Agreement among Agents based on Decisional Structures and its Application to Group Formation

Rafik Hedfi
Department of Computer Science, School of Techno-Business Administration,
Nagoya Institute of Technology.
rafik@itolab.mta.nitech.ac.jp

Takayuki Ito
Department of Computer Science,
School of Techno-Business Administration,
Nagoya Institute of Technology.
ito.takayuki@nitech.ac.jp

ABSTRACT
There has been an increased interest in automated negotiation systems for their capabilities in reaching an agreement through negotiation among autonomous software agents. In real life problems, the negotiated contracts consist of multiple and interdependent issues which tend to make the negotiation more complex. In this paper, we propose to define a set of similarity measures used to compare the agents' constraints, their utilities as well as their certainties over their possible outcomes. Precisely, we define a decision value-structure which gives a reasonable condition under which agents having similar decision structures can form a group. We think that a collaborative approach is an efficient way to reason about agents having complex decisional settings, but show similarities in their constraints, preferences or beliefs. Agents will tend to collaborate with agents having the same decisional settings instead of acting selfishly in a highly complex and competitive environment. Therefore, formed groups will benefit from the cooperation of its members by satisfying their constraints as well as maximizing their payoffs. Under such criterion, the agents can reach an agreement point more optimally and in a collaborative way. Experiments have been performed to test the existence of the decision value-structure as well as its capability to describe an agent's decision structure. Moreover, the decision value-structure was used for group formation based on measuring the agents similarities.

Keywords
Multi-attribute Utility, Decision Theory, Multi Objective Optimization, Uncertainty, Group Formation, Collaboration.

1. INTRODUCTION
Automated negotiation is a process by which a group of autonomous agents interact to achieve their design objectives. The agents will attempt to reach an agreement and satisfy their contradictory demands through a bargaining process. In an agent-mediated system, an important aspect of the solution is the way in which the agents negotiate to propose contracts to each other, under specific requirements and constraints. In real life situations, agents have to take into consideration multiple attributes simultaneously during the bargaining process, such as the quality, quantity, delivery time, etc. ([7]). In this paper, we propose to define a new approach to tackle the complexity of utilities with interdependent attributes by providing a new model for multi-attribute utility representation, which takes into consideration the possible interdependencies between attributes. In the real world, we believe that people who have similar decisional structures could reach an agreement more smoothly. In this paper, we propose also a new criterion for potential consensus under a number of assumptions, related to the decisional structure of the agent, defined as a Constraint-Utility-Belief space. In fact, adopting a cooperative behavior during the negotiation process may improve the performance of the individual agents, as well as the overall behavior of the system they form, by achieving their own goals as a joint decision [6]. To put this straightforward, we assume that our model is based on the following assumptions. In real life, we believe that people who have similar beliefs (certainties) relative to a specific situation, as well as the same preferences (utilities) over the same common outcomes (attributes), could reach a reasonable agreement more optimally and smoothly, than if they had different certainties or preferences over different outcomes. To support this claim, we first describe the different aspects of the decisional structure of an agent as a Constraint-Utility-Belief space. Most importantly, we define a unique decision value-structure for each agent, which gives a reasonable criterion, under which agents' decisional structures can be compared. We point out that in the case of similar decision value-structures, the agents can form groups, as an initial step before making coalitions which satisfy their constraints and maximizes their payoffs. Therefore, the agents can reach an agreement point more efficiently and in a collaborative way. We argue that the advantage of such approach is that the agents having strongly different decisional structures *i.e.* different decisional value-structures, do not need to cooperate. Instead, they can find agents having similar settings, and form groups.

At this end, in the case of multi-attribute negotiation we must define the main components needed by an agent to make decisions.

There have been several works in the context of multi-attribute negotiation for its importance in commerce as well as in social interactions. Different approaches and methods were proposed to analyze multi-attribute utilities for contracts construction. [12] presented the notion of convex dependence between the attributes as a way to decompose utility functions. [9] proposed an approach based on utility graphs for negotiation with multiple binary issues. [2] proposed also a model inspired from Bayesian and Markov models, through a probabilistic analogy while representing multi-attribute utilities. The same idea was firstly introduced by [11] 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. In another work by [8], a similar concept was introduced by the notion of Expected Utility Networks which includes both utilities and probabilities. [3] 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 remainder of the paper is structured as follows. Section 2 provides a formal definition of our model based on the notion of Decisional Structure of an agent with all its components. Section 3 describes a method used by the agent to construct his proposals or contracts, based on his decisional structure. In section 4, we elaborate a possible usage of the decisional structure as a group formation criterion through a set of similarity metrics. In section 5, we generalize the use of those metrics by the Decisional Value-Structure function as a method to compare agents' decisional structures. The experiment and the analysis of the model are described in section 6. In section 7 we present the conclusions and outline the future work.

