}+\frac{L}{2}} f(x) - \gamma \, \mathrm{d}x \tag{6.16b}$$

The result is $\mathcal{A}_f = 203.328$ and $\mathcal{A}_{l\times h} = 203.49$, which can lead us to (6.17).

$$\lim_{\rho\to\infty} \mathcal{A}_f = \mathcal{A}_{square} \tag{6.17}$$

Bell:

The bell shape is preserved by taking $\rho = 2$ and by defining the width, radius and height based on (β, ζ, μ). Cone:

The branches are straight segments, therefore (6.12) needs to be linear for those branches. Since the exponential component of (6.12) is of the form $e^{g(x)}$, $g(x)$ should be logarithmic to yield a linear representation. However, since we are restricted to the general form of (6.12) we ought to represent the logarithmic function as Taylor series. We start from the simple case (6.18).

$$log(x) = -\sum_{\rho=1}^{\infty} \rho^{-1}(1-x)^{\rho} \tag{6.18}$$

Rewriting (6.10) according to (6.18) gives (6.19).

$$f(x;\beta,\gamma,\delta,\mu,\zeta) = \gamma + \beta e^{\delta - \sum\limits_{\rho=1}^{\infty} \rho^{-1}|\zeta x - \mu|^{\rho}} \tag{6.19}$$

It is possible to linearize the branches and lower the two bottom inflection points by removing the ρ^{-1} term under the summation in (6.19) and preserve the general form (6.10), as shown in Figure 6.5. Additionally, the 1 in (6.18) is removed to fit the mean μ/ζ of the cone. The resulting approximation of two dimensional cone of height β, an apex angle proportional to ζ, and a center μ/ζ is given in (6.20).

$$f(x;\beta,\gamma,\delta,\mu,\zeta) = \gamma + \beta e^{\delta - \sum\limits_{\rho=1}^{\infty} |\zeta x - \mu|^{\rho}} \tag{6.20}$$

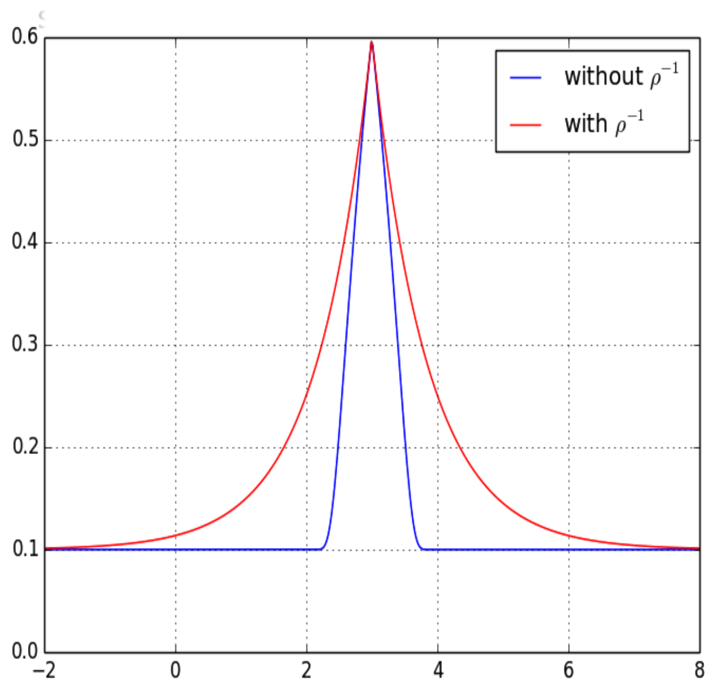


Figure 6.5: Taylor expansion with and without ρ^{-1}

Now, we take the general case of $n-$dimensional utility space with a contract point (6.10c). Most of constraints are symmetrical with respect to the dimensions, which is due to the absolute value in (6.10a). Hence, we assume that $\delta = 0 \; \forall x_i, \; i \in [1, n]$. Similarly, we assume that ζ is invariant for all dimensions, although it could be defined in a specific way for each dimension and therefore yield a parallelepiped instead of a cube. Let us now take the constraints one by one.

Cubic

For an $n-$dimensional utility space, a cube-constraint utility function is represented as in (6.21). It relies only on the exponent ρ as it should be large ($\rho \simeq 10^3$) and even. In this case, ρ could control the precision of the fit by flexibly adjusting φ to any right angle of the cube.

$$f(\vec{x}; \rho, \beta, \gamma, \delta, \mu, \zeta) = \gamma + \beta e^{-\sum_{i=1}^{n} |\zeta x_i - \mu_i|^\rho} \tag{6.21}$$

We note that for cubes, β acts like the height of the cube and is equivalent to the utility that will be assigned to any contract contained in the cube.

Bell

We need to choose an exponent $\rho = 2$ as to fit an $n-$dimensional Gaussian distribution. It is important that the result in (6.22) is adequately fitted to the real dimensions of the bell to be approximated. This could be done by finding the right relationship between the width, radius, center of the bell and the parameters ζ and μ. For instance, by assuming that the width of the bell is equal to $2\sigma\sqrt{log(2)}$, σ being the standard deviation of (6.22).

$$f(\vec{x}; \beta, \gamma, \delta, \mu, \zeta) = \gamma + \beta e^{-\sum_{i=1}^{n} |\zeta x_i - \mu_i|^2} \tag{6.22}$$

Cone

For an $n-$dimensional utility space, a cone-constraint utility function is represented as in (6.23).

$$f(\vec{x}; \beta, \gamma, \delta, \mu, \zeta) = \gamma + \beta e^{-\sum_{\rho=1}^{\infty} \sum_{i=1}^{n} |\zeta x_i - \mu_i|^\rho} \tag{6.23}$$

We can clearly see the Taylor expansion in the exponent (6.23) necessary for the linearization of the exponential form.

6.3.5 Experimental Analysis

In this section, we propose few examples of $3-$dimensional cubic and conic constraints, and show how they could be approximate based on the general form (6.10). For each example, the constraint and its fitting function are visualized. Herein, we evaluate the alignment, or the fitting, between the constraints and their functional approximations. The bell constraint is unchanged, whether it is defined using our functional parametric form, or as a Bell, Gaussian or Normal distribution in the large sense. To this end, we use two approaches: one is by random sampling from the constraint and then counting the contracts that fall within the concavity of f by getting $f(\vec{x}) > 0$. The second approach computes the volumes of the constraint and compares it to the integral of the approximated function. If the approximation is adequate, bother measures should coincide.

For the cubes, we start by randomly selecting n contracts from the cube constraint. One contract $\vec{x} = (x_1, x_2, x_3)$ is selected by picking each x_i, $i \in [1, 2, 3]$, within the bounds defined by the width, length and height of the cube.

Figure 6.6 provide an example of such evaluation, where the 100% of the contracts fall in both the cube and the function f concavity. Now, we compute the volume of f as in (6.24).

$$\mathcal{V}_f = \int_{x=-\theta+\mu_1}^{\theta+\mu_1} \int_{y=-\theta+\mu_2}^{\theta+\mu_2} f(x; \rho, \beta, \gamma, \delta, \mu, \zeta) \, \mathrm{d}y \, \mathrm{d}x \tag{6.24}$$

with $\theta = \frac{1}{\zeta}(\frac{\rho-1}{\rho})^{1/\rho}$.

We find that both volumes coincide by yielding $V_f = 635.797$ and $V_{cube} = 636.0$. We also note that in this case, ρ affects largely the precision of the approximation as in (6.25).

$$\lim_{\rho \to \infty} \mathcal{V}_f = \mathcal{V}_{cube} \tag{6.25}$$

In the case of a conic constraint, random n contracts are selected from the cone constraint using Algorithm 8. As shown in Figure 6.7, performing the evaluation for $n = 100$ contracts gives 93% of contracts, with $V_{cone} = 50.26$ integral of f $V_f = 31.62$.

It is possible to enhance this precision by adjusting the ζ of the function until it covers the relatively important parts of the cone. Particularly, this is done by breaking the ζ into $\{\zeta_i\}_{1 \leq i \leq n}$

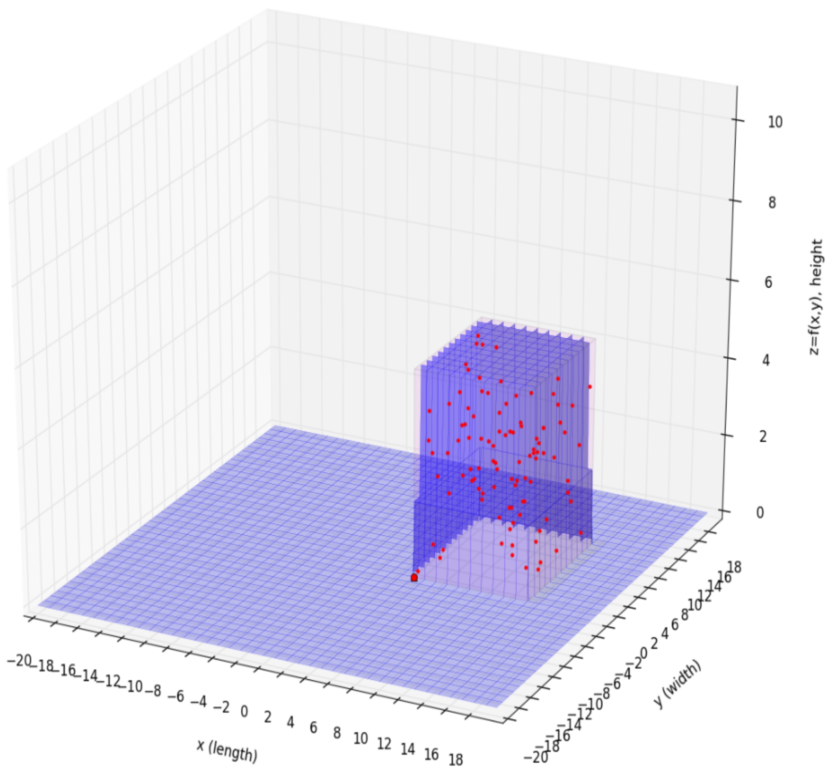


Figure 6.6: $3-$dimensional cubic constraint and its approximation

as to define a linear approximation of the cones sides, yielding the new approximation (6.26). However, this method requires additional sampling from the initial cone in order to define the new coefficients.

$$f(\vec{x};\beta,\gamma,\delta,\mu,\zeta) = \gamma + \beta e^{-\sum\limits_{\rho=1}^{\infty}\sum\limits_{i=1}^{n}|\zeta_i x_i - \mu_i|^{\rho}} \tag{6.26}$$

6.3.6 Summary

We have provided a practical way to approximate a family of constraint-based utility function based on one unified parametric form. The new representation reduces the complexity inherent to the definition of the constraints and adds more flexibility when hard constraints are present. Moreover, the new form allows an approximation of the existing utility functions as to reflect several economic types. This stems from the fact that any utility function could be translated into a wavelet-like transform built from a generalized Gaussian mixture.

Algorithm: RandFromCone

Input: Center $o(o_x, o_y, 0)$, radius r, height h

Output: One random contract in the cone

```
begin
    s ← (o_x, o_y, h)  // Apex s ∈ ℝ^3
    α ← RandomFrom(0, 1)
    return αRandomFromDisk(o, r) + s(1 − α)
```

Algorithm 8: Random Contract from Cone

Algorithm: RandomFromDisk

Input: Center $o(o_x, o_y, 0)$, radius r

Output: One random contract from the Cone disk

```
begin
    o' ← RandomOnCricle(o, r) // Apex o' ∈ ℝ^2
    α, β ← RandomFrom(0, 1)
    r_x ← o'_x α + o_x(1 − α)
    r_y ← o'_y β + o_y(1 − β)
    return (r_x, r_y, 0)
```

Algorithm 9: Random Contract from Cone Disk

As a future direction to be investigated, we are thinking about using the current parametric form for Opponent Modeling. In fact, we have shown how the constraints' descriptions could collapse to fewer parameters (6.10b) that could potentially be defined using prior distributions and the underlying hyper-parameters. Particularly, these parameters could be estimated using well known Machine Learning techniques, for instance as a Generalized Gaussian Process similar to [102]. Additionally, it is possible to extend the current canonical form to embed risk aversion and discounting for their importance in real negotiation settings. Added to (6.10b), discounting and risk aversion parameters could improve predicting the opponent's behavior.

Algorithm: RandomOnCricle

Input: Center $o(o_x, o_y, 0)$, radius r

Output: One random contract on a circle

begin

$\psi \leftarrow 2\pi RandomFrom(0, 1)$

$r' \leftarrow rRandomFrom(0, 1)$

$x \leftarrow o_x + r'cos(\psi)$

$y \leftarrow o_y + r'sin(\psi)$

return $(r_x, r_y, 0)$

Algorithm 10: Random Contract from Circle

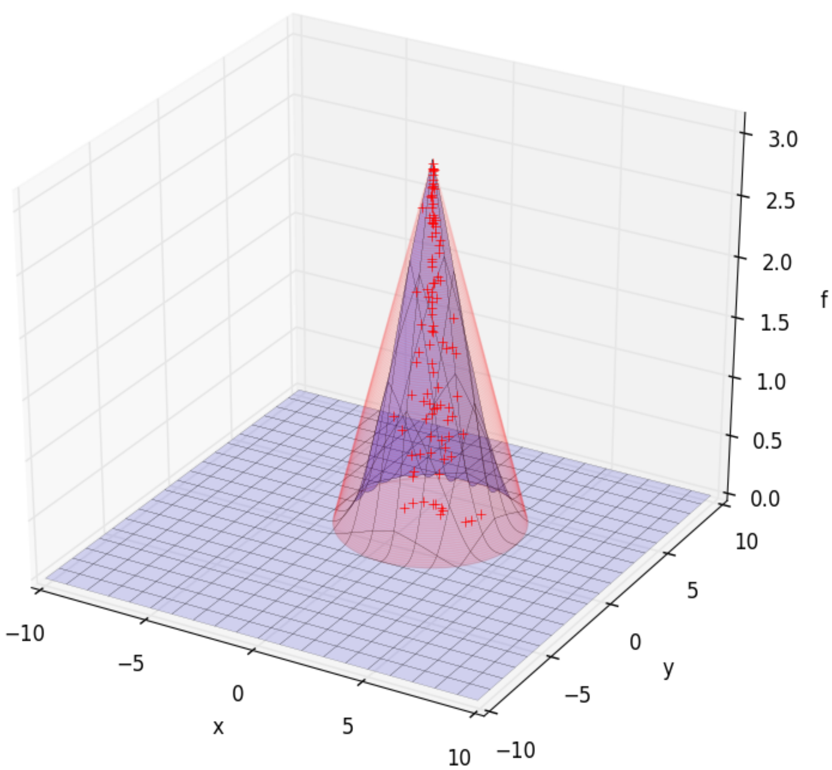


Figure 6.7: $3-$dimensional cone constraint and its approximation

Chapter 7

Applications in Preferences Elicitation and Collaborative Design

7.1 Asymptotic ME Principle for Preferences Elicitation

7.1.1 Introduction

Decision making involves generally two principal components. One dealing with the judgements about the uncertainties in the given world, whereas the other is related to the preferences over a set of possible consequences or outcomes. Several techniques have been proposed to help the decision maker in his analysis, by suggesting ways of formalization of the preferences as well as the assessment of the uncertainties. Although these techniques are established and proven to be mathematically sound, experience has shown that in certain situation we tend to avoid the formal approach by acting intuitively [62]. Especially, when the decision involves a large number of attributes and outcomes. In this case, most of the decision makers tend to use pragmatic and heuristic simplifications such as considering only the most important attributes and omitting the others [43].

