Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Addressing Complexity in Multi-Issue Negotiation via Utility Hypergraphs

Rafik Hadfi and **Takayuki Ito**
Department of Computer Science and Engineering
Graduate School of Engineering, Nagoya Institute of Technology
Gokiso, Showa-ku, Nagoya 466-8555, Japan
rafik@itolab.nitech.ac.jp, ito.takayuki@nitech.ac.jp

1 Introduction

There has been a great deal of interest about negotiations having interdependent issues and nonlinear utility spaces as they arise in many realistic situations (Ito, Hattori, and Klein 2007; Marsa-Maestre et al. 2013). In this case, reaching a consensus among agents becomes more difficult as the search space and the complexity of the problem grow. Nevertheless, none of the proposed approaches tries to quantitatively assess the complexity of the scenarios in hand, or to exploit the topology of the utility space necessary to concretely tackle the complexity and the scaling issues.

We address these points by adopting a representation that allows a modular decomposition of the issues and constraints by mapping the utility space into an issue-constraint hypergraph. Exploring the utility space reduces then to a message passing mechanism along the hyperedges by means of utility propagation. Adopting such representation paradigm will allow us to rigorously show how complexity arises in nonlinear scenarios. To this end, we use (Hadfi and Ito 2013) for complexity assessment 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. We evaluated our model using parametrized random hypergraphs, showing that it can optimally handle complex utility spaces while outperforming previous sampling approaches.

2 Approach Overview

We start from the formulation of nonlinear multi-issue utility spaces used in (Ito, Hattori, and Klein 2007). The contract space is an $n-$dimensional utility space, defined over a finite set of issues $\mathbb{I} = \{i_1, \dots, i_k, \dots, i_n\}$. The issue k, namely i_k, takes its values from a finite set $\mathbb{I}_k$ with $\mathbb{I}_k \subset \mathbb{Z}$. A contract $\vec{c}$ is a vector of issue values with $\vec{c} \in \times_{k=1}^{n} \mathbb{I}_k$. An agent's utility function is defined in terms of m constraints, making the utility space constraint-based. The constraint c_j is a region of the $n-$dimensional utility space. We say that c_j has value $w(c_j, \vec{c})$ for contract $\vec{c}$ if c_j is satisfied by $\vec{c}$. That is, when the contract point $\vec{c}$ falls within the hyper-volume defined by constraint c_j, namely $hyp(c_j)$. The utility of $\vec{c}$ is thus defined as $u(\vec{c}) = \sum_{c_{j \in [1,m]},\, \vec{c} \in hyp(c_j)} w(c_j, \vec{c})$. Figure 1a illustrates an example of $2-$dimensional utility spaces.

New representation. The utility u is nonlinear in the sense that it does not have a linear expression against $\vec{c}$. This is true to the extent that the linearity is evaluated with regard to the issues of $\vec{c}$. However, from the same expression, we can say that the utility is in fact linear, but in terms of the constraints c_j. The utility space is therefore decomposable according to the c_j constraints. We propose to transform u into a modular, graphical representation. Since one constraint can involve one or multiple issues, we adopt a hypergraph representation. We assign to each c_j a factor Φ_j, with $\Phi = \{\Phi_j\}_{j=1}^{m}$. The utility hypergraph is thus defined as $G = (\mathbb{I}, \Phi)$. Nodes in $\mathbb{I}$ define the issues and the hyperedges in Φ are the factors. To each 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 constraint satisfaction problem in a standard graph. Each factor Φ_j has a sub-utility function ϕ_j defined as $\phi_j(\vec{c}) = w(c_j, \vec{c})$.

As an example, Figure 1b illustrates the utility hypergraph of $u(i_1, i_2)$ (Figure 1a), where issues are represented as white circles and the 50 constraints as red squares.

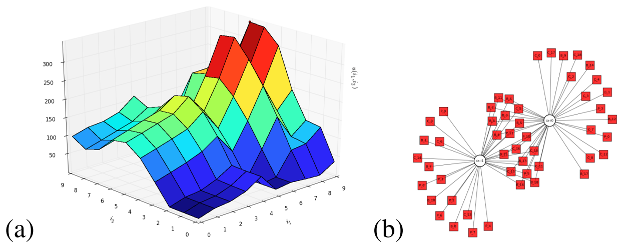

Figure 1: $2-$dimensional utility space and its hypergraph

Message passing. The hypergraph exploration is inspired from the sum-product message passing algorithm for belief propagation. However, the multiplicative algebra is changed into an additive algebra to support the utilities necessary for the assessment of the contracts. The messages circulating in the hypergraph are the contracts we are optimizing. The goal of the main algorithm (AsynchMP) is to find a bundle of optimal contracts that maximizes u. Assuming that ϕ_j is the

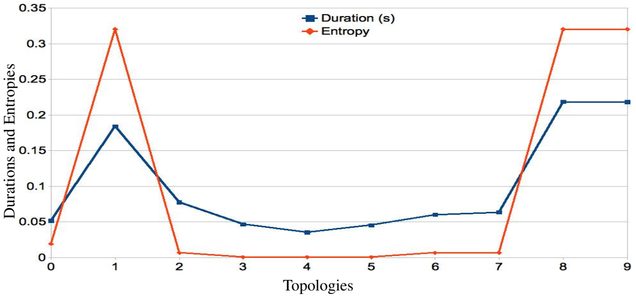

Figure 2: $H(\pi_k)$ and $\Delta(\pi_k)$ for $\pi_{k\in[0,9]} \in \{\mathcal{U}, \mathcal{D}, \mathcal{PL}\}$

sub-utility of factor Φ_j, we distinguish two types of messages: from issue i_k to Φ_j (1a), and from Φ_j to issue i_k (1b).