2. DECISIONAL STRUCTURE

In the following section we will provide an overview of our theoretical model used for the representation of an agent's decisional structure. In fact, by decisional structure, we refer to the overall settings or information used by the decision maker *i.e.* the agent, to elaborate his strategies and make his decisions. In other words, the decisional structure of an agent can be considered as the decision space of the agent representing all his possibilities. Therefore, we will initially focus on a microscopic representation of an agent i regardless from his environment or the other agents. The macroscopic view will be developed in the next sections in the case of group formation. An agent i will define a unique tuple (1) representing his decisional structure.

$$Agent\ i \longmapsto (G_i, U_i, B_i) \tag{1}$$

This tuple will be characterized by the attributes and constraints of the agent i, represented by a Directed Acyclic Graph G_i [2] . The preferences of the agents will be represented by the utilities U_i of the agent. The agent's beliefs or certainties will be represented by the probability distributions B_i. The tuple can be described in the equations (2).

$$G_i = (V_i, E_i) \tag{2a}$$

$$V_i = \{v^i_j \sim a^i_j\}^n_{j=1},\ a^i_j = (x_1, ...x_{m_j}) \tag{2b}$$

$$E_i = V_i \times V_i = \ \{d_j\}^{m_d}_{j=1} \tag{2c}$$

$$U_i = \ \{u^i_j\}^n_{j=1} \tag{2d}$$

$$B_i = \ell_i = \ \{\ell^i_j\}^n_{j=1} \tag{2e}$$

$$= \ \{\ \ell^i_j\ [p_{i,j,1} : x_{i,j,1},\ \ldots p_{i,j,m} : x_{i,j,m_j}]\ \}^n_{j=1} \tag{2f}$$

The static structure of the agent in (2a), defines the attributes (2b) and the dependencies (2c) between them, represented as a Directed Acyclic Graph G_i. In (2b), each vertexv^i_j of the graph corresponds to an attribute a^i_j *i.e.* an outcome or a prospect. An attribute a^i_j is defined as a vector of the possible values that can be taken by a^i_j. In the discrete case a^i_j (2b) and in the continuous case $a^i_j \in [x_1,\ x_{m_j}]$. In (2c), constraints are represented by the arcs $\{d_j\}^{m_d}_{j=1} \subset G_i$, and connect the vertices representing dependent attributes. But, it can be used to compute the utilities by mirroring the same dependence structure as a conditional dependence between the utilities [11]. This dependence structure could be updated dynamically during a negotiation process when the agents are collaborative. In (2d), utility functions U_i of the agent i represented as a function-vector $\{u^i_j\}^n_{j=1}$. In our model, we assume that the decision maker *i.e.* the agent follows the axioms of normative utility functions ($\sum_j u^i_j = 1$) [13]. Furthermore, we assume that the used utility functions have the properties of non-satiation (${u'}^i_j(x) > 0$) and risk aversion (${u''}^i_j(x) < 0$) [5]. Each utility function u^i_j is defined over a domain D_j related to the possible values taken by the attribute a_j as in (3).

$$u^i_j : D_j \rightarrow [0, 1] \tag{3}$$

Another important aspect of our utility functions is that they are defined in term of dependencies as conditional utilities, and therefore embody the notions of conditional probabilities and probability independence [11]. In our model, we use this representation for the computation of the utilities in respect to the functional dependencies. We refer the reader to the work proposed in [2] and related to conditional utilities and the conditional independence. In (2e), the belief or the certainty structure B_i of an agent i characterized by all the lotteries $\{\ell^i_j\}^n_{j=1}$ (2f) where each lottery ℓ^i_j is associated to the attribute a^i_j, according to the probability distribution $p_{i,j}$ over the outcomes $x_{i,j,k} \in a^i_j$ with $\sum^{n_j}_{k=1} p_{i,j,k} = 1$. The lotteries of an agent i over the set of attributes a^i_j can be represented by the lottery (4).

$$\ell^i_j\ [p_{i,j,1} : x_{i,j,1},\ \ldots, p_{i,j,n} : x_{i,j,n}] \tag{4}$$

The probabilities $p_{i,j}$ are the subjective probabilities [1] of the agent i and represent his certainties about the possible outcomes. Each probability associated to an attribute, can be seen as a random variable over the possible values of an attribute [3].

