Querying User Preferences in Automated Negotiation

Adel Magra
VU Amsterdam, Netherlands
a.magra@vu.nl

Tim Baarslag
CWI, Netherlands
T.Baarslag@cwi.nl

Peter Spreij
University of Amsterdam, Netherlands
p.j.c.spreij@uva.nl

2023 IEEE International Conference on Agents (ICA) | 979-8-3503-1278-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICA58824.2023.00021

***Abstract*—An automated negotiating agent must take into account the preferences of its user to negotiate effectively. In practice, these preferences are not always fully known; therefore the agent needs to support user preference uncertainty. We present a general framework to tackle the problem of user preference uncertainty in automated negotiation. We model the user's preferences as a utility function that is unknown to the representative agent. The utility is parametrizable by a finite dimensional real vector. The agent possesses a prior belief on this parameter and can query the user for information. We are interested in determining which queries will most reduce uncertainty of the belief through what we call their information potential. We propose an optimization problem with the goal of finding a sequence of queries maximizing the information potential. We present an application of this framework to a special type of linear additive utilities defined on a multi issue negotiation domain. We establish optimal querying algorithms for this application, and experimentally assess the quality of their robust guarantees.**

I. Introduction

In many modern scenarios, automated assistance is essential for effective negotiations. Consider smart grid technologies facilitating rapid peer-to-peer energy exchanges, a pace too swift for human involvement [1]. Negotiating agents strive to secure optimal outcomes for their users, considering factors like offering strategies, deal acceptance, and opponent modeling [2], [3]. While considerable progress has been made, challenges persist in accurately representing human negotiators [4]. A critical challenge involves preference elicitation, where agents must gather user preferences to achieve favorable deals while minimizing user inconvenience. This can occur through two methods: online and offline elicitation.

Online methods, such as [5]–[9], query users to maximize expected utility during negotiations. These queries enhance negotiation outcomes by providing valuable information.

Offline methods can be passive or active. Passive approaches estimate utility using pre-negotiation information, as seen in [10] and top agents like KakeSoba and SAGA in [11]. Active offline elicitation involves querying users for preference information before negotiations, often using techniques like Conditional Preference nets (CP-nets) [12], [13].

Yet, a formal mathematical framework for optimal use of offline querying in negotiation remains absent. This paper aims to fill that gap by introducing a framework focusing on pre-negotiation uncertainty reduction. We assume user utility follows a parameterized model and agent queries reduce parameter uncertainty. We define the query's information potential, measuring worst-case uncertainty reduction, and formulate a sequential decision problem to maximize it. We apply this framework to linearly additive multi-issue negotiation domains in section III. We present OQS-n, an algorithm for optimal query sequences in simple multi-issue domains. Subsection III-B extends this to practical negotiation settings, offering optimal query sequences up to length $n-1$, where n is the issue count. These sequences yield robust uncertainty reduction guarantees. In section IV, we experimentally compare this guarantee's quality against a stochastic exploratory querying strategy.

II. The Information Framework

Given a negotiation domain Ω, we define $\mathcal{F}$ to be the set of all functions from Ω to $[0,1]$. Namely, $\mathcal{F} := \{f : \Omega \to [0,1]\}$. We make the assumption that a known subspace $\mathcal{U} \subseteq \mathcal{F}$ contains an unknown true utility of the user u^*. We call $\mathcal{U}$ the utility space. We add the assumption that $\mathcal{U}$ is parametrizable, namely that each utility u can be uniquely described by a finite vector of d parameters $\theta \in \Theta \subseteq \mathbb{R}^d$, with Θ a compact set. Let θ^* be the parameter corresponding to u^*. Furthermore, we assume the existence of a query space $\mathcal{Q}$ that corresponds to all possible queries the agent can make to the user. Each query $q \in \mathcal{Q}$ must have a finite amount of possible answers that are captured by a finite set $\mathcal{A}$. We assume that each query can be appropriately and truthfully answered by the user and we capture this idea through an answer function $a : \Theta \times \mathcal{Q} \to \mathcal{A}$. In our case, when querying the user with a query q, we would observe $a(\theta^*, q)$ which we abbreviate by $a^*(q)$. We also consider a cost function $c : \mathcal{Q} \to [0,1]$ representing the user bother associated with asking queries. Before the user is queried by the agent, we quantify the uncertainty on the user model through a probability distribution Π on Θ. We call Π the prior belief. Since we assumed truthfulness of answers, querying the user with $q \in \mathcal{Q}$ and receiving answer $a^*(q) \in \mathcal{A}$ informs us that $\theta^* \in \{\theta \in \Theta : a(\theta, q) = a^*(q)\}$. Querying has the effect of narrowing down the belief by discarding the set $\{\theta \in \Theta : a(\theta, q) \neq a^*(q)\}$. The querying cycle presented in the diagram of Figure 1 summarizes the setting of our framework.