Herein, we provide a model for decision making in situations subject to a partially available information and with a large predictive uncertainty with regards to the outcomes. We provide a formulation of this situation using an Information Theory approach and more precisely through the Maximum Entropy (ME) principle for preferences elicitation [2]. The considered situation is characterized by a "High-cognitive Load" [94, 16] or Bounded Rationality [89]. In fact, we think that the bounded rationality or the cognitive limits of the mind (of the decision maker, or the agent) could be re-interpreted as the inability to grasp the large number of alternatives, attributes, outcomes and uncertainties. Thus, *the bounded-rationality is caused by the unbounded possibilities* that the decision maker is facing, which yields a huge amount of information to be considered. This amount of information is intractable and yet infinite for a simple optimization technique that seeks an optimal choice given the available information. Especially, when the decision has to be made in a finite amount of time despite the infinite number of possibilities. It is worth mentioning that this problem could be seen as an instance of the Frame Problem [80], in the context of preferences elicitation. In such situations, the elicitation of a good utility function is not a realistic option and one should resort to other, less quantitative forms of preference representation. For instance, adopting a *Ceteris Paribus* preferential statements might be an option, as it was widely discussed in Philosophical Logic and Artificial Intelligence [23]. In fact, we ought to focus on what needs to be known (and represented) about a given environment and to omit what can be safely omitted. It is under such *Ceteris Paribus* assumption that we will propose our model to deal with the infinity of outcomes. Our main result could be seen as an extension of the ME principle whenever the set of prospects is countably infinite, and involving uncertainties. We think that using entropy methods enables us to capture the characteristics of such decision problems, as well as to the way solutions are realistically established by humans.

In the context of utility representation, several models were provided. For instance, [14] proposed a model which takes into consideration the uncertainties over the utilities by considering a person's utility function as a random variable, with a density function over the possible outcomes. In the work of [2], probability and utility are considered with some analogy, thus yielding the notion of joint utility density function for multiple attributes. The application of Information Theory to Utility Theory gave a new interpretation of the notion of utility dependence, but most importantly, it allowed the elaboration of the MEU principle [2] as a way to assign utility values when only partial information is available. The same author assumed a continuous entropy measure on a continuous bounded domain of outcomes, which is true when the support of the distribution is finite. However, this continuity hypotheses does not hold when the support is countably infinite, which makes the information measure discontinuous in all probability distributions with countably infinite support [50]. In other words, when the number of plausible outcomes that could be elicited by the ME utility function is countably infinite. In our approach, we adopted the same ME utility elicitation, but after establishing the behavior of the entropy with regard to the infinity of the outcomes.

7.1.2 Convergence of preferences

We start by providing our main framework, by adopting the utility-probability analogy and its usage for entropy maximization. We also provide a utility-based interpretation of the notion of convergence. Therefore, we start by stating it probabilistically, and then use utility increment vectors to define the utility convergence. Then, we define the continuity of the *Shannon* entropy with respect to a distance metric D_U. These are the first steps before treating in the next section the ME principal for countably infinite prospects.

Probability-Utility analogy

The analogy between probability distributions and utility was established in [2]. Therefore, we use the same formalism as to name the utility vectors and the utility density functions. As in [2], we assume that the utility values are represented as a vector, namely a utility increment vector ΔU_i referring to a discrete utility function with one attribute i. As in (7.1), the vector ΔU_i contains the utility values $\{\Delta u_j\}_{j=0}^{k}$ of the $k+1$ ordered and discrete outcome $\{x^j\}_{j=0}^{k}$, starting from the lowest outcome x^0 to the highest outcome x^k, named x^*. We also define the sequence $\Delta U_{(n)}$ of n utility increment vectors as in (7.2).

$$\Delta U_i = (\Delta u_0, \ldots, \Delta u_j, \ldots, \Delta u_k), \tag{7.1}$$

$$\sum_{j=0}^{k} \Delta u_j = 1,\ \Delta u_j \geq 0\ \forall j \in [0, k]$$

$$\Delta U_{(n)} = \{\Delta U_i\}_{i=1}^{n} \tag{7.2}$$

A sequence of utility increment vectors can be compared to a sequence of random variables. It is built according to an analogy with a probability mass function $P = (p_1, \ldots, p_k)$. Each discrete utility increment vector ΔU_i corresponds to a normalized utility $U_i(x)$ function as in (7.3).

$$U_i(x) = \int_{x^0}^{x} u_i(x)\,\mathrm{d}x \tag{7.3}$$

$$u_i(x) = \frac{d}{dx} U_i(x) \tag{7.4}$$

That is, for a given prospect $x \in [x^0, x^*]$, $U_i(x)$ is determined by integrating the utility density function $u_i(x)$ [2] from the least preferred prospect x^0 up to the prospect x (the accumulated welfare from x^0 to x). The normalized utility function $U_i(x)$ has the same mathematical properties as a cumulative distribution function (CDF) as both are non-decreasing and range from zero to one (7.5).

$$\begin{gathered} 0 \leq U_i(x) \leq 1 \\ \frac{d}{dx} U_i(x) \geq 0 \; \forall x \end{gathered} \tag{7.5}$$

All along the next sections, we will make usage of these notions of utility increment vector ΔU, the sequence of utility increment vectors $\Delta U_{(n)}$ as well as the utility density function $u(x)$.

Distance measure

The distance between two utility functions is defined based on the notion of similarity that could exist between them. By similarity, we mean the strategic equivalence [62], *i.e.*, two utility functions u_1 and u_2 are strategically equivalent, written $u_1 \sim u_2$, if and only if they imply the same preference ranking for any two lotteries. For instance, to compare two utility functions we can define the total variational distance that reflects the difference between the accumulated welfare all along the considered prospects. In the discrete case of two utility increment vectors, it is reduced to the $\mathcal{L}_1$-norm as it is shown in (7.6).

$$D_V(\Delta U_1, \Delta U_2) = \sum_j |\Delta U_{1,j} - \Delta U_{2,j}| \tag{7.6}$$

where D_V stand for the variational distance, and $\Delta U_{i,j}$ is the j^{th} element of ΔU_i. In case we are comparing two utility increment vectors ΔU_1 and ΔU_2 having respectively different dimensions

L and M, (7.7) becomes:

$$D_V(\Delta U_1, \Delta U_2) = \sum_{j=1}^{L} |\Delta U_{1,j} - \Delta U_{2,j}| + \sum_{j=L+1}^{M} |\Delta U_{2,j}| \tag{7.7}$$

We can also use the divergence between two utility density functions (7.4) based on the *Kullback-Leibler* divergence given in (7.8), for both discrete and continuous cases.

$$D_U(\Delta U_1, \Delta U_2) = \sum_{j} \Delta U_{1,j} \; ln(\frac{\Delta U_{1,j}}{\Delta U_{2,j}}) \tag{7.8a}$$

$$D_u(u_1, u_2) = D_{KL}(u_1||u_2) = \int_x u_1(x) \; ln(\frac{u_1(x)}{u_2(x)}) \tag{7.8b}$$

where we adopt the convention $D_u(u_1, u_2) = 0$ if $u_2(x) = 0$ but $u_1(x) > 0$ for some x. Moreover, based on the *Pinsker*'s inequality [100] we have (7.9).

$$\frac{1}{2} \, [D_V(\Delta U_1, \Delta U_2)]^2 \leq D_U(\Delta U_1, \Delta U_2) \tag{7.9}$$

Both divergence (7.8) and the variational distance (7.6) can be used as measures of the difference between two utility increment vectors (respectively utility density functions) defined on the same set of prospects. However, once applied to utility functions, *Pinsker*'s inequality has the important implication that for two utility increment vectors ΔU_1 and ΔU_2 defined on the same set of prospects, if $D_U(\Delta U_1, \Delta U_2)$ (or $D_U(\Delta U_2, \Delta U_1)$) is small, then so is $D_V(\Delta U_1, \Delta U_2)$ (or $D_V(\Delta U_2, \Delta U_1)$). Furthermore, for a sequence of utility increment vectors $\Delta U_{(n)}$, as $n \to \infty$, if $D_U(\Delta U, \Delta U_n) \to 0$, then $D_V(\Delta U, \Delta U_n) \to 0$, *i.e.*, the convergence in divergence is a stronger notion of convergence than the convergence in variational distance. Thus, we will use the divergence measures (7.8) as to define the continuity of the *Shannon* entropy, in the sense that we study the convergence of a sequence of utility increment vectors as well as their entropies. Using (7.8) fits better with the idea of maximum likelihood estimation we might use in order to use the MEU.

Utility Convergence

From a probabilistic viewpoint, we say that a sequence P_n, (with a CDF F_n), is converging in distribution to a distribution P (with a CDF F), and noted as in (7.10).

$$P_n \xrightarrow{d} P \tag{7.10}$$

$$\text{with} \quad \lim_{n\to\infty} F_n(x) = F(x)\ \forall x \in \mathbb{R} \tag{7.11}$$

where F is continuous in x. As we mentioned in section 7.1.2, a utility function can be represented by a CDF as in (7.3). If we assume that:

$$F(x) = \int_{x_{min}}^{x} f(x)dx \tag{7.12}$$

and if (7.3) is an analogy with (7.12), then what is the utilitarian significance of the sequence $F_n(x)$? To understand this setting, we rely on Merging Theory [93] [74], which studies whether the beliefs of an agent, once updated after successive realizations of the process, converge to the true distribution. Now, we can interpret a sequence of preferences $\Delta U_{(n)}$ as a process over time. In fact, we can consider the sequential decision problem as in the case of a Bayesian agent observing the successive realizations of a discrete stochastic process $\{\Delta U_{(n)} |\ n \in T\}$ on the space of outcomes, indexed by n, and where n varies over a time index set T. The evolution of the process is announced round after round to the observer who observes a true distribution and holds an a priori preferences' belief on the process. We consider the preferences merging, namely, the convergence of this sequence of preferences $\Delta U_{(n)}$ to the limiting preferences ΔU. Firstly, let's assume that the decision maker is given the task of assessing a utility increment vector (7.13) for $k+1$ outcomes at time n.

$$\Delta U_{(n)} = (\Delta {u_0}^{(n)}, \ldots, \Delta {u_k}^{(n)}) \tag{7.13}$$

For instance, the assessment of the preferences $\Delta U_{(n+1)}$ is different from $\Delta U_{(n)}$, to the extent that new information have been made available to the decision maker, and used to update the preferences. A sequence $\Delta U_{(n)}$ converging to ΔU can be written as in (7.14).

$$\Delta U_{(n)} \xrightarrow{U} \Delta U \tag{7.14}$$

This notion of utility convergence reflects the idea that we expect to see the next outcome in a sequence of utility increment vectors $\Delta U_{(n)}$ to become better and better modeled by ΔU. The yielded convergence is expressed by the limit (7.15), for the discrete case.

$$\lim_{n\to\infty} \Delta U_{(n)} = \Delta U \tag{7.15}$$

This notion of convergence will be used to define the continuity of the *Shannon entropy* with regard to utility.

7.1.3 Asymptotic MEU

In this section, we provide the MEU principle, in the case of a countably infinite support. We start by providing the definition of the support of utility function (respectively a utility increment vector). Let a utility function u defined on the outcomes set D. The set of all the values that u could take is $\{u(x)|x \in D\}$.

Definition 9. *The support of u denoted by S_u, is the set of all the outcomes x in a set D, such that $u(x) > 0$.*

$$S_u = \{x|x \in D, u(x) > 0\} \tag{7.16}$$

If $S_u = D$, we say that u is strictly positive. Otherwise, u contain null utility values, which correspond to the undesirable, unwanted outcomes. The support of a discrete utility function will be used in the case of a large number of outcomes, literally approaching infinity. It is the case of utility functions that vanish for a certain number of outcomes ($u(x_j) = 0$), while being strictly positive for another infinite number of outcomes ($u(x_{i\neq j}) > 0$). In the discrete case we have:

Definition 10. *The support of ΔU is the set of all the indexes j such as $\Delta u_j \in \Delta U$ and $\Delta u_j > 0$.*

$$S_{\Delta U} = \{j \in \mathbb{N},\ \Delta u_j \in \Delta U \mid \Delta u_j > 0\} \tag{7.17}$$

Problem Statement and Method

Before stating the problem, let's give an example of situations, where the notion of infinity could arise while considering the outcomes. Let's consider a multi-attribute utility function u over a set of attributes $a = \{a_1, \ldots, a_j, \ldots, a_n\}$, with $Domain(a_j) = D_j$. We want to define a preference ranking over the complete assignments on a.

Each complete assignment to a corresponds to a possible outcome of the decision makers action. Given the sizes of the attributes' domains, we can compute the number of possible assignments as $np = \prod_{j=1}^{n} card(D_j)$. Now, let's define another utility function u', over the domain $D = \times_{j=1}^{n} D_j$. It is obvious that the complexity of the assessment of u' grows up exponentially as the domains' sizes of the n attributes a_j are growing up.

Now, consider the case of complex systems subcontracting and manufacturing, for example, a *Boeing 747-400*, which is made in 33 countries and contains 6×10^6 parts [51]. Let's consider that each part of the plane is an attribute, therefore, designing the plane is reduced to finding and instantiating the attributes by assigning values from their domains. In the end, the best design will be chosen amongst all the possible instantiations, in other words, the outcomes. While keeping in mind the goal of providing efficient and automated tools for preference elicitation, we can highlight the considerable effort and complexity that will arise if we want to build a utility function over such possible set of outcomes. Given 6×10^6 interdependent attributes with a maximal size domain d, even if $d = 2$ (which is unlikely since we are dealing with complex systems), the number of possible combinations is $np = 2^{6 \times 10^6}$, which is too large to be treated quantitatively ($np \sim \infty$).

It is under such assumptions of infinity that we will adopt a *Ceteris Paribus* preferential statement, notably statements in which “all else being equal", and by varying a number of variables (in our case, $[1, \ldots, L]$) while holding the others ($[L+1, \ldots, M]$) constant [23]. In other words, we will reduce the actual frame of M infinite outcomes (ΔU) to L outcomes ($\Delta U'$), which could be reasonably assessed, under *Ceteris Paribus*.

In our case, we are assessing a utility increment vector ΔU_{meu} containing all the preferences as discrete elements (7.1). We also take the support $S_{\Delta U}$ as countably infinite, which makes the assessment process more difficult, as we mentioned in the previous example. The idea here is to use another utility increment vector $\Delta U'$ to contain the reasonably assessed preferences of the decision maker, according to his subjective belief. Then, we try to estimate ΔU with a new utility increment vector ΔU_{meu} with respect to the ME principle and by minimizing $D_U(\Delta U, \Delta U')$, described in (7.8).