3. UTILITY MAXIMIZATION

3.1 Contract Representation

An agent i will represent a contract $\vec{C^i}$ as a vector of attributes $\vec{C^i} = (\ a^i_1, \ldots, a^i_j, \ldots, a^i_n\ )$, where each attribute corresponds to a vertex $v^i_j \in V_i$ as we mentioned in (2b). Therefore, finding the optimal contract $\vec{C^*}$ having the highest utility among the contracts $\vec{C}_{i \in N}$, corresponds to solving the objective function (5) [4].

$$\vec{C^*} \ = \ \arg\max_{\vec{C}} \sum_{i \in \mathbb{N}} u_i(\vec{C_i}) \tag{5}$$

However, we assume the existence of a number of constraints, describing the relations or interdependencies (2c) between the attributes [2] . In other words, to compute the utility of a single attribute, we must take into consideration the other attributes. Meanwhile, we will associate a specific utility function u_i to each attribute a_i, with i as an attribute index. The overall utility of a contract $\vec{C}$ can be represented in the equation (6).

$$u(\vec{C}) \ = \ \sum_{a_i \in \vec{C}} u_i(a_i / \{a_{j \neq i}\}) \tag{6}$$

It is obvious that none of the overall attributes are needed to compute the utility of a single attribute. It means that based on a graphical representation of the interdependencies (2c), we will only use the connected attributes. The edges d_i representing the constraints or dependencies between attributes. Since the dependencies will exist only between the connected vertices, each vertex a_i will de-

Table 1: Conditional Utility functions

Utility u_i	Conditional Utility $u_i/\{u_j\}_{j=1}^{7}$
u_1	u_1
u_2	u_2
u_3	$u_3/\{u_1, u_2\}$
u_4	u_4
u_5	$u_5/\{u_3, u_4\}$
u_6	u_6
u_7	$u_7/\{u_4\}$

pend on its parent vertices giving the equation (7).

$$u(\vec{C}) \;=\; \sum_{a_i \in \vec{C}} u_i(a_i/\pi(a_i)) \tag{7}$$

Where $\pi(a_i)$ is the set of all the parents of the vertex a_i. This representation means that in order to compute the utility of the attribute a_i we need to use the attributes $\pi(a_i)$ and their corresponding utility functions. Therefore, the objective function (5) can be written as $\vec{C^*} \;=\; \arg\max_{\vec{C}} u(\vec{C})$. The final equation is described as in (8)

$$\vec{C^*} \;=\; \arg\max_{\vec{C}} \sum_{a_i \in \vec{C}} u_i(a_i/\pi(a_i)) \tag{8}$$

3.2 Example of Contract Construction

Suppose we are dealing with contracts with a number of attributes equal to 7. The goal is to find the optimal contract $\vec{C^*}$ satisfying the interdependencies between the attributes. Each agents will organize his attributes and constraints in a specific way defined by the Directed Acyclic Graph in **Figure 1**.

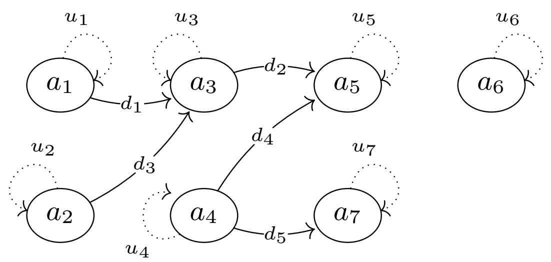

Figure 1: Constrained attributes

As we can see in **Figure 1**, the DAG will represent the contract from a statical viewpoint *i.e.* the structure and the interdependencies between the attributes. Moreover, a utility function u_i has to be associated to each vertex v_i, in order to compute the utility of the corresponding attribute a_i. Based on the graph in **Figure 1**, the interdependency relations between attributes will yield the same dependencies among the utility functions as shown in **Table 1**. In the concrete case, an attribute a_j can have different values and therefore will be represented by a vector $a_i \;=\; \{x_j \in D_i\}_{j=1}^{m_j}$ Maximizing an utility function u_i is finding the value $x^* \in D_i$ representing the maximal extrema of u_i such as in (9).

$$u_i(x^*) \;\geq\; u_i(x_k)\, \forall k \in [1, m_j] \tag{9}$$

Thus, we are interested in maximizing the sum of the increasing functions U_i. Therefore, the optimal contract can be written as a vector $\vec{C^*} \;=\; (\ a_1^*, \ldots, a_i^*, \ldots, a_n^*\ )$, where a_i^* is the maxima of u_i. The optimal contract's utility is computed according to the equation (10).

$$u(\vec{C^*}) \;= \sum_{i \in \mathbb{N}} u_i(a_i^*/\pi(a_i^*)) \tag{10}$$

3.3 Agent's Optimal Contract

The algorithm **Optimal_Contract** is used to find the optimal contract based on the attributes (2b), the utilities (2d), and the interdependencies among the attributes (2c).