Definition 1 (Posterior set)**.** Given a query $q \in \mathcal{Q}$ and its answer $a^*(q) \in \mathcal{A}$, we define the posterior set $\Theta_{(q,a^*(q))}$ by:

$$\Theta_{(q,a^*(q))} := \{\theta \in \Theta | a(\theta, q) = a^*(q)\}.$$

Furthermore, given a finite subset of queries $Q \subseteq \mathcal{Q}$ and their corresponding answers, we let

$$a^*(Q) := \{(q, a^*(q)) | q \in Q\}.$$

979-8-3503-1278-2/23/$31.00 ©2023 IEEE
DOI 10.1109/ICA58824.2023.00021

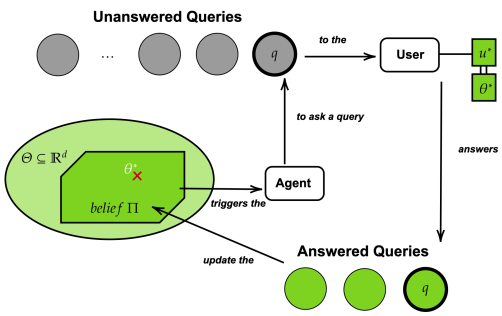

Fig. 1. The querying cycle in our framework.

In this way, we can extend the definition of posterior set for finitely many multiple queries to be:

$$\begin{aligned}\Theta_{a^*(Q)} &:= \{\theta \in \Theta | \forall q \in Q, a(\theta, q) = a^*(q)\} \\ &= \bigcap_{q \in Q} \Theta_{(q, a^*(q))}.\end{aligned}$$

The objective when querying the user is to gain the most information possible on θ^*, or equivalently to reduce the uncertainty of our initial belief on θ^* captured by Π. This means that we would like to query $q \in \mathcal{Q}$ that will significantly reduce the possibilities considered for θ^*. Namely, we want q to lead to the "smallest" possible posterior set, i.e. the smallest value of $\Pi(\Theta_{(q,a^*(q))})$. Indeed, the larger $\Pi(\Theta_{(q,a^*(q))})$, the less information we gain on θ^* by asking q. The worst case being $\Pi(\Theta_{(q,a^*(q))}) = 1$, since this happens when for example $\Theta_{(q,a^*(q))} = \Theta$, which essentially means that asking q added no information. We quantify this idea that a query contains information through what we call the information potential.

Definition 2 (Information Potential)**.** The initial information potential of a query $q \in \mathcal{Q}$ is

$$I(q) := \min_{a \in \mathcal{A}} -\log \Pi(\Theta_{(q,a)}). \tag{1}$$

More generally, given a subset of answered queries $a^*(Q)$, we define the information potential of q as

$$I(a^*(Q), q) := \min_{a \in \mathcal{A}} -\log \frac{\Pi(\Theta_{a^*(Q) \cup \{(q,a)\}})}{\Pi(\Theta_{a^*(Q)})}. \tag{2}$$

Note that $I(q) = I(\varnothing, q)$.