$$\mu_{i_k \to \Phi_j}(i_k) = \sum_{\Phi_{j'} \in \mathcal{N}(i_k) \backslash \Phi_j} \mu_{\Phi_{j'} \to i_k}(i_k) \tag{1a}$$

$$\mu_{\Phi_j \to i_k}(i_k) = \max_{i_1} \dots \max_{i_{k'} \neq k} \dots \max_{i_n} \Bigg[\tag{1b}$$

$$\phi_j(i_1, \dots, i_k, \dots, i_n) + \sum_{i_{k'} \in \mathcal{N}(\Phi_j) \backslash i_k} \mu_{i_{k'} \to \Phi_j}(i_k) \Bigg]$$

We start from the leaves of the hypergraph, *i.e.*, the issues. At $t = 0$, the content of the initial messages is defined according to $\mu_{i_k \to \Phi_j}(i_k) = 0$ and $\mu_{\Phi_j \to i_k}(i_k) = \phi'_j(i_k)$, with $\phi'_j(i_k)$ being the sub-utility of i_k in the factor Φ_j.

Complexity. We identify the parameters that could potentially affect the complexity of the utility space and thus the probability of finding optimal contract(s). We distinguish the constraint-issue connectivity, π. It refers to the number φ_j of issues involved in each c_j. Thus, a utility space (resp. hypergraph) profile is a tuple of the form (n, m, π). This parametrization will be used in the study of the complexity. If we take the distribution π to be a propagation topology, or hypergraph traversal, we can assess the corresponding complexity using information entropy (Hadfi and Ito 2013).

3 Experimental Results

When exploring G, it is possible to adopt several propagation topologies that differ in performance. Other than the initial topology defined by π, we can adopt other topologies as propagation strategies. The goal is to find the topology that minimizes the exploration cost when searching for optimal contracts. Let us consider the strategies $\{\pi_k\}_{i=0}^{9}$ that are either uniform ($\mathcal{U}$), deterministic ($\mathcal{D}$) or power-law ($\mathcal{PL}$). Their corresponding durations and complexities are illustrated in Figure 2. Both entropy (H) and duration (Δ) fluctuate similarly when describing the complexity of the underlying strategy. Besides, $\mathcal{U}$ is the most complex topology since it possesses the highest entropy and duration, as opposed to $\mathcal{D}$ and $\mathcal{PL}$.

Firstly, let us consider the general propagation mechanism (AsynchMP) that naively implements (1), and show how it outperforms the Simulated Annealing (SA) approach in (Ito, Hattori, and Klein 2007) for optimal contracts finding. For the profile $(40, [20, \dots, 100], \pi(\Phi_j) \leq 5 \forall j)$, Figure 3 shows the resulting difference in the performances.

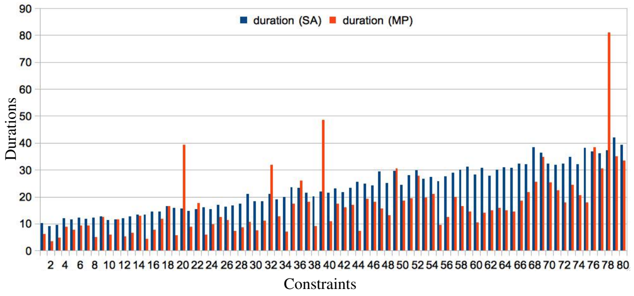

Figure 3: AsynchMP vs. SA

Now, we propose to improve AsynchMP by adopting a $\mathcal{PL}$ topology as a new propagation strategy, namely AsynchMPi, that prioritizes high degree constraints' nodes. To this end, we evaluate both AsynchMP and AsynchMPi for profiles $\{(100, 100, \pi_k)\}_{k\in[1,5]}$ with $\pi_k(\Phi_j) \leq p_k$, $p_k \in [5, 10]$ and $k \in [1, 100]$. Figure 4 shows that restricting the message passing to high degree nodes results in a drastic decrease in the duration of the search time. 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. In fact, assessing the complexity of a utility space (resp. 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 unless we consider π.

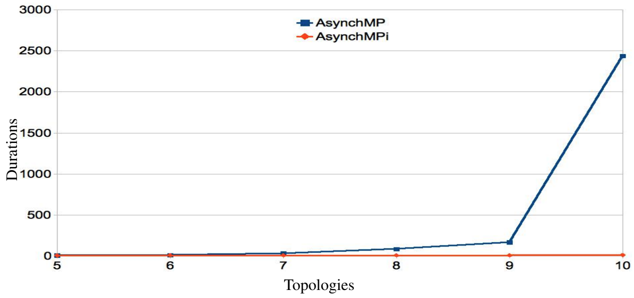

Figure 4: Duration when high degree nodes are used

References

Hadfi, R., and Ito, T. 2013. Cognition as a game of complexity. In *In Proceedings of 12th International Conference on Cognitive Modeling (ICCM)*.

Ito, T.; Hattori, H.; and Klein, M. 2007. Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, 1347–1352.

Marsa-Maestre, I.; Lopez-Carmona, M.; Carral, J.; and Ibanez, G. 2013. A recursive protocol for negotiating contracts under non-monotonic preference structures. *Group Decision and Negotiation* 22(1):1–43.