Algorithm: Optimal_Contract
Input: DAG G_i of the Agent i
Output: Optimal Contract C^*

begin
  Topologic ordering of a_i according to $\pi(a_i)$;
  for $k \leftarrow |\pi(a_i)|_{min}$ **to** $|\pi(a_i)|_{max}$ **do**
    foreach a_i *satisfying* $|\pi(a_i)| = k$ **do**
      Find a_i^* satisfying
      $u_i(a_i^*) \geq u_i(x_j),\ j \in [1, m_i],\ x_j \in D_i$;
    end
  end
  $C^* \leftarrow (\ a_1^*, a_2^*, a_3^*, \ldots, a_i^*, \ldots, a_n^*\ )$;
  return C^*
end

Algorithm 1: Optimal contracts finding

Based on our example in **Figure 1**, the vertices a_i will be sorted according to the number of parents *i.e.* the in-degree $deg^-(a_i)$, which will describe the number of constraints of the related attribute.
An attribute a_i with $deg^-(a_i) = 0$ is called a *free attribute*, as the corresponding utility is computed only by using the attribute a_i's utility function u_i without any reference to other utility functions or other attributes. Similarly, an attribute with $deg^+(a_i) > 0$ is a *non-free attribute* or *dependent* and is subject to $deg^+(a_i)$ constraints. The topological sort of the attributes a_i within G_i is based on the $deg^-(a_i)$.

4. GROUP FORMATION

4.1 Group formation metrics

The nonlinearity and the complexity of the agents preferences is basically due to the different constraints they are trying to satisfy, as well as their utilities and the way probabilities are affected. Generally, our approach tends to capture and analyze the similarities between the agents constraints, utilities and beliefs. Being part of the same group means that all its members have close constraints, utilities and certainties. Therefore, it is important to define the similarity functions, to be able to compare between two agents' decisional spaces and decide whether they can be part of the same group or not.

4.2 Metric related to the Graph

We define the measure sim as the degree of similarity between two graphs G_1 and G_2. In other words, how much the agents whose graphs G_1 and G_2 share constraints and how close they are in term of vertices and edges. The similarity measure is calculated by multiplying the $Jaccard$ indexes relative to the vertices and the edges sets.

This similarity measure can be definied by (11).

$$sim : G \times G \to [0,1] \tag{11a}$$

$$sim(G_1, G_2) = J_V(V_1, V_2) \times J_E(E_1, E_2) \tag{11b}$$

$$= \frac{|V_1 \cap V_2|}{|V_1 \cup V_2|} \times \frac{|E_1 \cap E_2|}{|E_1 \cup E_2|} \tag{11c}$$

The extreme value $sim(G_1, G_2) = 0$ means that the agent 1 and the agent 2 do not have the same attributes nor share common constraints, whereas $sim(G_1, G_2) = 1$ means that they have exactly the same attributes and the same constraints. Therefore, it might be interesting to consider these similarities' measures between agents' DAGs as a way to form groups and maybe think of potential coalitions. Under these hypothesis, each agent i has a vector $SG_i = \{sim(G_i, G_k)\}_{k \neq i}$ containing all the similarity values between his graph G_i and the other agents' graphs G_k. Using this vector, the agent can selected the set of agents having similar structures (attributes, constraints). This can be a first step for a future collaboration between the agents being part of the same group.

4.3 Metric related to the Utilities

As mentioned in **2**., the utility functions have the properties of *non-satiation* and are *risk aversion* . Under these hypothesis, we assume that the behavior of these functions can be used to compare the utilities of two agents. Let's consider two utility functions $u_i : D_i \to [0,1]$, $u_j : D_j \to [0,1]$ and the domain $D = D_i \cap D_j$. If we suppose that u_i and u_j are similar ($u_i \sim u_j$), then (12) holds.

$$u_i \sim u_j \Longrightarrow \forall x \in D, \exists \epsilon, |u_i(x) - u_j(x)| \leq \epsilon \tag{12}$$

The main purpose of comparing utility functions is finding a similarity measure enabling us to say whether two agents have the same preferences over the same outcome (attribute) or not. We can propose a way to compare two agents' utilities by comparing their accumulated wealth for the same outcome x. In this case, we have to consider the utility value as if it was a cumulative distribution function. Comparing two agents' utilities u_i and u_j is comparing their integrations from the last preferred outcome x_{min} up to the outcome x. Therefore (13) holds.

$$u_i \sim u_j \Longrightarrow \int_{x_{min}}^{x} (u_i(x) - u_j(x))\, \mathrm{d}x \simeq 0 \tag{13}$$

The comparison measure of two utility functions u_i and u_j up to an outcome x will be defined as in (14).

$$sim(u_i, u_j) = \int_{x_{min}}^{x} (u_i(x) - u_j(x))\, \mathrm{d}x \tag{14}$$

We notice that both utilities have the same type *i.e.* correspond to the same outcome (domain). Therefore comparing the overall n utilities U_i and U_j of two agents i and j can be determined as in (15).