Let $\Delta U = (\Delta u_1, \ldots, \Delta u_M)$ be the true utility increment vector to be assessed, where M is a large number that tends to infinity. We can see M as the number np we provided in the example above. Let $\Delta U' = (\Delta u'_1, \ldots, \Delta u'_L)$ be the utility increment vector that the decision maker was able to assess, due to the reduction of the number of outcomes to L ($L < M$), under *Ceteris Paribus*. We propose then to find the entropy maximization utility increment vector with respect to the minimal possible distance between ΔU and $\Delta U'$, that is, $D_U(\Delta U, \Delta U') \leq \epsilon$. One way to solve this maximization problem is to adopt the approach used in [2], by finding the continuous utility function u^* that maximizes the entropy (7.18).

$$u^*(x) = \arg\max_{u(x)} H(u(x)) \tag{7.18}$$

Since $S_{\Delta U}$ is countably infinite, (18) cannot be solved with *Lagrange* multipliers and simple derivation methods. In fact $H(\Delta U)$ is discontinuous whenever $S_{\Delta U}$ is countably infinite, and most importantly when the continuity measure is based on the *KL-distance* [50].

In the next section, we introduce the *Shannon* entropy and define its continuity as well as its discontinuity whenever the considered utility increment vector has an infinite support.

Continuity of the Entropy Measures

The *Shannon* entropy measures are functions mapping a probability distribution to a real value. They can be described as the measure of uncertainty about a discrete random variable X having a probability mass function p.

Definition 11. *The entropy H(X) of a random variable X is:*

$$H(X) = -\sum_{x} p(x) \log p(x) \tag{7.19}$$

We adopt the convention that summation is over the support of the given probability distribution, in order to avoid undefined cases. An important characteristics of *Shannon* entropy, is that it measures the spread of a probability distribution and therefore, achieves its maximum value when the distribution assigns equal probabilities to all the outcomes. This concept was used as a method to assign prior probability distributions that maximize *Shannon* entropy measure under partial information constraints. It is possible to apply *Shannon* entropy measures to a utility increment vector reflecting the spread of the prospects [2] as in (7.20).

$$H(\Delta u_1, \ldots, \Delta u_n) = -\sum_{i=1}^{n} \Delta u_i \log(\Delta u_i) \tag{7.20}$$

In the case where the outcomes are finite, the *Shannon* entropy measures are continuous function. We propose to focus on the case where the entropy measure H is applied to utility increment vectors ΔU with countably infinite elements, situation reflecting the high uncertainty. More precisely we are interested in studying the continuity of H with respect to the distance measures we established in the section 7.1.2. For instance, entropy is discontinuous with respect to the *Kullback-Leibler* divergence [50]. We should highlight that the underlying utility functions follow the axioms of normative utility functions [98], which gives (7.21).

$$\int_{x} u(x) = 1 \tag{7.21}$$

We propose to define the continuity of a function f that will be lately extended into the entropy measure H.

Definition 12. *Let π_X be the set of all utility density functions on the set of outcomes X and let $u \in \pi_X$. $f : \pi_X \to [0,1]$ is continuous at u if, given any $\epsilon > 0$, $\exists \delta > 0$ such that: $\forall u' \in \pi_X$: $D_u(u,u') < \delta \implies |f(u') - f(u)| < \epsilon$.*

For the discrete case, we have the following definition.

Definition 13. *Let π_k be the set of all utility increment vectors defined for k outcomes, and let $\Delta U \in \pi_k$.*
$f : \pi_k \to [0,1]$ is continuous at ΔU if, given any $\epsilon > 0$, $\exists \delta > 0$ such that: $\forall \Delta U' \in \pi_k$: $D_U(\Delta U, \Delta U') < \delta \Longrightarrow |f(\Delta U') - f(\Delta U)| < \epsilon$.

If f fails to be continuous at the utility density function u (resp. ΔU), then we say that f is discontinuous at u (resp. ΔU). Given the notion of convergence we defined in section 7.1.2, we can provide the following definitions of the discontinuity of the function f.

Definition 14. *Let π_X be the set of all utility density functions on the set of prospects X and let $u \in \pi_X$. A function $f : \pi_X \to [0,1]$ is discontinuous at u if there exists a sequence of utility density functions $u_{(n)} \in \pi_X$ such that :*

$$\lim_{n \to \infty} D_u(u_{(n)}, u) = 0 \tag{7.22}$$

but $f(u_{(n)})$ does not converge to $f(u)$, i.e.,

$$\lim_{n \to \infty} f(u_{(n)}) \neq f(u) \tag{7.23}$$

Similarity, for the discrete case we have:

Definition 15. *Let π_k be the set of all utility increment vectors defined for k prospects. Let $\Delta U \in \pi_k$.*
A function $f : \pi_k \to [0,1]$ is discontinuous at ΔU if there exists a sequence of utility increment vectors $\Delta U_{(n)} \in \pi_k$ such that :

$$\lim_{n \to \infty} D_U(\Delta U_{(n)}, \Delta U) = 0 \tag{7.24}$$

but $f(\Delta U_{(n)})$ does not converge to $f(\Delta U)$, i.e.,

$$\lim_{n \to \infty} f(\Delta U_{(n)}) \neq f(\Delta U) \tag{7.25}$$

Discontinuity

In this section, we establish the discontinuity of H at any utility increment vector ΔU having a countably infinite support. Let $\Delta U_{(n)}^{(a,b)}$ be a sequence of utility increment vectors with the real parameters a and b. We will use this sequence to show that H is discontinuous at $\Delta U_1 = (1, 0, 0, \ldots)$.

For fixed real numbers a and b and an integer n, such as $a > 1$ and $b > 0$ and $n > a$. We define $\Delta U_{(n)}^{(a,b)}$ as in (7.26).

$$\Delta U_{(n)}^{(a,b)} = \{1 - (\frac{\log a}{\log n})^b, \frac{1}{n}(\frac{\log a}{\log n})^b, \ldots, \frac{1}{n}(\frac{\log a}{\log n})^b, 0, 0, \ldots\} \tag{7.26}$$

Based on our definition of convergence in section 7.1.2, we show that the sequence $\{\Delta U_{(n)}^{(a,b)}\}$ converges to $\Delta U_1 = (1, 0, 0, \ldots)$. Computing the distance between the vector ΔU_1 and the parametrized vector $\Delta U_{(n)}^{(a,b)}$ gives (7.27).

$$D_U(\Delta U_1, \Delta U_{(n)}^{(a,b)}) = -\left(\log(1 - \left(\frac{\log a}{\log n}\right)^b\right) \tag{7.27}$$

We have $\Delta U_{(n)}^{(a,b)} \xrightarrow{U} \Delta U$, which is given in (7.28).

$$\lim_{n \to \infty} D_U(\Delta U_{(n)}^{(a,b)}, \Delta U_1) = 0 \tag{7.28}$$

Then, the entropy of $\Delta U_{(n)}^{(a,b)}$ is given by (7.29).

$$\begin{aligned} H(\Delta U_{(n)}^{(a,b)}) = &- \left[1 - \left(\frac{\log a}{\log n}\right)^b\right] \log \left[1 - \left(\frac{\log a}{\log n}\right)^b\right] \\ &- n \left[\frac{1}{n}\left(\frac{\log a}{\log n}\right)^b\right] \log \left[\frac{1}{n}\left(\frac{\log a}{\log n}\right)^b\right] \end{aligned} \tag{7.29}$$

Hence, we have (7.30). The proof is provided in A.1.

$$\lim_{n \to \infty} H(\Delta U_{(n)}^{(a,b)}) = \begin{cases} 0 & \text{if } b > 1 \\ \log a & \text{if } b = 1 \\ \infty & \text{if } 0 < b < 1 \end{cases} \tag{7.30}$$

Proposition 1. *Based on Definition 7. and if we take $f = H$, $a > 1$ and $0 < b \leq 1$ in (7.30), we have:*

$$\lim_{n\to\infty} H(\Delta U_{(n)}^{(a,b)}) = \infty \neq H(\Delta U_1) \tag{7.31}$$

$$\textit{but} \qquad \lim_{n\to\infty} \Delta U_{(n)}^{(a,b)} = \Delta U_1 \tag{7.32}$$

Therefore, we can state that H is discontinuous at the utility increment vector $\Delta U_1 = (1, 0, 0, \ldots)$.

Bound and Majorization

Given (7.30), we propose to find a bound to η in (7.33).

$$\eta = |H(\Delta U) - H(\Delta U')| \tag{7.33}$$

with ΔU and the $\Delta U'$ the utility increment vectors we provided at the beginning of section 7.1.3. In fact, if the dimension M of ΔU is finite and known, (7.33) is also finite and we propose to find its upper bound (7.34).

$$\sup_{\Delta U} |H(\Delta U) - H(\Delta U')| \tag{7.34}$$

Since the utility increment vector $\Delta U'$ is available to the decision maker (assessed under *Ceteris Paribus* as we mentioned above), we will start by solving (7.35).

$$\begin{aligned} &\sup_{\Delta U} |H(\Delta U')| \\ &\text{subject to} \qquad D_U(\Delta U, \Delta U') \leq \epsilon \end{aligned} \tag{7.35}$$

With a finite value of M, (7.35) is reduced to finding (7.36).

$$\max_{\substack{\Delta U \\ D_{\Delta U}(\Delta U, \Delta U') \leq \epsilon}} \qquad H(\Delta U) \tag{7.36}$$

Now we can think about the majorization of (7.36) and thus providing the solution ΔU_{meu}.
Let $\gamma = \sum_{i=1}^{L} \Delta u_i$. We can write ΔU as in (7.37).

$$\Delta U = (\Delta u_1, \ldots, \Delta u_M) \tag{7.37a}$$

$$\Delta U' = (\frac{\Delta u_1}{\gamma}, \ldots, \frac{\Delta u_L}{\gamma}) \tag{7.37b}$$

$$\Delta U'' = (\frac{\Delta u_{L+1}}{(1-\gamma)}, \ldots, \frac{\Delta u_M}{(1-\gamma)}) \tag{7.37c}$$

Therefore, $H(Q)$ can be written as in (7.38).

$$H(\Delta U) = h(\gamma) + \gamma * H(\Delta U') + (1-\gamma) * H(\Delta U'') \tag{7.38}$$

with h the binary entropy function (7.39).

$$h(x) = -x\log(x) - (1-x)\log(1-x) \tag{7.39}$$

Since $\#(\Delta U'') = (M-L)$, we can define an upper bound for $H(\Delta U'')$ [104], as in (7.40).

$$H(\Delta U'') \leq log(M-L) \tag{7.40}$$

(7.38) and (7.40) give (7.41)

$$\begin{aligned} \max_{\substack{\Delta U \\ D_{\Delta U}(\Delta U, \Delta U') \leq \epsilon}} \quad & H(\Delta U) \leq \\ & h(\gamma) + \gamma * H(\Delta U') + (1-\gamma) * \log(M-L) \end{aligned} \tag{7.41}$$

Therefore, the maximum value that could be reached by the entropy H is shown in (7.42).

$$H(\Delta U^*) = h(\gamma) + \gamma * H(\Delta U') + (1-\gamma) * \log(M-L) \tag{7.42}$$

The utility increment vector that achieves this maximum is ΔU_{meu} (7.43).

$$\Delta U_{meu} = \left(\Delta u_0, \ldots, \Delta u_L, \frac{(1-\gamma)}{(M-L)}, \ldots, \frac{(1-\gamma)}{(M-L)}\right) \tag{7.43}$$

The solution (7.43) is the optimal utility increment vector that ensures the ME with respect to the given information (7.37b). The specified preferences $\Delta U'$ could be characterized by a utility density function, while the rest of the vector ΔU_{meu} will be a uniformly distributed utility increment vector.

7.1.4 Realization of the Process

In this section we provide an example illustrating a situation of high uncertainty as well as the asymptotic behavior of *Shannon* entropy as to the feasibility of the solutions treated above.

In fact, we consider a symmetric one-dimensional random walk representing the evolution of the decision maker's preferences through time. We propose to study the way the sequence of utility increment vectors behaves asymptotically when the given number of states if countable infinite Let a discrete-time Markov chain with an infinite number of transient states, and where all the states communicate with each other, except for the absorbing state. This example corresponds to the *Gambler's ruin* problem, as represented in Figure 7.1.

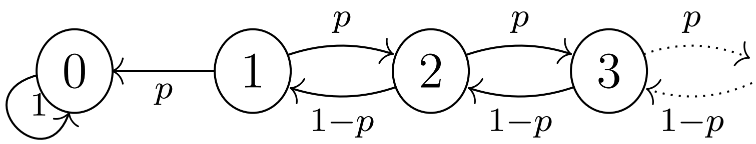


Figure 7.1: The Gambler's Ruin Random Walk

We consider the corresponding symmetric one-dimensional random walk, where $\Delta U_{(n)}$ represents the state of the system at time n, and where the state 0 is an absorbing state. Let $p = 1/2$ be the sequence's probability of taking a step to the right or to the left.

We start by finding the probability $p(\Delta U_{(n)} \neq 0)$, reflecting that the fact that system is somewhere on the sequence but not in the state 0. Assuming that $\Delta U_{(n)} \neq 0$ means that the system has never been in the state 0, therefore we can write (7.44).

$$\begin{aligned}
& p(\Delta U_{(n)} \neq 0) \\
&= p(\Delta U_{(n)} \neq 0, \Delta U_{(n-1)} \neq 0) \\
&= p(\Delta U_{(n)} \neq 0 | \Delta U_{(n-1)} \neq 0)\; p(\Delta U_{(n-1)} \neq 0) \\
&= p(\Delta U_{n} \neq 0 | \Delta U_{(n-1)} \neq 0)\; p(\Delta U_{(n-1)} \neq 0, \Delta U_{(n-2)} \neq 0) \\
&= p(\Delta U_{n} \neq 0 | \Delta U_{(n-1)} \neq 0)\; p(\Delta U_{(n-1)} \neq 0 | \Delta U_{(n-2)} \neq 0) \\
&\qquad \dots\; p(\Delta U_{1} \neq 0 | \Delta U_{0} \neq 0)\; p(\Delta U_{0} \neq 0) \\
&= \frac{1}{2^n}
\end{aligned} \tag{7.44}$$

Where for n=0 we have (7.45).

$$\Delta U_{(n)} = 1 \Rightarrow p(\Delta U_0 \neq 0) = 1 \tag{7.45}$$

From (7.45) we have (7.46).

$$p(\Delta U_{(n)} = 0) = 1 - p(\Delta U_{(n)} \neq 0) \tag{7.46a}$$
$$= 1 - \frac{1}{2^n} \tag{7.46b}$$

In order to evaluate the asymptotic behavior of the system and how it will reach the absorbing state we compute the limit (7.47).

$$\lim_{n\to\infty} p(\Delta U_{(n)} = 0) = 1 \tag{7.47}$$

Thus, we are certain that the system will converge to a deterministic distribution and therefore be absorbed in state 0 with a probability 1, written as in (7.48).

$$\Delta U_{(n)} \xrightarrow{U} \Delta U = \{1, 0\} \tag{7.48}$$

After finding the limit of the sequence $\Delta U_{(n)}$, we propose to check the behavior of the entropy in accordance to the behavior of $\Delta U_{(n)}$. By plotting the different values taken by $H\left(\Delta U_{(n)}\right)$ for different states' bounds, we obtain the plot in Figure 7.2. We can check that the approximation (7.49) holds,

$$P\left(\Delta U_{(4000)} = 0\right) \approx 1 \tag{7.49}$$

while the entropy $H\left(\Delta U_{(n=4000)}\right)$ does not end at $H(1)$. The entropy of the limiting distribution is (7.50).

$$H\left(\Delta U_{(4000)}\right) \approx 1.08817 \neq H(1) \tag{7.50}$$

In the case of a discrete-time Markov chain with countably infinite states, the uncertainty about the overall process (sequence of increment vectors) does not reflect the behavior of the limiting distribution.