Lemma 1. $\forall q \in \mathcal{Q}, \forall Q \subseteq \mathcal{Q}:$

$$0 \leq I(a^*(Q), q) \leq \log |\mathcal{A}|.$$

Proof. By definition of posterior sets, we have that $\forall a \in \mathcal{A}$, $\Theta_{a^*(Q) \cup \{(q,a)\}} \subseteq \Theta_{a^*(Q)}$. It follows that:

$$\log \frac{\Pi(\Theta_{a^*(Q) \cup \{(q,a)\}})}{\Pi(\Theta_{a^*(Q)})} \leq \log \frac{\Pi(\Theta_{a^*(Q)})}{\Pi(\Theta_{a^*(Q)})} = \log 1 = 0.$$

Hence $I(a^*(Q), q) \geq 0$. For the upper bound observe that since $a^*(q) \in \mathcal{A}$, q can be answered in at most $|\mathcal{A}|$ different ways. The posterior sets corresponding to those answers form a partition of $\Theta_{a^*(Q)}$. Namely, $\Theta_{a^*(Q)}$ is equal to the disjoint union over $a \in \mathcal{A}$ of the sets $\Theta_{a^*(Q) \cup \{(q,a)\}}$. Consequently,

$$\sum_{a \in \mathcal{A}} \Pi(\Theta_{a^*(Q) \cup \{(q,a)\}}) = \Pi(\Theta_{a^*(Q)}).$$

This implies that there exists at least one answer a_0 whose posterior set must satisfy the inequality:

$$\Pi(\Theta_{a^*(Q) \cup \{(q,a_0)\}}) \geq \frac{\Pi(\Theta_{a^*(Q)})}{|\mathcal{A}|}.$$

We conclude that:

$$I(a^*(Q), q) \leq -\log \frac{\Pi(\Theta_{a^*(Q) \cup \{(q,a_0)\}})}{\Pi(\Theta_{a^*(Q)})} \leq \log |\mathcal{A}|.$$

□

Corollary 1. If the logarithm base in Definition 2 is equal to $|\mathcal{A}|$ then $I(q) \in [0, 1]$. Similarly for $Q \subseteq \mathcal{Q}$, it is true that $I(a^*(Q), q) \in [0, 1]$. We thus can assume without loss of generality that the information potential is in $[0, 1]$.

A. The Optimization Problem

At this point, it helps to picture ourselves in the agent's shoes. Initially, the queries appear to us as unactivated oracles that can provide us with information which will reduce uncertainty of our belief on θ^*. We want to be as frugal as possible in activating those oracles. It therefore makes sense that we start by activating an oracle whom we know will not disappoint. Ideally, we activate q_1, a maximizer of the information potential:

$$q_1 \in \arg\max_{q \in \mathcal{Q}} I(q).$$

Once activated, q_1 becomes obsolete. The next rational thing to do is to repeat the same process with the other queries and to stop when it becomes too costly to activate another query, or when none of the remaining queries have positive information potential. Within our framework, optimal querying can be reformulated as the following problem: Can we find a sequence of queries which maximizes the information potential? In this work, we will only consider constant cost functions c. Noting that the cost's value will only affect the length T of the query sequence, we will discard it and will assume that the maximal number of queries T to be asked is fixed. We consider the following sequential optimization problem:

Problem 1. Given $T \in \mathbb{N}$ find queries $q_1, \ldots, q_T$ such that (s.t):

$$\begin{aligned}&q_1 \in \arg\max_{q \in \mathcal{Q}} I(q) \text{ and } I(q_1) > 0, \\ &\forall t \geq 2, q_t \in \arg\max_{q \in \mathcal{Q}} I(a^*(\{q_1, \ldots, q_{t-1}\}), q) \text{ and} \\ &I(a^*(\{q_1, \ldots, q_{t-1}\}), q_t) > 0.\end{aligned}$$

III. Optimal Querying in Multi-Issue Domains

Let us now illustrate our framework with a practical application to linear additive utility functions, which are a common type of utility function considered in the automated negotiation literature (e.g. in [2], [11], [10]). To define such functions, we first require a multi issue domain Ω; namely, $\Omega = I_1 \times \ldots \times I_n$ for some $n \in \mathbb{N}$. Each element I_k of this decomposition is called an issue, which represent qualitative or quantitative elements considered in the negotiation, such as price, color, and quantity.