$$sim(U_i, U_j) = \prod_{k=1}^{n} sim(u_k^i, u_k^j) \tag{15}$$

4.4 Metric related to Beliefs

The agents have different certainties when it comes to decide about the outcomes and their related preferences. Therefore, we think about a way to compare these certainties defined as lotteries. Two agent i and j will share the same certainties (beliefs) for an outcome a_k, if their respective probability distributions p_k^i and p_k^j over a_k are close or similar. A possible way to consider this similarity is to use the cross entropy. Assuming that for a certain attribute $a_k = (x_1, ...x_{m_k})$ and for two lotteries ℓ_k^i and ℓ_k^j relative to two agents i and j, each lottery will correspond respectively to a probability distributions p_k^i and p_k^j over a_k. Therefore, we can define the cross entropy of p_k^i and p_k^j as in (16).

$$sim(p_k^i, p_k^j) = \sum_{l=1}^{m_k} p_k^i(x_l) \log[p_k^j(x_l)] \tag{16}$$

Generally, each agents i has a vector of lotteries ℓ_i over the n attributes and defined as his certainty structure B_i as in (2e) and (2f). We can define a similarity measure comparing two agent's certainty structures B_i and B_j as in (17).

$$sim(B_i, B_j) = \sum_{k=1}^{n} sim(p_k^i, p_k^j) \tag{17}$$

5. DECISIONAL STRUCTURE VALUE FUNCTION

After defining the agent's metrics we will focus on how to exploit them in order to satisfy the common constraints as well as the possible similarities between the agents's belief and utilities. For example, the agents sharing the same constraints (same graphs structure) and having the same beliefs (same probability distributions over the outcomes) could form groups by opening and sharing their utility functions according to a specific strategy. As in (1), the tuple (G_i, U_i, B_i) of an agent i describes his constraints, preferences and beliefs in a way that identifies the agent from the other agents' configurations. However, if the values G_i, U_i and B_i represent in a unique way their corresponding agent, it is possible to construct a bijective function f which maps each agents tuple (G_i, U_i, B_i) to a unique real value $dsv_i \in [0,1]$ identifying the agent in a unique way. This function can be assimilated to an *Hilbert Space Filling Curve* [10] or can be constructed by a binary expansion of real numbers. This function can be described by the definition (18).

$$f : D_J \times D_U \times D_P \to [0,1] \tag{18a}$$

$$f(g_i, u_i, p_i) = dsv_i \tag{18b}$$

The domains D_J, D_U and D_P of f are equal to [0,1]. We will develop in the next section the proper use of this function f in the context of group formation and agents clustering. The function f must be injective $i.e.$ for two agents i and j having different settings (g_i, u_i, b_i) and (g_j, u_j, b_j) we will have (19).

$$(g_i, u_i, b_i) \neq (g_j, u_j, b_j) \Longrightarrow f(g_i, u_i, b_i) \neq f(g_j, u_j, b_j) \tag{19}$$

It is possible to prove not only the existence of an injection from $[0,1]^3$ to $[0,1]$ but also a bijection. In fact, that bijection exists and it can be proven using the *Cantor-Bernstein-Schroeder* theorem as following :

i. There is an injection g satisfying (20).

$$g : [0,1] \to [0,1]^3 \tag{20a}$$

$$g(x) = (x, 0, 0) \tag{20b}$$

ii. It is possible to define an injection $h : [0,1]^3 \to [0,1]$ given by representing the tuple (x, y, z) in binary and then interlacing the digits before interpreting the result in base 10, yielding the image of (x, y, z). Using binary for the representation of the strings is a way to avoid the 9's with the dual representation in base 10 and therefore, preserving the injection.

iii. Based on $i.$ and $ii.$, we can apply the *Cantor-Bernstein-Schroeder* theorem, which states that if there are two injections g and h as in (21a) and (21b),

$$g : A \to B \tag{21a}$$
$$h : B \to A \tag{21b}$$

Then there is a bijection f between A and B. Hence, it is possible to find f satisfying the condition (19).

An interesting usage of the function f is in a mediated negotiation where a mediator is gathering bids from the agents and trying to find the optimal contract. In fact, f provides to the mediator a way to group the agents based on their similarities without the need for the agents to open their utility spaces or their constraints. In this situation, the mediator can establish a feedback mechanism to update his constraints according to the settings of the agents. The convergence to the optimal solutions, ensuring social welfare, will be based upon the agents' feedback as well as the initially established mediator's constraints. Each agent i has only to provide the decisional structure value (dsv) which can be seen as a fuzzy indicator about the agent's Constraint-Utility-Belief Space ($[0,1]^3$). Once these values are collected, the mediator can analyze and predict the possibilities of consensus reaching and the convergence to final contract. This is done before starting any utility space sampling or any computationally consuming task, used for example in [4].
The main advantage of using the dsv is to avoid bidding when the bids are likely to yield a complex and nonlinear utility space. Furthermore, having nonlinear space tends to make the consensus finding process complex, especially when there is a mediator. In fact, the mediator has to collect the bids and explore a highly nonlinear utility space in order to find the Pareto optimal contracts [4]. Instead, we can find an appropriate grouping of the bids based on certain criteria (including similarity measures) defined by the decisional structure values of the agents.
As we mentioned above, f is bijective, as the agents do not need to open their utilities nor their belief nor their constrains. Instead, they can know exactly how close and how similar their decision structures are and hence to decide whether to go for a collaborative strategy or act regardless from the others. The closeness degree between two agents stands upon the monotonicity of f when mapping to [0,1]. The function f can capture enough information that allows a meaningful clustering of agents based on their common interests : Constraints, Attributes, Utilities, Belief, Certainty, etc.