7.1.5 Summary

We consider the maximum entropy principle for utility elicitation in the case of high uncertainty, *i.e.*, when the decision maker is facing a large number of outcomes while having a small number

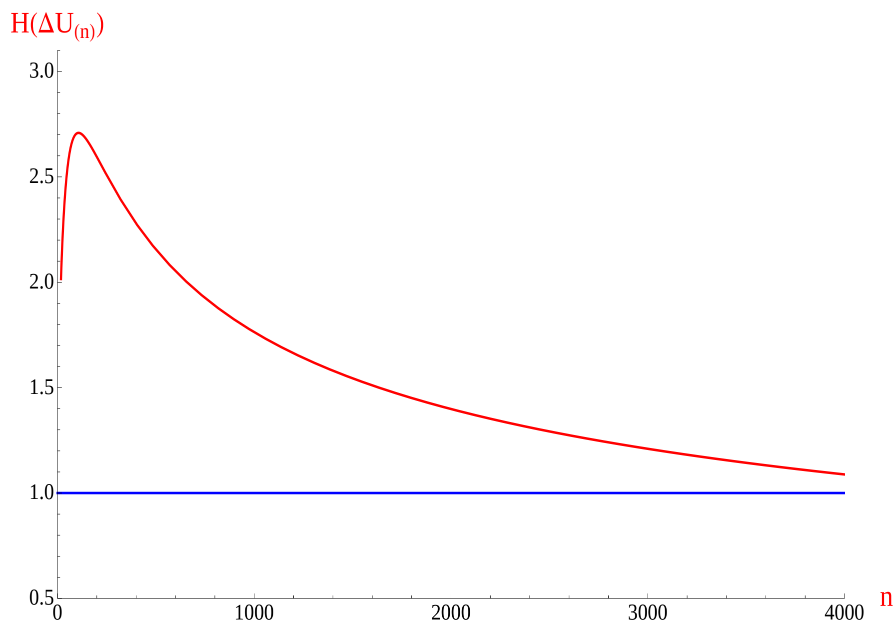


Figure 7.2: Entropy of the sequence $\Delta U_{(n)}$

of training sets. Despite its practical importance, as to understand how decisions are made by humans in bounded rationality, this problem has not been studied from the perspective of entropy maximization in the asymptotic case. To address this problem, we assumed that this situation of high uncertainty could be translated into a countably infinite number of prospects. The decision maker is asked to provide a related utility function that maximizes the entropy, given the available information. For instance, solving this type of problems could rely on a Lagrange multipliers method. But, like we have shown, this method, and general derivation-based methods could not be used due to the discontinuity of the *Shannon* entropy whenever the considered support is infinite. Therefore, we proposed another method based on finding a limiting least upper bound of the entropy, and thus giving a utility increment vector that maximizes it.

As an application, we give the example of a symmetric one-dimensional random walk on a discrete-time Markov chain representing the evolution of the decision maker's preferences throughout time. In this example, we show how the entropy of the limiting preferences' distribution is different from the overall entropy of the Markov chain.

As an important research issue to be further investigated, is the interpretation of this asymptotic behavior of the uncertainty in the case of a Markov process. In fact, we seek a relation between the increase of uncertainty that underlies the whole process and the uncertainty with regard to the final, limiting, preference distribution. Furthermore, we think about considering the case of multi-attribute utility increment vectors, and therefore generalize the univariate asymptotic case for the multivariate case. We also seek to characterize the MEU solution, in the case of a composed utility increment vector illustrating the case of multiple knowledge. In other words, finding a multivariate description of the different available utility distributions, for instance, as a *Dirichlet* utility increment distribution.

7.2 Complex Collaborative Design

7.2.1 Introduction

Over the past few decades, the Internet has experienced a significant expansion on multiple levels. Consequently, it has had a huge impact on economics and the Internet-based business models. Moreover, thanks to the decrease of transactions' costs, large communities of people have become connected, and therefore, many types of commercial relationships were created. Individuals have more access to advanced technologies due to the use of powerful softwares and performant computers.

At the same time, the new advances of the *Web 2.0* [17] gave the users more visual and interactive experiences by using Rich Internet Applications (RIA) [17, 1]. Several *electronic commerce* (e-commerce) business models were proposed for a wide range of applications and services [15].

Customer-to-Business (C2B) is a business model characterized by the inversion of roles known in the traditional business models [15]. In this model, costumers can develop and offer products or services to companies. Given the potential of an open and peer-to-peer environment, a C2B model can be efficiently implemented as a multi-agent framework. In this scheme, customers will interact with agents to develop their products, and the agents will negotiate the product with a given company.

On another level, collaboration and negotiation mechanisms have been studied widely and represent an exciting field of research [64, 66]. Especially, when the negotiation involves multiple and interdependent issues [34, 36, 30, 42] as it is the case of many real-world decision making problem. Several works have been done to deal with nonlinear utilities [53, 38, 35] and also to reveal possible utilities representations [30, 88, 14].

Most of real-word consensus making problems involve interdependent issues, which makes optimal contracts finding difficult. For instance, complex design is an example of domains involving a large number of issues governed by nonlinear relationships. Despite the multitude of models for multi-issue negotiation in nonlinear settings, none of the above works had proposed a concrete solution that could be deployed for a real-world problem. In fact, most of the proposed models are restricted to monetary domains that do not fully reflect the real challenges found in complex consensus making.

It is within this perspective that we propose our framework to show the possible interplay between the utilitarian nonlinearity found in the previous formulations and its analogous formulation found in complex design. It is our belief that it is possible to extend the constraint-based formulation of utility spaces into the constraint-based definition of the geometrical design space.

In the following, we will develop a collaboration architecture for products' design, according to a C2B scheme, and through the collaboration of an agent for the negotiation and the use of a 3D client application for the design of the product. At the same time, we will investigate possible representations of a specific type of contracts, representing a 3D object. We will also highlight the existing semantics between attributes and therefore establish possible assumptions related to general contracts modeling.

7.2.2 Collaborative Design System

The system is an online collaboration framework allowing users to design a given product (C2B scheme). Users can acquire the initial template of a model from a mediator and can use a 3D modeling IDE to propose their designs. The initial template is defined in terms of constraints, specified by the the mediator on the behalf of a company.

The end-user will have to specify his preferences by providing the design of the shapes and eventually some technical details of the model. Once the user has specified all his preferences, a bidding process will start and all the contracts related to the users' products will be sent to the mediator [34, 36].

An auction-like business logic will be deployed on the mediator. It will accept contracts from clients up to the final bidding time for the current product. Hence, a time limit must be specified for each design session. During this period, users will design their products and the agents might negotiate over the resulting contracts. Each bid from a user must be accompanied by a relevant product identifier which might be selected by the user to avoid the submission of an invalid code.

The design and bidding operations will be done through the collaboration of the human user and the agent. The user will have to design the product and the agent will negotiate the provided contract with the mediator. The mediator will synthesize all the received contracts into a unified group consensus. Then, an agreement might be made based on the assumption that the final contract will satisfy all the users' expectations.

A final contract ensures that the agents' choices hold and coincide with the constrained initial model of the product, *i.e.*, the template. After designing his product, each user will be expecting that hopefully the final contract will correspond to his design. The mediator will have to find the contract that represents all the individuals' preferences, therefore the best design of the product. In our context, an optimal contract will have to satisfy all the agents' bids and users' expectations.

More generally, the overall models of the agents will code the users' preferences just as if they are voting for the best product. For example, if the users are designing a specific type of toy, an elephant for instance. The mediator will try to maximize the expected number of designs (contracts) which represent elephants by finding the intersections between all the provided models and by maximizing the most common features or attributes.

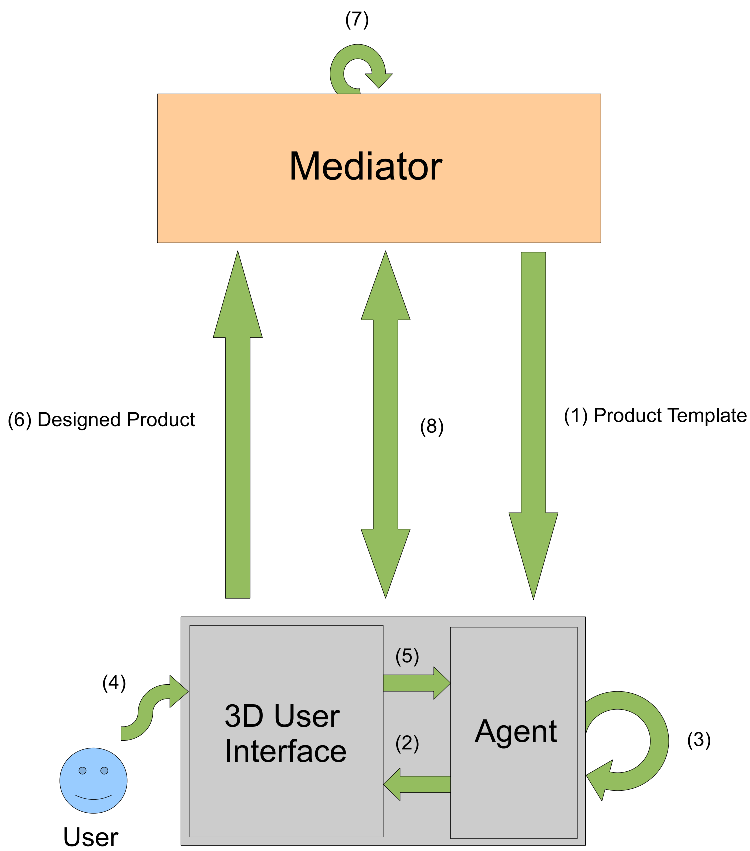


Figure 7.3: Architecture

7.2.3 System Architecture

Description

Our framework is based on a client/server model, where a client is represented by both the 3D user interface for the design and the automated agent. The server represents the mediator where we implement the business logic used for the consensus making. Several agents will be participating in the design process guided by their interface agents.

As indicated in Figure 7.3, the coupling of the UI and the agent was done to separate tasks during the Human-Agent interaction phase. The main purpose of the UI is the acquisition of the user's preferences (4) through means of 3D shapes modifications. Initially, the agent is used for the acquisition of the product template provided by the mediator (1). The template will be issued to the UI from the agent (2), allowing the user to design his product according to his viewpoint (4). The agent will assist the user by allowing him to specify a parametrized description of the templates. And based on this parametrization, the agent will recursively generate the structure of the model.

Once the new product is available (5), the agent will prepare a contract (3) specifying the product attributes and will send it back to the mediator (6). Eventually, the agent can generate several contracts and start a negotiation with multiple bids (8) [53, 78, 65].

The mediator will collect all the contracts issued from the clients and try to find a combination of bids that maximizes social welfare (7).

In the case where the initial constraints are not satisfied by all the agents, a back-and-forth process might be used (8). In addition, the mediator will have to change slightly the initial template by modifying the constraints' bounds in the way that eases the design for the agents (users). In this case, all the submitted contracts will converge to a unique contract which satisfies both of the mediator constraints and the clients' preferences.

Since the agents are negotiating over contracts based on an initial model (template), there will be no possible conflicts between the agents on the designed product: all the chosen values for each attribute will fall within the constrained shapes specified initially by the mediator in the template.

This whole negotiation process will yield a better design of the product at each iteration. Once the time limit is reached, the final contract will be adopted by the mediator and sent to the users. The time limit must ensure enough time for the clients to design their products, and also for the agents to negotiate the contracts. In this way, the negotiation process will converge and yield the optimal contract.

Contract Construction

In our work, we focus on a specific type of contracts, representing a product as a 3D model. As we will show later, these types of contracts share common characteristics which make them different from other studied contracts [96, 38, 14], in the way attributes are represented and connected [38, 88, 14].

Therefore, we introduce the concept of *graphical* and *non-graphical* attributes. A graphical attribute corresponds to a variable which can be modeled and manipulated by the user in a three dimensional space. A 3D shape will be represented as a collection of graphical attributes, including points, colors and so forth. A non-graphical attribute is related to an abstract type of attribute, that cannot be represented in 3D space. For example, the type of an engine, the class of a fuel, the category of a metal, the price of a chair, are non-graphical attributes.

In our modeling, we adopt the X3D standard [17, 1] as a framework to represent our designed products. In addition, all our graphical objects will be defined using the *IndexedFaceSet* (IFS) geometry node [1]. This component allows the encapsulation of a set of attributes in the same entity, which will be identified as a group. A group will contain a number of interdependent attributes describing the same entity and sharing the same space.

As we can see in Figure 7.4, a graphical attribute v_i can be represented by a vertex $v_i(x_i, y_i, z_i)$ in a 3D space. However, it might also be a color, an opacity indicator or any other graphical element [1].

A group of graphical attributes is an IFS composed of a finite number of vertices, connected under the assumption that these vertices will describe a compact and unique simplex. Ultimately, the whole designed product will be a collection of groups, assembled to form the global 3D product. This structure can be generalized using a multilayered structure with a recursive definition.