Definition 3 (Linear Additive Utility Function)**.** Consider the notation $[n] := \{1, \cdots, n\}$. A utility u on Ω is said to be linearly additive if $\forall k \in [n], \exists\ \theta_k \in [0,1]$ and functions $u_k : I_k \to [0,1]$ s.t $\sum_{k=1}^{n} \theta_k = 1$ and s.t $\forall\ \omega = (\omega_1, \ldots, \omega_n) \in \Omega$:

$$u(\omega) = \sum_{k=1}^{n} \theta_k \cdot u_k(\omega_k).$$

We call the θ_k's the issue weights, and for each value $\omega_k \in I_k$, we will call $u_k(\omega_k)$ the value weight of ω_k.

A. Simple Multi-Issue Domains

We apply our framework to a simple multi-issue domain and utility function that satisfy the following assumptions:

1) The issues are compact intervals: $I_i = [L_i, H_i]$ with $L_i, H_i \in \mathbb{R}$ for all $k \in [n]$.
2) The issue's utility functions $u_1^*, \ldots, u_n^*$ are strictly monotone with minimal value 0 and maximal value 1.
3) $u_1^*, \ldots, u_n^*$ are known to the agent.

The first and third assumptions are made to relate to situations where the utilities of issues follow a simple preferential scale. The second assumption shifts our focus to eliciting the weights. Note that the weights are an abstract representation for the preferences of the human user. Directly querying for their values is counterintuitive for humans, especially in large domains, as it is complex to elicit weights on many issues. A remedy to this challenge is to consider simpler queries such as outcome comparisons which are easier to preform for humans. Another approach popular in the mechanism design literature to reduce the cumbersomeness of querying is to summarize the outcomes (referred to as bids) using a bidding language, so as to simplify the preferences' representation (see [14]). We will apply our framework using the first approach. The framework components are the following:

- $\mathcal{U} := \{u_\theta(\omega) = \sum_{i=1}^{n} \theta_i \cdot u_i^*(\omega) \mid \theta_i \geq 0, \sum_{i=1}^{n} \theta_i = 1\}$ is the utility space. The true issue weights vector θ^* lives in the paramterization of $\mathcal{U}$ that is:

$$\Theta := \left\{ \theta \in \mathbb{R}^n : \forall i \in [n], \theta_i \geq 0, \sum_{i=1}^{n} \theta_i = 1 \right\}.$$

  We recognize that this is the standard $n{-}1$ simplex Δ^{n-1}.
- The query space consists of outcome comparison of questions of the type: "Do you prefer ω over ω'?". We use the notation: $(\omega \leq \omega')?$. Let then the query space be $\mathcal{Q} = \{(\omega \leq \omega')? : \omega, \omega' \in \Omega\}$. It follows that the possible answers to a query are $\mathcal{A} = \{1, 0\}$, s.t the answer function is:

$$a(\theta, (\omega \leq \omega')?) = \begin{cases} 1 \text{ if } u_\theta(\omega) \leq u_\theta(\omega'), \\ 0 \text{ if } u_\theta(\omega) > u_\theta(\omega'). \end{cases}$$

- The prior belief Π on Θ is uniform.

Let us now observe the effect of a query on $\Theta = \Delta^{n-1}$. Let q be the query comparing two arbitrary outcomes $x, y \in \Omega$, i.e. $q = ((x_1, \ldots, x_n) \leq (y_1, \ldots, y_n))?$ Suppose q is answered yes ($a^*(q) = 1$). Then we obtain:

$$\begin{aligned} u^*(x) \leq u^*(y) &\iff \sum_{i=1}^{n} \theta_i^* u^*(x_i) \leq \sum_{i=1}^{n} \theta_i^* u^*(y_i) \\ &\iff \sum_{i=1}^{n} \theta_i^* (u^*(x_i) - u^*(y_i)) \leq 0 \\ &\iff \sum_{i=1}^{n-1} \theta_i^* (d_i - d_n) + d_n \leq 0, \\ &\forall i \in [n], d_i := u_i^*(x_i) - u^*(y_i). \end{aligned} \tag{3}$$