6. EXPERIMENTAL ANALYSIS

In the following experiments, we provide a method for group formation based on the similarity between the decisional values of the agents. We also provide an application of the decisional structure in the design of vectors called *vectorial design*.
Given the set $C = \{d_i\}_{i=1}^{N}$ of all the decisional structure values (dsv) of the agents, we propose to partition C into k disjoint clusters using the *K-Means* algorithm. Finding the optimal partitioning of C corresponds to finding the k clusters as in (22).

$$C^* = \arg\min_{C} \sum_{i=1}^{k} \sum_{d_j \in C_i} \|d_j - \delta_i\|^2 \tag{22}$$

Each cluster or group C_i is centered around a specific structure value δ_i which refers to the agent having the decisional structure that is more likely to describe the common features of the group C_i.

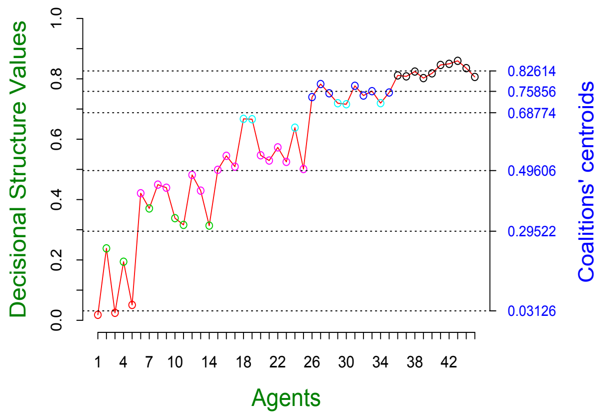

Figure 2: Agents' *dsv* values

Figure 2 illustrates a process of grouping of 45 agents, based on their decisional structure values *i.e.* d_j. We propose to partition these agents into 6 groups each of which is characterized by a group centroid δ_i. The resulting groups can be described by their corresponding centroids which are represented in **Figure 2** in blue, on the right axis.

The decisional structure values δ_i were generated based on the function f defined in **5.**, which was applied on the G_i, U_i and B_i variables of the 45 agents. The corresponding DSVs must be unique for each agent. Under such hypothesis, the injectivity of the function f will stand and there will be no risk for collisions *i.e.* two different agents, having different decisional structures but having the same DSV. Based on the original tuples G_i, U_i and B_i, we found that the agents being part of a group (C_j, δ_j) had close constraints, utilities and probabilities. This result was evaluated firstly by comparing the similarities between two agents decisional structure values dsv_i and dsv_j based on the distance $d = |dsv_i - dsv_j|$. Secondly, we measured the distances $d_g = sim(G_i, G_j)$, $d_u = sim(U_i, U_j)$ and $d_b = sim(B_i, B_j)$, defined in **4**. We found that the distance d is related to the distances d_g, d_u and d_b. The result confirms the characteristics of the bijective function f defined in **5.**, and its ability to describe uniquely an agent's decisional structure.
In **Figure 3**, we can see that there is a number of agents grouped around the same dsv value. In this case, the agents 2, 3, 4, 5 and 9 can be grouped into a cluster G based on the assumption that they have common decisional structures. According to this information, and whenever its shared to the overall agents (1 to 10), the agents not being part of G can choose to join this group or not. In case they accept to join, it is probable that they should start adapting and updating their constraints, preferences and beliefs similarity to the initial agents of G.