After designing the product *i.e.* the design of graphical and non graphical attributes, the final contract will be defined as a collection of the whole attributes grouped according to a certain logic. Therefore, a contract $\vec{s}$ containing *n* groups, can be represented as in (7.51).

$$\vec{s} = (\ G_1, \ldots, G_i, \ldots, G_n\ ) \tag{7.51}$$

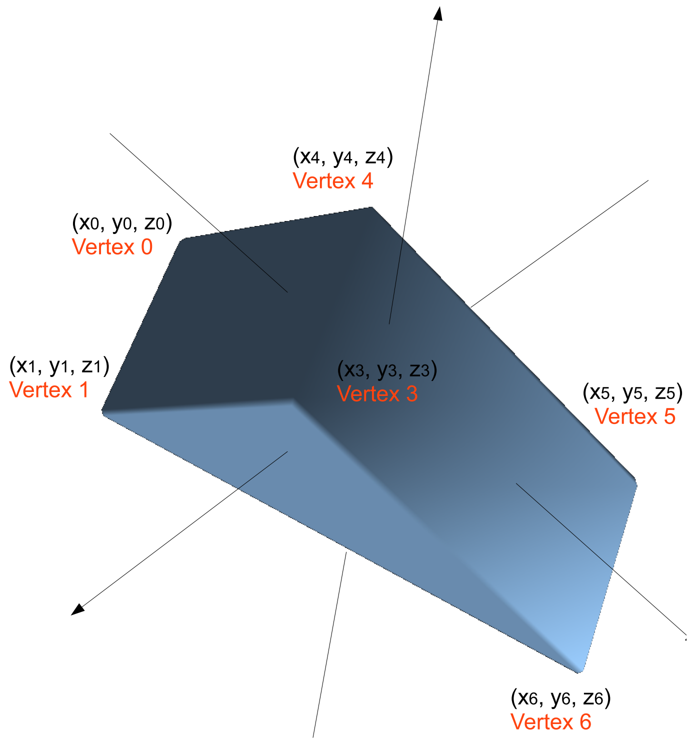


Figure 7.4: An IFS representation

where every group G_j is a collection of attributes:

$$G_i = (\overbrace{\underbrace{v_{i,1},\ v_{i,2},\ \ldots, v_{i,n_i}}_{Vertices},\ c_i,\ s_i, \ldots,}^{Graphical\ attributes}\ \overbrace{a_{i,1},\ a_{i,2},\ \ldots,\ a_{i,m_i}}^{Non-graphical\ attributes}\ ) \tag{7.52}$$

A group G_i has n_i vertices and m_i non graphical attributes. Other graphical attributes can be added to the group, for example, the color c_i and the shininess s_i.

Groups Decomposition

As we have stated above, the negotiation process starts when the template of the product is sent to the client. Then, the user will have to modify it according to his conception under the guidance of the agent which proposes ways of generating the new transformations. In fact, the agent will recursively alter the initial template given the parametrization chosen by the user.

Initially, the template is represented as an XML document including the X3D graphical representation of the product as well as additional information. In other words, the template is a simple, minimalistic, unfinished representation of the product. The XML representation of the product must be converted into an appropriate format. For this purpose, we provide an algorithm that disassembles the XML representation of the product into more atomic components, in order to be manipulated by the user through the UI. The product is represented by an object which can be described by the relation (7.53).

$$\begin{aligned} Object &= \bigcup_{j=1}^{m} IFS_j \\ &= \bigcup_{j=1}^{m}\bigcup_{i=1}^{n} v_i^j(x_i^j, y_i^j, z_i^j) \end{aligned} \tag{7.53}$$

Initially, the algorithm will parse the template and extract all the m groups *i.e.* the IFSs. During the design of the product, the user will alter the attributes of the product according to his preferences. In parallel, the new configuration of the object is computed and updated dynamically.

As we can see in the procedure *Desing_shape*, the input is the X3D template to change (Object), and the output is the new product. The whole process is executed on 4 matrices T, F, E and R. The process is described as following.

- $T_{n,3}$: is build based on the product template. It contains all the attributes of the product (the graphical and non graphical).

- $F_{m,3}$: is built based on **T** by removing redundant vertices. Those redundancies are due to the representation of the faces of an object by vertices' indexing [1].

- $E_{m,3}$: contains the modifications made by the user for m vertices. The ith entry in **E**, corresponds to the new coordinate of the vertex v_i.

- $R_{n,3}$: the resulting matrix, containing the newly designed product.

- **procedure** Design_shape($Object$)

Extract all the IFS_j from Object.

for each $IFS_i \in Object$ **do**

Extract vertexes $v_i(x_i, y_i, z_i) \in IFS_i$

$T_{n,3} := \bigcup_{i=1}^{n} v_i(x_i, y_i, z_i)$

$F_{m,3} := Remove\_Redundant\_Vertexes(T_{n,3})$

$E := F$ // Initialization.

for each modification $e_i(x_i^e, y_i^e, z_i^e)$ on $v_i(x_i, y_i, z_i)$ **do**

E[i] := E[i] - $e_i(x_i^e, y_i^e, z_i^e)$ // New Positon.

$R_{n,3}$:= f(T, F, E) // New Object.

View the new Object in the UI.

return $R_{n,3}$.

7.2.4 Consensus Making

Some assumptions regarding the way the mediator operates to find the optimal contract must be introduced. In our adopted model [34, 36, 33], neither the agents nor the users do share their preferences with other designers.

It is also important to mention that the way contracts are represented is governed by a constraint-based viewpoint that is coherent with the usage of the X3D representation (IFSs).

We note that in our model the design space is considered according to two different viewpoints. One viewpoint is static, purely geometrical, defined as the 3D design space of the user. In other words, it is the visible configuration or choice space that the user is perceiving.

The second viewpoint is dynamical, and is governed by the user's preferences that may change during the consensus making process. In fact, a chosen shape or vertex will be assigned a specific utility value. Even if the user is unaware of these assignments, it is important to keep in mind that the agent will have to perform some preferences elicitation during the interaction with both the user and the mediator. Hence the necessity to assign evaluations to the different possibilities that are given to the user.

The design space is defined in terms of constraints, making the utility space a constraint-based utility space. That is, all the models are built based on an aggregation of shapes according to the cube constraints $c_{k\in[1,m]}$ defined in the mediator's template as sub-regions of the total $\mathbb{R}^3$ design space. We say that the cube c_k has value $w(c_k, \vec{s})$ for contract $\vec{s}$ iff c_k is satisfied by $\vec{s}$. That is, when the contract point $\vec{s}$ falls within the sub-space of c_k. This is the case where the user's chosen shape falls in the bounds of the cubic constraint defined by the mediator. The utilitarian evaluation of a possible design $\vec{s}$ is thus defined as in (7.54).

$$u_i(\vec{s}) = \sum_{c_{k\in[1,m]},\ \vec{s}\in c_k} w(c_k, \vec{s}) \tag{7.54}$$

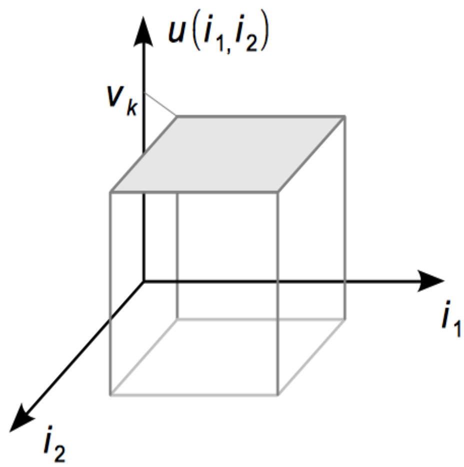


Figure 7.5: Cubic constraint

The mediator will gather all the submitted bids from the agents, and search for the optimal contract(s). In our architecture, maximizing the negotiation outcome is finding the contract $\vec{s^*}$ that corresponds the most to the clients' expectations. Therefore, the objective function can be described as in (7.55).

$$\vec{s^*} \ = \ \arg\max_{\vec{s}} \sum_{i\in N} u_i(\vec{s}) \tag{7.55}$$

The used protocol tries to find the contract $\vec{s^*}$ that maximizes social welfare, *i.e.*, the total utilities for all agents [53, 38]. In our model, utilities represent the users' preferences which are specified through the design process. The utilities specified by a client as a set of preferences will also represent the same utilities u_i of the agent.

In case the agents are proposing bundles of models instead of one single contract, we will be using the approach [53] illustrated in *search_solution*. That is, the mediator starts by finding all the combinations of bids, one from each agent that are mutually consistent.

Ag: A set of agents

B: A set of Bid-set of each agent ($B = \{B_0, B_1, \ldots, B_N\}$)

The set of bids from agent i is $B_i = \{b_{i,0}, b_{i,1}, \ldots, b_{i,m}\}$

procedure search_solution(B)

$SC := \bigcup_{j \in B_0} \{b_{0,j}\}$

$i := 1$

while $i < |Ag|$ **do**

$SC' := \emptyset$

for each $s \in SC$ **do**

for each $b_{i,j} \in B_i$ **do**

$s' := s \cup b_{i,j}$

if s' is consistent

then $SC' := SC' \cup s'$

$SC := SC'$, $i := i + 1$

$maxSolution$ = getMaxSolution(SC)

return $maxSolution$

The resulting bids represent the contract that satisfies the constraints set by the mediator for designs' election. The mediator employs an averaging method that maximizes social welfare. The averaging method has the ability to cancel conflicting preferences and provide a centered estimation of the group preferences.

7.2.5 Experimental Results

The implemented architecture is composed of the agent and the mediator. Herein, the agent is nothing more then a Python-coded plugin deployed under the Blender 3D environment. Figure 7.7 shows the main interface of Blender as well as the user interface of the agent (red square). Once Blender is launched, the agent interface will be visible in the Tools panel. The user will firstly connect to the mediator in order to load the template and launch the agent. When started, the agent will start generating the proposed parametric changes while acknowledging the user's modifications of the 3D model.

The adopted parametric design is described by a number of constraints or initial parameters specified by the mediator through the template. For instance, in Figure 7.7, we are designing the structure of a four-legged animal based on multiple parameterizations of the model's parts: the body, the 4 legs, the feet, the neck, the forehead and the head. Each user will have to pick specific values for these parameters, and the agent will start building the structure of the model recursively through means of geometric *extrusion*.

The parametrized recursive extrusion performed in Figure 7.6 are performed as following.

1. The agent starts by creating an initial cube, the *seed*, corresponding to an example of initial constraint. An example of such constraints is represented in Figure 7.5 and relating to a two-dimensional attribute-space with a particular utility value v_k.

2. Recursively, and depending on the user's specified parameters for the body, the agent will extrude the cube into a line of cubes. Then, the line will be extruded again into a plane of cubes, and the same plane into a bigger structure until forming the main body of the animal.

3. The agent will then pick four distinct cubes located under the structure and perform an extrusion towards the bottom. These extruded lines will form the legs. Similarly, the feet will be extruded by taking the faces of the bottom of each leg. All these operations are governed by the dimensions specified by the user.

4. The neck of the structure will be extruded by picking a cube on the top of the body. The head and the forehead are generated in a similar manner based on the parameters specified by the user.

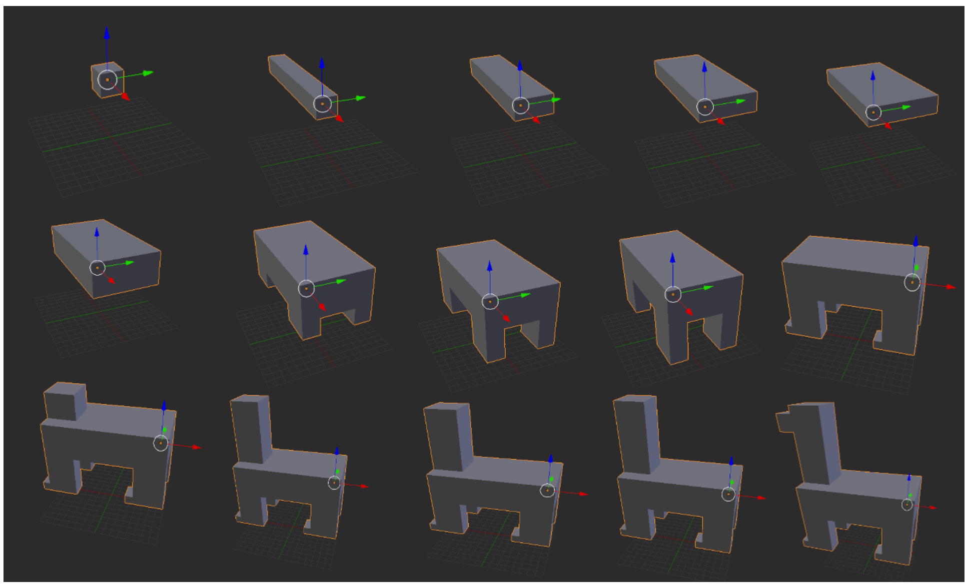


Figure 7.6: Parametrized Extrusion

It is important to highlight that such choice of parametrized generative design gives more autonomy to the interface agent through the recursive generation. Indeed, the recursion and the parametrization will generate an initial structure that the user can elaborate upon until finding his ideal design. Additionally, this generative method makes the design process easier for an inexperienced user, especially in the presence of complex 3D design environment like Blender. The way the design is parametrized will basically relate the user's preferences. Since There are many agents involved in the design process, we should find the right mechanism yielding the final col-

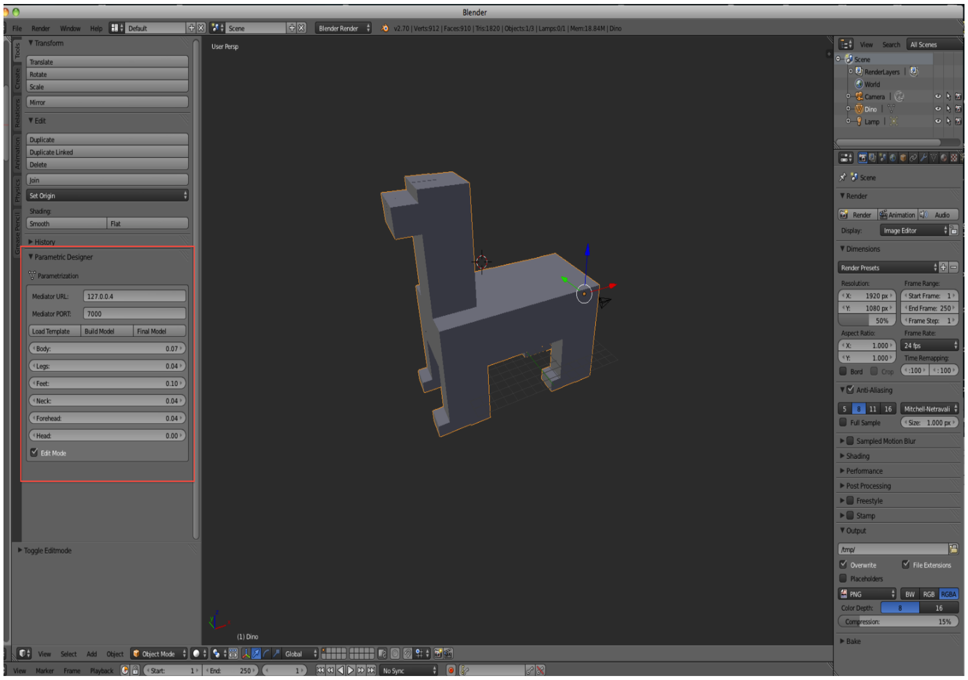


Figure 7.7: Interface Agent under Blender

lective design. Satisfying both the mediator constraints as well as the conflicting user-preferences is the objective.

7.2.6 Discussion

Adopted Formalism

Besides the implementation of the collaborative architecture for 3D design, our work revealed a number of points which might be investigated and generalized. In fact, we focus on a specific type of contracts, where we design a product by providing all its attributes and also by specifying the inter-dependencies between them. What makes this design easy compared to contracts characterized with nonlinear utilities [53, 78, 65], is the fact that the attributes are well structured and assembled according to a certain logic. This logic is defined by the geometrical topology of the manipulated attributes as part of a 3D object.