The last inequality above describes a half-space in $\mathbb{R}^{n-1}$. If q was answered no, then we would have obtained the complement of this half-space. We therefore have that a query corresponds exactly to a hyperplane in $\mathbb{R}^{n-1}$. Furthermore, as per the domain assumptions, we have that the u_i^*'s are monotone with image $[0,1]$, meaning that they are bijective. This gives us that for every possible tuple $(a_1, \ldots, a_n)$ in $[-1,1]^n$, there exists a query q corresponding to the hyperplane $a_1\theta_1 + \ldots a_{n-1}\theta_{n-1} + a_n = 0$. This is enough to describe all hyperplanes, and so the queries are in surjection with the hyperplanes. Now, given a set of answered queries $a^*(Q)$, the posterior set $\Theta_{a^*(Q)} \subseteq \Delta^{n-1}$ is identified with its projection on $\mathbb{R}^{n-1}, \mathcal{T}(\Theta_{a^*(Q)})$. Because queries are identified with hyperplanes, we have that $\mathcal{T}(\Theta_{a^*(Q)})$ is a closed intersection of half-spaces, and hence is a convex polytope in $n-1$ dimensions. Since every query divides every posterior set in at most two parts, it follows from our previous discussion on Π, that a query with maximal information potential (equal to 1 from Corollary 1) is a query that divides $\mathcal{T}(\Theta_{a^*(Q)})$ in two parts of equal volume. Since there always exists a hyperplane that can divide a convex polytope in two parts of equal volume, we have that if $\Theta_{a^*(Q)} \neq \varnothing$, then there exists a query q satisfying $I(a^*(Q), q) = 1$.

The idea to obtain optimal queries for a domain of n issues is thus to successively bisect Δ^{n-1}, preferably along its longest edge. Such procedures exist in the Global Optimization literature and have been studied in the context of Branch & Bound algorithms where partitioning the simplex serves to refine the search space for optima ([15], [16]). Based on this idea, we present the OQS-n algorithm (Algorithm 1) to find an optimal query sequence for any number of issues n. In view of the discussion in the previous paragraph, OQS-n produces an optimal sequence of queries for Problem 1.

B. A General Optimal Algorithm

The assumptions established in section III-A are usually too strong to be satisfied. For instance, the value weights functions $u_1^*, \ldots, u_n^*$ might not satisfy the monotonicity assumptions or

Algorithm 1 OQS-n

Input: $\Omega, u_1^*, \ldots, u_n^*, T$

$P \leftarrow V(\Delta^{n-1})$ // Store the n vertices of Δ^{n-1}

for $t \in \{1, \ldots, T\}$ **do**
  $(p, q) \leftarrow \arg\max_{(p_i,p_j)\in P^2} d(p_i, p_j)$ // Longest edge
  $m \leftarrow \frac{1}{2}(p + q)$
  $\ell \leftarrow \mathcal{HP}(P\backslash\{p, q\}, m)$ // ℓ is the hyperplane bisecting P
  $q \leftarrow Query(\ell)$ // Find a query corresponding to ℓ
  $a \leftarrow Ask(q)$
  $P \leftarrow Update(q, a)$
end for

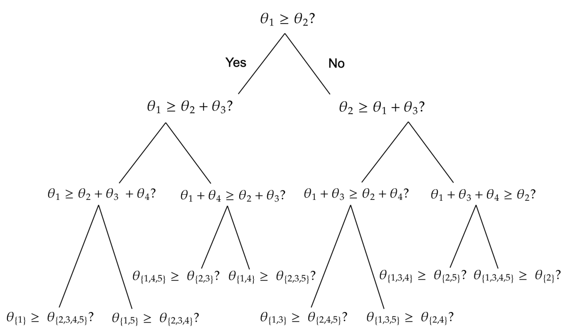

Fig. 2. Decision tree for optimal query sequence in a domain with 5 issues.

might be unknown. For this reason, we consider a restricted version of OQS-n which will require less assumptions. The negotiation domain is $\Omega = I_1 \times \ldots \times I_n$ and the unknown linear additive user utility is $u^*(\omega) = \sum_{i=1}^n \theta_i^* \cdot u_i^*(\omega_i)$. The issues $I_1, \ldots, I_n$ can be any sets, they do not have to be intervals. We only have **two assumptions**. The first one is that only the best and worst outcomes w.r.t u^* are known to the agent. Namely, the agent possesses knowledge of $\omega^{\max}$ and $\omega^{\min}$ defined as:

$$\begin{aligned} \omega^{\max} &= \arg\max_{\omega\in\Omega} u^*(\omega), \\ \omega^{\min} &= \arg\min_{\omega\in\Omega} u^*(\omega). \end{aligned} \tag{4}$$