Generally, The decisional value structures are constructed based on the graphical constraints, utilities and beliefs. As we can see in **Figure 4**, the red curve represents the graphical constraints values, the utilities are represented by the green curve, and the blue values represent the beliefs. The overall similarity is represented by the black curve. For example, we can see that the agent 1 and

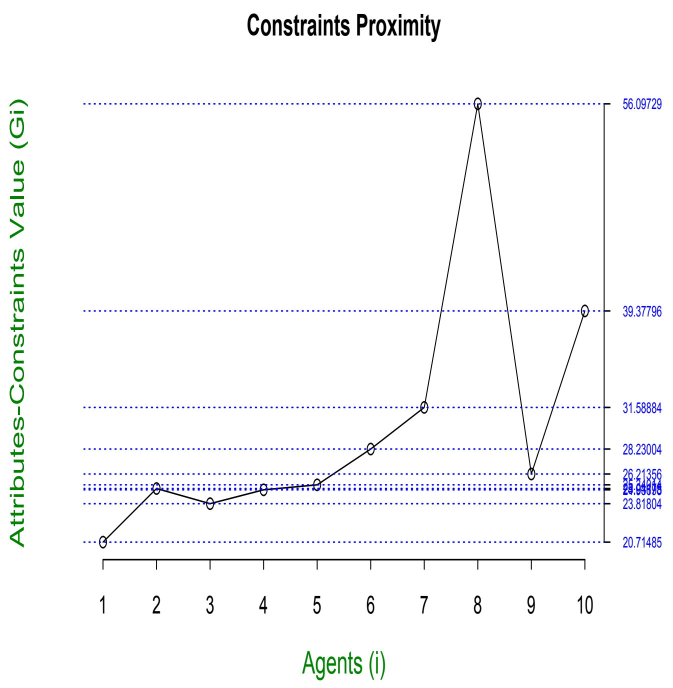

Figure 3: Dominant Group

Table 2: Agents' vectors values

$Agents$	x_1	x_2	$Values$
A_1	0.12	0.96	0.05683
A_2	1.87	1.83	0.68083
A_3	1.34	1.45	0.38637
A_4	1.41	1.57	0.44097
A_5	2.32	2.92	1.36735
A_6	2.39	3.01	1.4523

the agent 6 have close DSVs, and this can be seen based on the closeness in the red, green and blue curves *i.e.* the graphical constraints points, utilities and belief points.

A concrete application of such method of comparison is the case of *vectorial design*, where a user designs graphically a vector. A vector can represent an object, a product, or more generally a multi-attribute contract. As an example, 6 agents are designing 6 different vectors. For the sake of simplicity, we can think about the vector as a 2-points vector with components x_1 and x_2. In **Table 2** we can see that for each two values x_1 and x_2 we can represent the design vector by a unique value, locating the agents design in the overall designed vectors. This will give an idea about the degree of closeness between the designed vectors. The degree of closeness of the agents's vectors can be provided as a shared information to the overall agents while they are designing their vector. In fact, sharing such information dynamically and in real time can give the agents an idea on how their vectors are located in the group, and how to change their vector accordingly. This information can be represented as in **Figure 5**, and is available to each agent. On the x axis, we have the agents's indexes from 1 to 6 represented by 6 bars, and on the y axis we represent their corresponding values. Whenever an agent changes his vector, the representation in **Figure 5** will change accordingly. Such method of collaborative design will give the agents the possibility to orient their design based on the overall group's preferences, ensuring social welfare. It is possible to extend the simple vector represented by x_1 and x_2 to a more complex vector. Another example of vectorial design is represented in **Figure 6** where 7 agents are designing 7 vectors. At different times, each agent A_i will provide a vector $V_{A_i} = (X_{i1}, X_{i2})$, where X_{ij} are real values. During the design process, each agent A_j can visualize the similarities between his design and the other agents $A_{k \neq j}$ as in **Figure 7**. Therefore A_j can update his vector according to the evolution of the other agents' designs.