The interdependency relations can be understood while modifying a vertex in a group. Changing the coordinates of a vertex will consequently alter the configuration of the surrounding vertices of the same group. Altering a vertex $v_{i,j}$ from a group G_j will not affect any vertex $v_{i,k}$ from another group G_k. We say that $v_{i,j}$ and $v_{i,k}$ are independent. More generally, for any contract, dependencies between attributes could be represented in the same way, and consequently, decomposed into groups under a certain hypothesis. The decomposition hypothesis relate to the type of the problem in hand as well as the objectives. For instance, a negotiation problem concerned with monetary prospect will be decomposed based on how partial components of the problem relate to each other internally as to maximize an internal utility function. In our model, the hypothesis were defined by the geometry of the 3D shapes we are manipulating. Plus, the nature of the utility space being explored, which can be described as the union of two subspaces : a three dimensional space for graphical attributes (3D design), and another subspace for non graphical attributes. After considering the nature of the manipulated contracts, we can assume that, unlike the other ways of grouping attributes into clusters [88, 14], this representation retraces more the hierarchical and graphical coupling and cohesion between attributes. As long as the encapsulation between groups is respected, the same way of composition for 3D objects can be adopted to decompose multi-attributed contracts into groups, or eventually recursive groups. For instance, the IFS formalism could be extended to other domains, only by specifying the issues and the constraints that govern them. In the same way and IFS is a constrained collection of graphical attributes. Graphical modeling of issues is a realistic way to understand and analyze attributes' dependencies, by positioning our product in the adequate utility space. In this way, constraints could be identified easily and optimal contracts could be designed. We assume that the separation between the graphical 3D structure of a product and the non graphical features is more intuitive than manipulating attributes without taking into consideration the possible correlations between them.

Consensus Model

Once the user has finished his design, the agent will submit the produced model to the mediator. After collecting the models as IFS collections, the mediator will perform an averaging over all the vertices forming the faces of the models. The goal is to find the locations that describe the general tendency of the agents' design as well as attempt to discard bad designs by giving the priority to uniform distributions amongst the clouds of points. For instance, Figure 7.8 provide an example of designs taken amongst 6 agents. The advantage of the adopted parametrized method is that it allows an extrusion diversity. By adopting the averaging mechanism mentioned above,

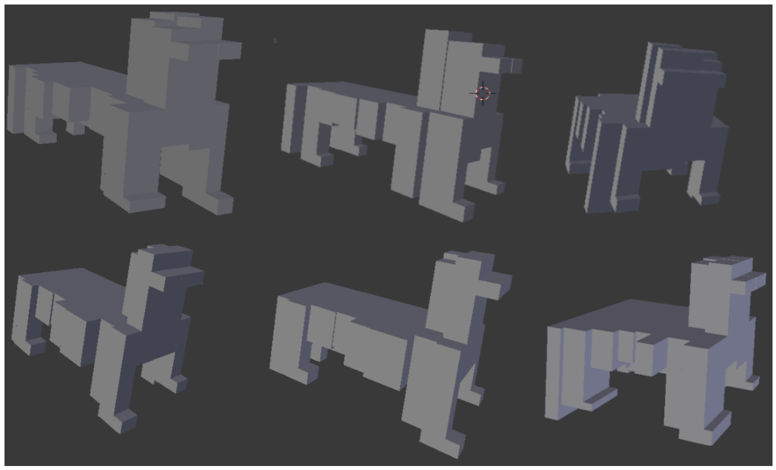


Figure 7.8: Submitted Models

the resulting model for Figure 7.8 is given in Figure 7.9. It is important to highlight the advantage of the averaging in its ability to smoothen the rugged faces of the IFSs by providing aligned faces' edges.

The collective behavior of the users, coupled with the recursive extrusion-based assistance of the agents is reflected in the final model. This aggregation of the wisdom of the users [95] was allowed given the geometrical nature of the problem in hand as well as the usage of recursion in the way models are generated. The recursive and parametrized way of generating the models provides a compact designing method and a way of simulating distinct human users' behaviors.

7.2.7 Conclusion

We developed an agent-based architecture for 3D graphical design of products throughout a collaborative process. The negotiated contracts are different from the previously studied ones, in the

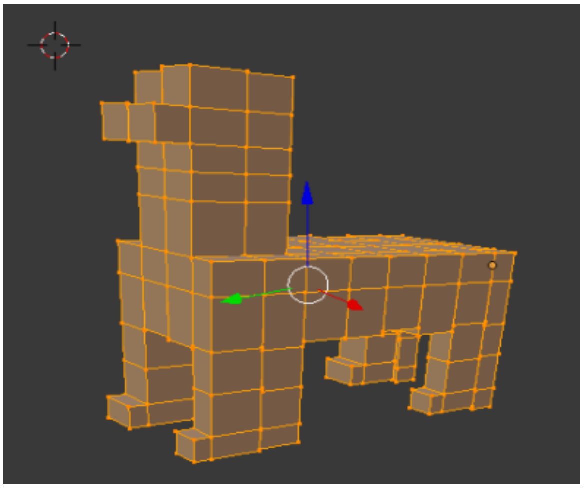


Figure 7.9: Consensus Model

way attributes are classified. In fact, in our case a contract represents a 3D object, designed by the user as to reflect his preferences. The whole process can be described as an agent-user interaction as well as a users-users collaboration. The agent-user interaction takes advantage from a parametrized and recursive generation method used to guide the user in his design.

Furthermore, we have investigated the possible representations of graphical contracts by studying the existing relationships between attributes, and therefore established possible assumptions related to general contracts modeling.

The next step will be an improvement of the constraints' management methods for non graphical attributes by allowing monetary attributes to be involved in the assessment process.

Chapter 8

Conclusion and Future Works

The goal of this thesis was to contribute to the area of decision making with complex domains, and particularly, when modeling preferences. The contributions of the work as well as the future directions are summarized the next sections.

8.1 Summary and Contributions

1. **Quantifying cognitive complexity using information entropy.** (Chapter 3)

   We propose to bridge the gap between complexity as perceived in cognitive sciences, *i.e.*, as a cognitive load in cortical areas [10], usually occurring during high-information load tasks [16]; and complexity as it is perceived in decision making, occurring under endogenous bounded rationality [21].

   We start by introducing a general model whereby cognition is performed on an abstract graphical representation that could be instantiated as a Neural Network, Bayesian network, Markov Random Field, CP-Net, Qualitative Probabilistic Network [101], and so forth. Then, we take cognition to be a graphical game of complexity between nature and an agent. Complexity is assessed based on the search cost in the problem structure and its underlying graphical topology. Particularly, we use the information theoretical notion of entropy as to evaluate the degree of spread of the concepts' degrees, reflecting the concepts (nodes, attributes, or issues) importance to the task in hand. This gives a method to assess the complexity of an exploration algorithm and its computational cost when used in a real decision making problem.

2. **Complexity, entropy and uncertainty in high-information load tasks.** (Chapter 3)

   We are interested in studying the evolution of cognitive processes with high-information load. This is done by drawing the interplay between the structural complexity of the cognitive graph of the decision problem, its entropy, and how the uncertainty of the whole process evolves.

   We show that the larger the cognitive space is, and the more uncertain the evolution of the process can be, despite the certainty of the final outcome. This is in fact due to the discontinuity of entropy whenever the cognitive graph has a countably infinite number of concepts. This counterintuitive situation recalls the problem of *infinite regress* as well as the *frame problem* in the sense that we incorporated the concept of *infinity* while reasoning about complex cognition.

   The entropy discontinuity could in fact account for the divergence or dissimilarity of belief within one individual (probabilistic) cognition space; in the way that can cause some kind of "collective wisdom" within one individual. This corroborates the idea that the temporal separation of guesses increases the benefit of within-person averaging by increasing the independence of guesses, thus making a second guess from the same person is more like a guess from a completely different person [99].

3. **Utility hyper-graphs for complex preferences representation.** (Chapter 4)

   We propose a novel representation for nonlinear utility spaces as to tackle the complexity problem. It has the merit of being modular and parametric, which allows search strategies evaluation as well as any graph-theoretic analysis. We also propose efficient optimization algorithms and heuristics for optimal contracts search based on message passing in the utility hyper-graph. This is performed by a *loopy utility propagation* scheme inspired by the loopy belief propagation used in probabilistic graphical models [87]. The proposed approach outperforms the other sampling-based methods and provides a better scaling for both issues and agents.

4. **Quantifying the complexity of nonlinear utility spaces.** (Chapter 5)

   We propose an efficient method to assess the complexity of a nonlinear utility space using its induced utility hyper-graph. This is based on the assumption that the hyper-edges of a utility hyper-graph are an instantiation of the abstract cognitive graph used in 1, and that it is possible to use the same entropy-based approach [47] as to measure complexity. This is done by assessing the entropy of the constraint-issue distribution because it reflects the degree of interdependence within the utility space. This has the merit of reflecting the computational complexity of any search algorithm, and corroborates other issue-related interdependency measures [40].

5. **Low-complexity search in utility hyper-graphs.** (Chapter 5)

   Based on the complexity measure, we provide several search strategies and identify the optimal strategy that minimizes the search cost. The optimal search strategy allows a low-complexity traversal of the utility hyper-graph while preserving the contracts optimality. This is an improvement of the message passing algorithm and an example of fast and frugal heuristics used in decision making under high-information load [16, 44, 45]. The optimal strategy uses a Power-law exploration topology, and corroborates the ubiquity of such distributions in complex systems undergoing phase transition. In our case, the phase transition is with respect to the accumulation of utility, or welfare.

6. **Using utility hyper-graphs in multilateral negotiation.** (Chapter 6)

   We provide an evaluation of the hyper-graphic representation in a multilateral mediated negotiation setting. To this end, we assume that the agents utility models are defined as utility hyper-graphs, and that they use our message passing mechanism when searching for their optimal contracts. Agents submit their proposals (bids, or contracts) to a mediator who tries to find the contract(s) that maximize social welfare. We show that under high complexity, the collective social welfare could be greater than the sum of the individual expected best utilities. This corroborates the assumption of nonlinearity by reflecting the idea that the nonlinearity of the (individual) agent's utility spaces is propagated from the issue-constraint level up to the agreement level through means of mediation. It is possible to propose a principled method that could exploit this nonlinearity property as to maximize social welfare in consensus-based platforms.

7. **Unifying Constraint-based utility functions.** (Chapter 6)

   We propose a formula that unifies all known constraint-based (Cubic, Bell, Conic, etc.) utility functions. The new representation leads us to a potential parametric model that could be used for opponent modeling in complex nonlinear negotiations.

   With this representation, the constraints' definitions collapse to fewer parameters that could potentially be defined using prior distributions and the underlying hyper-parameters. These parameters could be estimated online, using well known Machine Learning techniques during a real-time encounter between two agents.

8. **Asymptotic maximum entropy utility principle.** (Chapter 7)

   We propose a Maximum Entropy principle for preferences elicitation for utility functions with an infinite number of outcomes. This is the mirrored version of 2., in the sense that we study the evolution of preferences defined over an infinite number of outcomes. The new principle extends the maximum entropy principle for utility elicitation [2].

   We show that under high-information load, a stochastic preferences merging process is non-ergodic [58] to the extent that its entropy is discontinuous. This is another way of interpreting the merging process as a dynamic system. In this case, there are states of the system that cannot be occupied again when the system reaches the limiting distribution (or limiting preference profile), perceived as the equilibrium state. This state is known to be transient and irreversible, which makes the whole process non-ergodic with regard to uncertainty. Trying to analyze the whole system by taking a momentary snapshot at each time step does not reflect the behavior of the whole system on the long run. This independence between the states of the system could also account for [99].

8.2 Future Directions

As future directions, we are interested in the following areas of research:

1. **Hierarchical Automated Negotiation**: It is possible to exploit the structure of the utility hyper-graphs by proposing an hierarchical exploration scheme and evaluate it in a hierarchical negotiation scenario. Indeed, it is possible to recursively negotiate over the different layers of the problem according to a Top-down approach. Even the idea of issue could be abstracted to include an encapsulation of sub-issues, located in sub-utility spaces and represented by cliques in the hyper-graph. Consequently, search processes can help identify optimal contracts for improvement at each level. This combination of separating the system into layers, then using utility propagation to focus attention and search within a constrained region can be very powerful when the agent is making offers (bidding) during a short number of rounds. Additionally, it addresses general negotiations with multiple domains in the sense that the different components of the hierarchy could account for different knowledge-domains or ontologies. This is inherent to realistic negotiation scenarios such as the collaborative design of complex contracts.

2. **Divide-and-Conquer in Complex Automated Negotiation**: Analyze the interdependence arising between the issues (or between the constraints, as explained in section 4.2.3) as to assess their importance and influence on the contracts optimality. Being able to assess the issues importance could in fact be used to simplify complex negotiation scenarios by focusing on the most important sub-contracts. This allows the whole problem to be divided into independent components that could be separately negotiated by the agents. For instance, let us assume that two agents 1 and 2 have utility hyper-graphs G_1 and G_2 with K isomorphic components. In this case, it is possible to run K negotiation threads that can ensure a better agreement then if the agent had to negotiate over the whole representation. Having isomorphic components might not be realistic in the sense that, usually, real-world problems have one dense component. However, we could establish some assumptions that could help relax the general constrained form and allow independent components to emerge. This could

be done for instance by using the importance of the issues as a way to go beyond the initial graphical connectivity configuration.

3. **Opponent Modeling and Preferences Learning using GGMM**: The proposed GGMM approximation of constraint-based utility functions could be used for preferences learning as to allow the agent to learn and build the structure of her own utility space whenever this information is missing. This assumption is due to the bounded rational nature of the agent and the limited foresight and access to the whole structure of the problem.

   This could be done by building the induced dependence graph from the utility hyper-graph in the same way a Bayesian Network relates to its Factor Graph representation. Herein, two different dependence graphs could be built, between the constraints as well as between the issues (section 4.2.3). Several Machine Learning technique could then be used for Opponent Modeling. Additionally, this could allow the agent to learn her own preferences or the preferences of a human user in an interactive manner. Interacting with a human user as to elicit her preferences is important whenever the domains are large and require a human guidance in the way the utility models are built.

4. **Holonic Mechanism Design**: It is possible to generalize the message passing mechanism in utility hyper-graphs from the issue-constraint level to several levels of abstraction. Particularly, and as shown in Figure 8.1, an issue k could act as an agent k, having different valuations defined as bids $\{b_{k,1}, \ldots, b_{k,n_k}\}$ in the same way issue k has different valuations from a particular domain. Similarly, a constraint acts as a coalition defined by a mediator M with a consensus function f. This stems from the fact that our message passing scheme is in fact a generalized form of the Vickrey–Clarke–Groves (VCG) mechanism [54].

   Particularly, fixing a value for one issue while dynamically evaluating the other issues is similar to the Clark pivot mechanism used in VCG. The same applies for the payment scheme, whereby $\mu_{M \to k}$ is a payment sent from the mediator M to agent k, for k's bid b_k.

   It is possible to prove that the mechanism is efficient and Incentive Compatible (IC) all over the hierarchies of the architecture. The resulting architecture is a holonic multi-agent system for general purpose consensus making. This holonic architecture is a generalization of the hierarchical scheme described in 1.