The second assumption is that these outcomes have extreme utilities:

$$\begin{aligned} u^*(\omega^{\max}) &= 1, \\ u^*(\omega^{\min}) &= 0. \end{aligned} \tag{5}$$

These assumptions are standard in practice as it is easy to provide the best and worst outcomes for the user to its agent in almost all negotiation scenarios. Interestingly, we will see that they allow us to make a good enough use of OQS-n for reducing uncertainty on the issue weights $\theta_1^*, \ldots, \theta_n^*$. For a given set $S \subseteq [n]$, we define the negotiation outcome ω^S coordinate wise by:

$$\omega_i^S = \begin{cases} \omega_i^{\max} \text{ if } i \in S, \\ \omega_i^{\min} \text{ otherwise.} \end{cases}$$

Letting now $T \subseteq [n]$ s.t $S \cap T = \varnothing$, we obtain that the query $q = (\omega^S \geq \omega^T)?$, if answered yes, corresponds to the following restriction on u^*:

$$\begin{aligned} &\sum_{i=1}^n \theta_i^* \cdot u_i^*(\omega_i^S) \geq \sum_{i=1}^n \theta_i^* \cdot u_i^*(\omega_i^T) \\ &\iff \sum_{i\in S} \theta_i^* \geq \sum_{i\in T} \theta_i^*. \end{aligned}$$

Note that every query comparing two outcomes ω^S and ω^T corresponds to a hyperplane through the origin with normal vector whose coefficients are in $\{0, 1, -1\}^n$. For a subset S, define now:

$$\theta_S := \sum_{i\in S} \theta_i.$$

We have that $(\omega^S \geq \omega^T?)$ corresponds to the general question $(\theta_S \geq \theta_T?)$. Interestingly, for $n = 5$, the optimal query sequences of length 4 generated by OQS-5 are entirely made up of these types of questions (Figure 2). This is a consequence of the following theorem.

Theorem 2. Consider the following query space subset of $\mathcal{Q}$:

$$\mathcal{Q}_{n,1} := \{(\omega^S \geq \omega^T?) : S, T \subseteq [n], S \cap T = \varnothing\}.$$

OQS-n always selects a maximally optimal query sequence of length $n-1$ in $\mathcal{Q}_{n,1}$ for Problem 1.

The proof follows by induction on the first $n-1$ queries. It relies on the fact that the hyperplanes along which we can successively bisect Δ^{n-1} correspond to queries in $\mathcal{Q}_{n,1}$. Once answered, a query leads to a posterior set which is a convex polytope sharing at least two vertices with Δ^{n-1}. Through carefully considering all possible cases for the vertices of this polytope, it can be shown that the hyperplane bisecting along the edge formed by those two vertices always corresponds to a query in $\mathcal{Q}_{n,1}$. It is furthermore possible to explicitly derive those queries. The following corollary captures this fact. It allows us to derive an easy decision rule to describe the dynamic interaction of OQS-n with the user.

Corollary 2. The following decision rule can recreate the optimal query sequence generated by OQS-n up to length $n-1$:

$$\begin{aligned} &q_1 = (\omega^{\{1\}} \geq \omega^{\{2\}}?), \\ &\forall 2 \leq k < n-1, \text{ if } q_{k-1} = (\omega^S \geq \omega^T?), \\ &\text{then } q_k = \begin{cases} (\omega^{S\cup\{k+1\}} \geq \omega^T?) \text{ if } a^*(q_{k-1}) = 0, \\ (\omega^S \geq \omega^{T\cup\{k+1\}}?) \text{ if } a^*(q_{k-1}) = 1. \end{cases} \end{aligned} \tag{6}$$