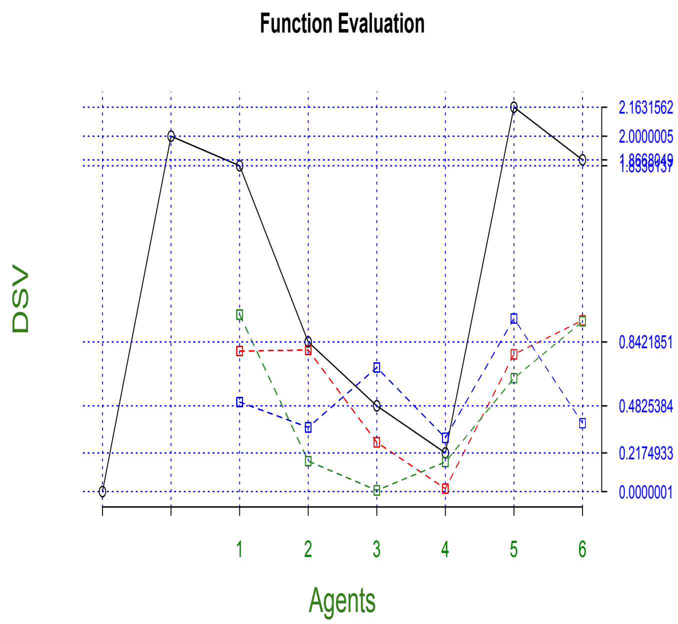

Figure 4: DSVs comparison

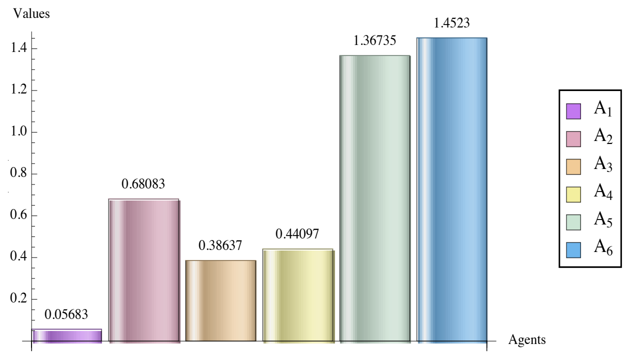

Figure 5: Proximity of the designed vectors

The represented values in **Figure 7** correspond to the designed vectors represented in **Figure 6**. We can can see that the vectors V_{A_6} and V_{A_5} are graphically close in **Figure 6**, therefore their corresponding values in **Figure 7** will be also close (1.30555 and 1.4523). The same comparisons can be done to the other vectors,

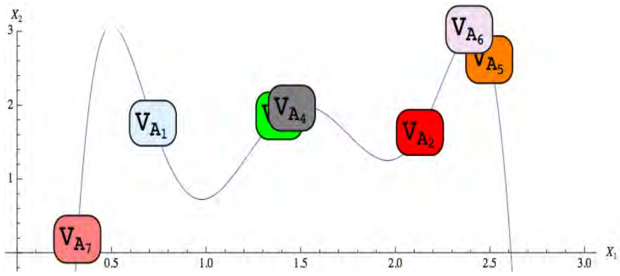

Figure 6: Vectors representation

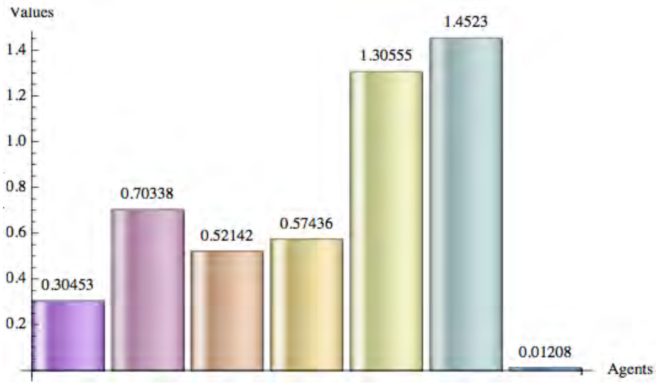

Figure 7: Decisional Values representation

allowing the agents to see the likelihood and the convergences of the global design.

7. CONCLUSION

The contributions of this paper are two-fold. On the one hand, we proposed a theoretical model to reason about multi-attribute contracts representation taking into consideration the attributes' interdependencies. On the other hand, we provided the notion of decisional structure value as a main criterion for agents' decisional settings comparison. The defined structure-value captures the main similarities between the agents' decisional settings. We have shown that it is possible to represent such decisional setting as a Constraints-Utilities-Belief space. Furthermore, we provided an example of usage of such value in the case of group formation based on the degree of similarity between the agent's decisional spaces.

As a future work, we would like to consider the performances of the method used to generate the decisional structure value. Moreover, we would like to elaborate a complete negotiation process, by defining a concrete protocol based on the formed groups. For example, we can develop the case where the agents being part of the same group can open and share their utility functions.

8. REFERENCES

[1] F. J. Anscombe and R. J. Aumann. A definition of subjective probability. *The Annals of Mathematical Statistics*, 34(1):199–205, 1963.

[2] R. I. Brafman and Y. 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.

[3] U. Chajewska and D. 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.

[4] T. Ito, H. Hattori, and M. 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.

[5] R. L. Keeney and H. Raiffa. *Decisions with multiple objectives*. Cambridge University Press, 1993.

[6] S. Kraus and O. Schechter. Strategic-negotiation for sharing a resource between two agents. *Computational Intelligence*, 19:9–41, 2003.

[7] G. Lai and K. Sycara. A generic framework for automated multi-attribute negotiation. *Group Decision and Negotiation*, 18:169–187, 2009.

[8] P. L. Mura and Y. Shoham. Expected utility networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 1999.

[9] V. Robu, D. J. A. Somefun, and J. L. 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.

[10] H. Sagan. Space-filling curves. New York, 1994. Springer-Verlag.

[11] Y. 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.

[12] H. Tamura and Y. Nakamura. Decompositions of multiattribute utility functions based on convex dependence. *Operations Research*, 31(3):488–506, 1983.

[13] J. von Neumann and O. Morgenstern. In *Theory of Games and Economic Behavior, 2nd ed. Princeton University Press, Princeton, NJ.*, 1947.