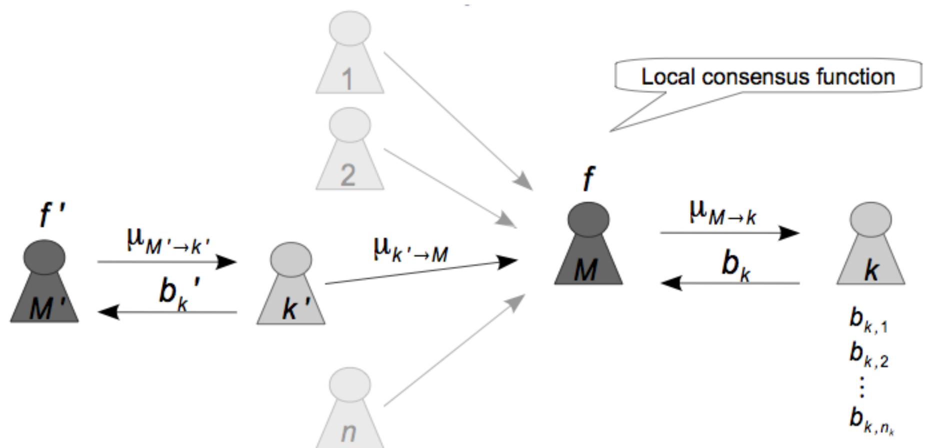


Figure 8.1: VCG as Message Passing in Hierarchical Utility Hypergraph

Appendix A

Appendix

A.1 Proof of (7.30)

$$H(\Delta U_{(n)}^{(a,b)}) = \underbrace{-\left[1-\left(\frac{\log a}{\log n}\right)^b\right]\log\left[1-\left(\frac{\log a}{\log n}\right)^b\right]}_{T_1} \tag{A.1}$$

$$\underbrace{-n\left[\frac{1}{n}\left(\frac{\log a}{\log n}\right)^b\right]\log\left[\frac{1}{n}\left(\frac{\log a}{\log n}\right)^b\right]}_{T_2} \tag{A.2}$$

We have $\lim_{n\to\infty} T_1 = -1 \times \log(1) = 0$. T_2 can be written as in (A.3).

$$\begin{aligned} T_2 &= -\left(\frac{\log a}{\log n}\right)^b\left[\log\left(\frac{1}{n}\right) + \log\left(\left(\frac{\log a}{\log n}\right)^b\right)\right] \\ &= \underbrace{\left(\frac{\log a}{\log n}\right)^b \log(n)}_{T_3} + \underbrace{\left(\frac{\log n}{\log a}\right)^b \log\left(\left(\frac{\log a}{\log n}\right)^b\right)}_{T_4} \end{aligned} \tag{A.3}$$

Using *L'Hôpital*'s rule, we can show that $\lim_{n\to\infty} T_4 = 0$, while for T_3, we have (A.4).

$$\begin{aligned} &\lim_{n\to\infty}(\log(a))^b \times \frac{log(n)}{(log(n))^b} = \\ &\lim_{n\to\infty}(\log(a)^b) \times (log(n))^{1-b} \end{aligned} \tag{A.4}$$

Hence

$$\lim_{n\to\infty} H(\Delta U_{(n)}^{(a,b)}) = \begin{cases} 0 & \text{if } b > 1 \\ \log a & \text{if } b = 1 \\ \infty & \text{if } 0 < b < 1 \end{cases}$$

A.2 Proof of Theorem 1

We start by considering the sequences (A.5).

$$\begin{aligned} Z &= \{z_i\}_i \\ Z_1 &= \{z'_i\}_i \\ Z_2 &= \{z''_i\}_i \end{aligned} \tag{A.5}$$

Let $z'_{a_-}, z'_{a_+} \in Z_1$ and $z''_{a_-}, z''_{a_+} \in Z_2$ relative to the nodes $a_+, a_- \in G$. Since we are comparing Z_1 and Z_2 after the addition of the edges (s, a_-) and (s, a_+) respectively to G_1 and G_2, only the values of $z'_{a_-}, z'_{a_+}, z''_{a_-}, z''_{a_+}$ are to be considered, while the other values are unchanged. Furthermore, we have $z'_{a_-} = z''_{a_-} + 1$ and $z''_{a_+} = z'_{a_+} + 1$.

Finding the total difference between the entropies of Z_1 and Z_2 is reduced to finding the difference between the entropies of $\{z'_{a_-}, z'_{a_+}\}$ and $\{z''a_-, z''_{a_+}\}$ as in (A.6).

$$\begin{aligned} & H(Z_1) - H(Z_2) = \\ & \quad = H(\{z'_{a_-}, z'_{a_+}\}) - H(\{z''_{a_-}, z''_{a_+}\}) \\ & \quad = H(\{z''_{a_-} + 1, z'_{a_+}\}) - H(\{z''_{a_-}, z'_{a_+} + 1\}) \\ & \quad = -(z''_{a_-} + 1)\log(z''_{a_-} + 1) - z'_{a_+}\log(z'_{a_+}) + z''_{a_-}\log(z''_{a_-}) + (z'_{a_+} + 1)\log(z'_{a_+} + 1) \\ & \quad = \left[(z'_{a_+} + 1)\log(z'_{a_+} + 1) - z'_{a_+}\log(z'_{a_+})\right] - \left[(z''_{a_-} + 1)\log(z''_{a_-} + 1) - z''_{a_-}\log(z''_{a_-})\right] \\ & \quad = h(z'_{a_+}) - h(z''_{a_-}) \end{aligned} \tag{A.6}$$

Let the function h be defined as in (A.7) :

$$h(x) = (x + 1)\log(x + 1) - x\log(x) \tag{A.7}$$

We have :

1. h is strictly monotonic: $h'(x) > 0 \;\; \forall x \in [1, +\infty]$
2. $z'_{a_-} < z'_{a_+}$ and $z'_{a_-} > z''_{a_-}$ (since $z'_{a_-} = z''_{a_-} + 1$).
   Hence $z'_{a_+} > z''_{a_-}$

3. From 1. and 2., we have $h(z'_{a_+}) - h(z''_{a_-}) > 0$.

   Hence $H(Z_1) > H(Z_2)$

That is, adding a link from s towards a_+ (rather than a_-) minimizes the entropy (A.8).

$$H(Z|s \sim a_-) > H(Z|s \sim a_+) \tag{A.8}$$

References

[1] The xj3d project, http://www.xj3d.org.

[2] Ali E. Abbas. Maximum entropy utility. *Operations Research*, 54(2):277–290, 2006.

[3] John Eric Adair. *Decision Making and Problem Solving Strategies*. Creating success. Kogan Page, 2007.

[4] K.J. Arrow. *Social Choice and Individual Values*. Cowles Foundation Monographs Series. Yale University Press, 1963.

[5] Haris Aziz. Multiagent systems: algorithmic, game-theoretic, and logical foundations by y. shoham and k. leyton-brown cambridge university press, 2008. *SIGACT News*, 41(1):34–37, 2010.

[6] Fahiem Bacchus and Adam Grove. Graphical models for preference and utility. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pages 3–10, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[7] MiguelA. Ballester and Guillaume Haeringer. A characterization of the single-peaked domain. *Social Choice and Welfare*, 36(2):305–322, 2011.

[8] L W Barsalou. Perceptual symbol systems. *Behavioral and Brain Sciences*, 22(4):577–609; discussion 610–660, 1999.

[9] W. Bechtel and A. Abrahamsen. *Connectionism and the Mind*. Wiley, 1991.

[10] Cocks Bernadine, Nandagopal Nanda, R. Vijayalakshmi, M. Thilaga, Dasari Naga, and Dahal Nabaraj. Breaking the camel's back: Can cognitive overload be quantified in the human brain? *Procedia - Social and Behavioral Sciences*, 97(0):21 – 29, 2013. The 9th International Conference on Cognitive Science.

[11] O Black. Infinite regress arguments and infinite regresses. *Acta Analytica Philosophy and Psychology*, pages 95–124, 1996.

[12] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

[13] Ronen I. Brafman and Yagil Engel. Directional decomposition of multiattribute utility functions. In *ADT '09: Proceedings of the 1st International Conference on Algorithmic Decision Theory*, pages 192–202, Berlin, Heidelberg, 2009. Springer-Verlag.

[14] Urszula Chajewska and Daphne Koller. Utilities as random variables: Density estimation and structure discovery. In *In Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 63 – 71, 2000.

[15] Deng-Neng Chen, B. Jeng, Wei-Po Lee, and Cheng-Hung Chuang. An agent-based model for consumer-to-business electronic commerce. *Expert Systems with Applications*, 34(1):469 – 481, 2008.

[16] Sheldon J. Chow. Heuristics, concepts, and cognitive architecture: Toward understanding how the mind works. *Electronic Thesis and Dissertation Repository. Paper 128.*, 2011.

[17] Web3D Consortium. http://www.web3d.org/.

[18] C.H. Coombs and G.S. Avrunin. *Single-peaked Functions and the Theory of Preference: A Generalization of S-R Theory*. Michigan mathematical psychology program. University of Michigan, Department of psychology, 1976.

[19] Hermann Cuntz, Alexandre Mathy, and Michael Häusser. A scaling law derived from optimal dendritic wiring. 2012.

[20] M. Dehmer and M. Popovscaia. Towards structural network analysis. *Buletinul Academiei de Stiinte a Republicii Moldova*, 1(62):3–22, 2010.

[21] Theodoros M. Diasakos. Complexity and bounded rationality in individual decision problems. Carlo alberto notebooks, Collegio Carlo Alberto, 2008.

[22] Chuong B Do and Serafim Batzoglou. What is the expectation maximization algorithm? *Nature biotechnology*, 26(8):897–900, 2008.

[23] Carmel Domshlak. Modeling and reasoning about preferences with cp-nets, 2002.

[24] D. L. Donoho. High-dimensional data analysis: the curses and blessings of dimensionality. In *American Mathematical Society Conf. Math Challenges of the 21st Century*. 2000.

[25] J. Doyle. Bounded rationality. *MIT Encyclopedia of the Cognitive Sciences, Cambridge: MIT Press*, 1999.

[26] Jonathan St B.T. Evans. Reasoning with bounded rationality. *Theory & Psychology*, 2(2):237–242, 1992.

[27] A. Farinelli, A. Rogers, and N.R. Jennings. Agent-based decentralised coordination for sensor networks using the max-sum algorithm. *Journal of Autonomous Agents and Multi-Agent Systems*, 28(3):337–380, 2014.

[28] Alessandro Farinelli, Alex Rogers, Adrian Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *Seventh International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-08)*, pages 639–646, May 2008. Event Dates: 12-16 May 2008.

[29] Shaheen S. Fatima, Michael Wooldridge, and Nicholas R. Jennings. An agenda-based framework for multi-issue negotiation. *Artificial Intelligence*, 152(1):1 – 45, 2004.

[30] Shaheen S. Fatima, Michael Wooldridge, and Nicholas R. Jennings. An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions. In *Proc. of the Eighth Inernational Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2009)*, pages 1041–1048, 2007.

[31] J.A. Fodor. *The Modularity of Mind: An Essay on Faculty Psychology*. A Bradford book. A BRADFORD BOOK, 1983.

[32] Katsuhide Fujita, Takayuki Ito, Tim Baarslag, Koen Hindriks, Catholijn M. Jonker, Sarit Kraus, and Raz Lin. The second automated negotiating agents competition (anac2011). In Takayuki Ito, Minjie Zhang, Valentin Robu, and Tokuro Matsuo, editors, *Complex Automated Negotiations: Theories, Models, and Software Competitions*, volume 435 of *Studies in Computational Intelligence*, pages 183–197. Springer Berlin / Heidelberg, Springer Berlin / Heidelberg, 2013.

[33] Katsuhide Fujita, Takayuki Ito, Hiromitsu Hattori, and Mark Klein. An approach to implementing a threshold adjusting mechanism in very complex negotiations: A preliminary result. In *Proc. of the 2nd International Conference on Knowledge, Information and Creativity Support Systems (KICSS-2007)*, 2007.

[34] Katsuhide Fujita, Takayuki Ito, and Mark Klein. Preliminary result on secure protocols for multiple issue negotiation problems. In *Proc. of the 11th Pacific Rim International Conference on Multi-Agents (PRIMA-2008)*, 2008.

[35] Katsuhide Fujita, Takayuki Ito, and Mark Klein. A representative-based multi-round protocol for multi-issue negotiations. In *Proc. of the Seventh International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2008)*, pages 1573–1576, 2008.

[36] Katsuhide Fujita, Takayuki Ito, and Mark Klein. Approximately fair and secure protocols for multiple interdependent issues negotiation. In *Proc. of the Eighth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2009)*, pages 1287–1288, 2009.

[37] Katsuhide Fujita, Takayuki Ito, and Mark Klein. A secure and fair negotiation protocol in highly complex utility space based on cone-constraints. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent*

Technology - Volume 02, WI-IAT '09, pages 427–430, Washington, DC, USA, 2009. IEEE Computer Society.

[38] Katsuhide Fujita, Takayuki Ito, and Mark Klein. A secure and fair negotiation protocol in highly complex utility space based on cone-constraints. In *Proceedings of the 2009 International Joint Conference on Intelligent Agent Technology (IAT-2009)*, 2009 (Short Paper).

[39] Katsuhide Fujita, Takayuki Ito, and Mark Klein. An approach to scalable multi-issue negotiation: Decomposing the contract space based on issue interdependencies. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 02*, WI-IAT '10, pages 399–406, Washington, DC, USA, 2010. IEEE Computer Society.

[40] Katsuhide Fujita, Takayuki Ito, and Mark Klein. Efficient issue-grouping approach for multiple interdependent issues negotiation between exaggerator agents. *Decision Support Systems*, 60:10–17, 2014.

[41] Yong Gao. Treewidth of erdös-rényi random graphs, random intersection graphs, and scale-free random graphs. *CoRR*, 2009.

[42] E.H. Gerding, D.J.A. Somefun, and Han La Poutre. Efficient methods for automated multi-issue negotiation: Negotiating over a two-part tariff. *International Journal of Intelligent Systems*, 21:99–119, 2006.

[43] G. Gigerenzer, P.M. Todd, and ABC Research Group. *Simple heuristics that make us smart*. Evolution and cognition. Oxford University Press, 1999.

[44] Gerd Gigerenzer. Why Heuristics Work. *Perspectives on psychological science*, 3(1):20+, 2008.

[45] Gerd Gigerenzer, Peter M. Todd, and Research A. B. C. Group. *Simple Heuristics That Make Us Smart*. Oxford University Press, New York, 1999.

[46] K Goldberg, A. J. Goldman, and M. Newman. The probability of an equilibrium point. 72B:93, 1968.

[47] Rafik Hadfi and Takayuki Ito. Cognition as a game of complexity. In *In Proceedings of 12th International Conference on Cognitive Modeling (ICCM)*, 2013.

[48] Rafik Hadfi and Takayuki Ito. Uncertainty of cognitive processes with high-information load. *Procedia - Social and Behavioral Sciences*, 97(0):612 – 619, 2013.

[49] Viviane M. Heberle. Modularity versus connectionism: Two different views on the architecture of the mind.

[50] Siu-Wai Ho and Raymond W. Yeung. On the discontinuity of the shannon information measures. *IEEE Trans. Inf. Theor.*, 55:5362–5374, December 2009.

[51] Marty Holzer. Aircraft fasteners: How to select the correct type of fastener. *Aircraft Maintenance Technology*, 2011.

[52] Takayuki Ito, Hiromitsu Hattori, and Mark Klein. Multi-issue negotiation protocol for agents : Exploring nonlinear utility spaces. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pages 1347–1352, 2007.

[53] Takayuki Ito, Mark Klein, and Hiromitsu Hattori. A multi-issue negotiation protocol among agents with nonlinear utility functions. *Multiagent and Grid Systems*, 4(1):67–83, 2008.

[54] Matthew O Jackson. Mechanism theory. *The Encyclopedia of Life Support Systems*, 2000.

[55] Nicholas R. Jennings. An agent-based approach for building complex software systems. *Commun. ACM*, 44(4):35–41, April 2001.

[56] Paul E Johnson. *Social Choice: Theory and Research*. Number 123 in Quantitative Applications in the Social Sciences. SAGE Publications, 1998.

[57] B. D Jones. Bounded rationality. *In Annual Review of Political Science*, page 2Ð297, 1999.

[58] Olav Kallenberg. Convergence of non-ergodic dynamical systems. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 53(3):329–351, 1980.

[59] M. Kamermans and T. Schmits. The history of the frame problem. *Artificial Intelligence*, pages 86–116, 2004.

[60] Wende Ke, Zhiping Peng, Quande Yuan, Bingrong Hong, Ke Chen, and Zesu Cai. A method of task allocation and automated negotiation for multi robots. *Journal of Electronics (China)*, 29(6):541–549, 2012.

[61] Ralph L. Keeney and Howard Raiffa. *Decisions with multiple objectives*. Cambridge University Press, 1993.

[62] R.L. Keeney and H. Raiffa. Decisions with multiple objectivesâ preferences and value tradeoffs. *Behavioral Science*, 39(2):169–170, 1994.

[63] Christopher Kiekintveld, Zhengyu Yin, Atul Kumar, and Milind Tambe. Asynchronous algorithms for approximate distributed constraint optimization with quality bounds. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 133–140, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[64] Mark Klein, Peyman Faratin, Hiroki Sayama, and Yaneer Bar-yam. The dynamics of collaborative design: Insights from complex systems and negotiation research. *Concurrent Engineering Research & Applications*, 11:201–209, 2003.

[65] Mark Klein, Peyman Faratin, Hiroki Sayama, and Yaneer Bar-Yam. Negotiating complex contracts. *Group Decision and Negotiation*, 12(2):111–125, 2003.

[66] Mark Klein, Hiroki Sayama, Peyman Faratin, and Yaneer Bar-Yam. A complex systems perspective on computer-supported collaborative design technology. *Commun. ACM*, 45(11):27–31, November 2002.

[67] Michael Krainin, Bo An, and Victor R. Lesser. An application of automated negotiation to distributed task allocation. In *IAT*, pages 138–145. IEEE Computer Society, 2007.

[68] Sarit Kraus. *Strategic Negotiation in Multiagent Environments*. MIT Press, Cambridge, MA, USA, 2001.

[69] Sarit Kraus, Katia Sycara, and Amir Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1âĂŞ2):1 – 69, 1998.

[70] Frank Kschischang, Senior Member, Brendan J. Frey, and Hans-andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001.

[71] J. Kwisthout and G. Tel. Complexity results for enhanced qualitative probabilistic networks. *International Journal of Approximate Reasoning*, 48:879–888, 2008.

[72] Johan Kwisthout and Iris van Rooij. Bridging the gap between theory and practice of approximate bayesian inference. *Cognitive Systems Research*, 24(0):2 – 8, 2013. Cognitive Systems Research:Special Issue on ICCM2012.

[73] Guoming Lai and Katia Sycara. A generic framework for automated multi-attribute negotiation. *Group Decision and Negotiation*, 18:169–187, 2009.

[74] E Lehrer. Relative entropy in sequential decision problems. *Journal of Mathematical Economics*, 33(4):425–439, 2000.

[75] Raz Lin, Sarit Kraus, Tim Baarslag, Dmytro Tykhonov, Koen Hindriks, and Catholijn M. Jonker. Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 30(1):48–70, 2014.

[76] Hans-Andrea Loeliger. An introduction to factor graphs, 2004.

[77] Miguel A. Lopez-Carmona, Ivan Marsa-Maestre, Enrique De La Hoz, and Juan R. Velasco. A region-based multi-issue negotiation protocol for nonmonotonic utility spaces. *Computational Intelligence*, 27(2):166–217, 2011.

[78] Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Juan R. Velasco, and Enrique de la Hoz. Effective bidding and deal identification for negotiations in highly nonlinear scenarios. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 1057–1064, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[79] Ivan Marsa-Maestre, MiguelA. Lopez-Carmona, JuanA. Carral, and Guillermo Ibanez. A recursive protocol for negotiating contracts under non-monotonic preference structures. *Group Decision and Negotiation*, 22(1):1–43, 2013.

[80] John Mccarthy and Patrick J. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In *Machine Intelligence*, volume 4, pages 463–502, 1969.

[81] Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo. Adopt: Asynchronous distributed constraint optimization with quality guarantees. *ARTIFICIAL INTELLIGENCE*, 161:149–180, 2006.

[82] Abbe Mowshowitz and Matthias Dehmer. Entropy and the complexity of graphs revisited. *Entropy*, 14(3):559–570, 2012.

[83] P. La Mura and Y. Shoham. Expected utility networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 1999.

[84] Mark E. J. Newman, Albert L. Barabási, and Duncan J. Watts, editors. *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[85] R. Olfati-Saber, J.A Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, Jan 2007.

[86] David C. Parkes. Bounded rationality. Technical report, 1999.

[87] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[88] Valentin Robu, D. J. A. Somefun, and J.A. La Poutre. Modeling complex multi-issue negotiations using utility graphs. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2005)*, pages 280–287, 2005.

[89] Ariel Rubinstein. *Modeling Bounded Rationality*. MIT Press, 1997.

[90] A.K. Sen. *Collective choice and social welfare*. Mathematical economics texts. Holden-Day, 1970.

[91] Glenn Shafer and Judea Pearl, editors. *Readings in Uncertain Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[92] Yoav Shoham. A symmetric view of utilities and probabilities. In *Proceedings of the Fifteenth international joint conference on Artificial intelligence (IJCAI'97)*, pages 1324–1329, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[93] Sylvain Sorin. Merging, reputation, and repeated games with incomplete information. *Games and Economic Behavior*, 29(1-2):274–308, October 1999.

[94] Kim Sterelny. Cognitive load and human decision, or, three ways of rolling the rock up hill. In Peter Carruthers, Steve Laurence, and Steven Stich, editors, *The Innate Mind: Culture and Cognition*. Cambridge University Press, Cambridge, 2006.

[95] J. Surowiecki. *The Wisdom of Crowds: Why the Many are Smarter Than the Few and how Collective Wisdom Shapes Business, Economies, Societies, and Nations*. Doubleday, 2004.

[96] Hiroyuki Tamura and Yutaka Nakamura. Decompositions of multiattribute utility functions based on convex dependence. *Operations Research*, 31(3):488–506, 1983.

[97] Flemming Topsøe. Entropy and equilibrium via games of complexity. *Physica A: Statistical Mechanics and its Applications*, 340(1):11–31, 2004.

[98] J. von Neumann and O. Morgenstern. Theory of games and economic behavior, 2nd ed. princeton university press, princeton, nj. 1947.

[99] E Vul and H Pashler. Measuring the crowd within: Probabilistic representations within individuals. *Psychological Science*, 19(7):645–647, 2008.

[100] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and M J Weinberger. Inequalities for the l 1 deviation of the empirical distribution. *Group*, (HPL-2003-97R1):1 – 10, 2003.

[101] Michael P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44(3):257 – 303, 1990.

[102] Colin R Williams, Valentin Robu, Enrico H Gerding, and Nicholas R Jennings. Using gaussian processes to optimise concession in complex negotiations against unknown opponents. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence. AAAI Press*, 2011.

[103] C. B. Yang. The origin of power-law distributions in self-organized criticality. *Journal of Physics A Mathematical General*, 37:L523–L529, October 2004.

[104] R.W. Yeung. *Information theory and network coding*. Information technology–transmission, processing, and storage. Springer, 2008.

Publications

International Journals

[1] Rafik Hadfi and Takayuki Ito. “Low-Complexity Exploration in Utility Hypergraphs”. Journal of Information Processing. Vol. 23 (2015) No. 2 pp. 176-184

[2] Rafik Hadfi and Takayuki Ito. “An Agent-Mediated Architecture for Collective Collaborative Design”. International Journal of Electronics Communication and Computer Engineering (IJECCE), 6(2), 210-216. (2015)

[3] Rafik Hadfi and Takayuki Ito. Distributed Group Formation based on Decisional Structures and Linguistic Mediation Rules”. International Journal of Electronics Communication and Computer Engineering (IJECCE), 2014

International Conferences

[1] Rafik Hadfi and Takayuki Ito. “Approximating Constraint-based Utility Spaces using Generalized Gaussian Mixture Models”. The 17th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA2014), Gold Coast, QLD, Australia, December 1-5, 2014

[2] Rafik Hadfi and Takayuki Ito. “Modeling Complex Nonlinear Utility Spaces using Utility Hyper-graphs”. The 11th International Conference on Modeling Decisions for Artificial Intelligence (MDAI2014), Tokyo, Japan, October 29-31, 2014

[3] Rafik Hadfi and Takayuki Ito. “Addressing Complexity in Multi-Issue Negotiation via Utility Hypergraphs”. Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI2014), Québec City, Québec, Canada, July 27 -31, 2014

[4] Rafik Hadfi and Takayuki Ito. “Uncertainty of Cognitive Processes with High-Information Load”. The 9th International Conference on Cognitive Science (ICCS2013), Sarawak,

Malaysia, August 27-30, 2013

[5] Rafik Hadfi and Takayuki Ito. “Cognition as a Game of Complexity”. Proceedings of the 12th International Conference on Cognitive Modeling (ICCM2013), Carleton University, Ottawa, Canada, July 11-14, 2013

[6] Rafik Hadfi and Takayuki Ito. “Asymptotic Maximum Entropy principle for Utility Elicitation under High Uncertainty and Partial Information”. The 25th Florida Artificial Intelligence Research Society Conference (FLAIRS2012) (published by the AAAI), Marco Island, Florida, USA, May 23-25, 2012

International Workshops

[1] Rafik Hadfi and Takayuki Ito. “Constraint-based Preferences via Utility Hyper-graphs”. The 8th Multidisciplinary Workshop on Advances in Preference Handling (MPREF2014) (In conjunction with AAAI 2014), Québec City, Québec, Canada, July 27 -31, 2014

[2] Rafik Hadfi and Takayuki Ito. “On the Complexity of Utility Hypergraphs”. The 7th International Workshop on Agent-based Complex Automated Negotiations (ACAN2014), Paris, France, May 5-6, 2014

[3] Tim Baarslag, Rafik Hadfi, Koen Hindriks, Takayuki Ito and Catholijn Jonker. “Optimal Non-adaptive Concession Strategies with Incomplete Information”. The 7th International Workshop on Agent-based Complex Automated Negotiations (ACAN2014), Paris, France, May 5-6, 2014

[4] Rafik Hadfi and Takayuki Ito. “Application of the Maximum Entropy Principle to Preferences Merging: Case of Utility Functions with Infinite Outcomes”. The 5th International Workshop on Agent-based Complex Automated Negotiations (ACAN2012), Valencia, Spain, June 4-5, 2012

[5] Rafik Hedfi and Takayuki Ito. “Agreement among Agents based on Decisional Structures and its Application to Group Formation”. The 4th International Workshop on Agent-based Complex Automated Negotiations (ACAN2011), Taipei, Taiwan, May 2-6 2011

[6] Rafik Hedfi and Takayuki Ito and Katsuhide Fujita. “Towards Collective Collaborative

Design: An Implementation of Agent-Mediated Collaborative 3D Products Design System". International Workshop on Multi-Agent Systems and Collaborative Technologies (IMASC2010), Chicago, Illinois, USA, May 17 - 21, 2010

Domestic Conferences and Workshops

[1] Rafik Hadfi, Yoshitaka Torii, Takayuki Ito, Sho Tokuda, Ryo Kanamori and Masaaki Tanaka. "A Cloud-based Architecture for Congestion Management". The 77th National Convention of IPSJ (IPSJ2015), Kyoto, Japan, March 17-19, 2015

[2] Masaaki Tanaka, Takayuki Ito, Rafik Hadfi, Yoshitaka Torii, Sho Tokuda and Ryo Kanamori. "A Congestion Management Simulation based on the Negotiation Mechanism among Vehicles and its Application" (in Japanese). The 77th National Convention of IPSJ (IPSJ2015), Kyoto, Japan, March 17-19, 2015

[3] Rafik Hadfi and Takayuki Ito. "Unifying Constraint-based Utility Functions". Joint Agent Workshops and Symposium (JAWS2014), Kyushu, Japan, October 27-29, 2014

[4] Rafik Hadfi and Takayuki Ito. "Distributed group formation among agents based on decisional structures and linguistic mediation rules". Joint Agent Workshops and Symposiums (JAWS/iJAWS2012), Shizuoka, Japan, October 24-26, 2012

[5] Rafik Hadfi and Takayuki Ito. "The Interpretation of Heuristics based on the Maximum Entropy Principle". Information Processing Society of Japan (IPSJ2012), Nagoya, Japan, March 6-8, 2012

[6] Rafik Hedfi and Takayuki Ito. "Coalition Formation based on Decisional Structures." Information Processing Society of Japan (IPSJ2011), Tokyo, Japan, March 2-4 2011

[7] Rafik Hedfi and Takayuki Ito. "A Learning Interface Agent for Collaborative Multi-attribute Design using Semi-Supervised Clustering". First International Joint Agent Workshop and Symposium 2010 (JAWS/iJAWS2010), Furano, Hokkaido, October 29, 2010

Awards

1. Monbukagakusho (MEXT) Scholarship, April 2009–March 2015
2. AAAI Student Scholarship Award, American Association for Artificial Intelligence (2014)
3. ACAN2011 Best Student Presentation Award
   - Rafik Hedfi and Takayuki Ito. "Agreement among Agents based on Decisional Structures and its Application to Group Formation". The 4th International Workshop on Agent-based Complex Automated Negotiations (ACAN2011), Taipei, Taiwan, May 2-6, 2011
4. IEEE Computer Society Japan Chapter JAWS Young Researcher Award
   - Rafik Hadfi and Takayuki Ito. "Unifying Constraint-based Utility Functions". Joint Agent Workshops and Symposium (JAWS2014), Kyushu, Japan, October 27-29, 2014