IV. Experiments

Thanks to Corollary 2, we have determined a simple optimal sequential querying mechanism for up to $n-1$ queries. Since each successive asked query perfectly divides the previous posterior set in two parts of equal mass, we obtain a guarantee on the reduction of uncertainty on the issue weights. Namely, for $k \leq n-1$, the sequence of queries produced by OQS-n: $q_1^\circ, \ldots, q_k^\circ$ satisfies the following for any θ^*:

$$\Pi(\Theta_{a^*(\{q_1^\circ,\ldots,q_k^\circ\})}) = 2^{-k}. \tag{7}$$

It is clear that no other querying mechanism can provide as good of a guarantee because there will always exist a $\theta \in \Theta$ for which the queries will lead to a posterior set with mass greater than or equal to 2^{-k}. Nevertheless, we would like to assess the relevance of this guarantee, specifically by asking the following question: How probable is it that a smart random querying strategy yields a posterior set with Π-measure smaller than the guarantee of OQS-n? We will answer by constructing a smart random strategy and then evaluate the frequency at which it can beat the guarantee. To this end, we define for $\theta \in \Theta$ and a query sequence $q_1, \ldots, q_k$, the following quantity:

$$T_n(\theta, q_1, \ldots, q_k) := 1\{\Pi(\Theta_{a_\theta(\{q_1,\ldots,q_k\})}) \geq 2^{-k}\}.$$

The indicator $T_n(\theta, q_1, \ldots, q_k)$ is equal to 0 only if the posterior set obtained when asking $q_1, \ldots, q_k$ has smaller mass than 2^{-k} under θ being the true parameter. This essentially means that $T_n(\theta, q_1, \ldots, q_k) = 0$ if asking $q_1, \ldots, q_k$ was better than asking the optimal $q_1^\circ, \ldots, q_k^\circ$ in view of reducing uncertainty. We assume that any $\theta \in \Theta$ is equally likely of being the true parameter, that is that our belief on θ^* is represented by the random variable ϑ where $\vartheta \sim \Pi$. Taking the expectation over ϑ allows us to define the following quantity:

$$T_n(q_1, \ldots, q_k) := \mathbb{E}[T_n(\vartheta, q_1, \ldots, q_k)].$$

The quantity $T_n(q_1, \ldots, q_k)$ represents the frequency at which the strategy of asking $q_1, \ldots, q_k$ is beaten by the strategy of asking $q_1^\circ, \ldots, q_k^\circ$. Now, in the general situation of an agent that has little knowledge on the utility of its user like we considered in subsection III-B, the only queries that it should consider asking are the ones in $\mathcal{Q}_{n,1}$ as defined in Theorem 2. The size of $\mathcal{Q}_{n,1}$ grows exponentially with n and so it very quickly becomes a large set of queries in which it can be cumbersome to make a decision on the next query to ask. A legitimate strategy could then be to randomly select queries to ask to the user. Naturally, to exploit the information of the received answers, we want to avoid asking redundant queries. This corresponds to queries with 0 information potential. For a given $k \leq n-1$, the smart random strategy is summarized by the distribution $\mathcal{D}_k$ on queries that is hierarchically defined as such:

$$\begin{aligned} &q_1 \sim U(\mathcal{Q}_{n,1}), \\ &\text{and } \forall\ 2 \leq i \leq k, \\ &q_i | q_{i-1}, \ldots, q_1 \sim U\left(\mathcal{Q}_{n,1} \backslash A_i\right) \\ &\text{with } A_i := \{q : I(a^*(\{q_1, \ldots, q_{i-1}\}), q) = 0\}, \end{aligned}$$

where for a set A, $U(A)$ is the uniform distribution over A. We can then define the following quantity of interest:

$$\mu_{n,k} := \mathbb{E}_{(q_1,\ldots,q_k)\sim\mathcal{D}_k}[T_n(q_1, \ldots, q_k)].$$

The expected value $\mu_{n,k}$ represents the frequency at which the random strategy described by $\mathcal{D}_k$ will yield a posterior set of greater mass than the 2^{-k} guarantee. The larger $\mu_{n,k}$ is, the harder it is to beat the guarantee provided by OQS-n. The most simple example is the case $\mu_{2,1}$. There is no choice to be made here as $\mathcal{Q}_{2,1}$ contains only two queries equivalent to the same question, so $\mu_{2,1} = 1$. For $n = 3$, the situation is already more interesting. The non-trivial queries in $\mathcal{Q}_{3,1}$ are all equivalent to one of the following six questions:

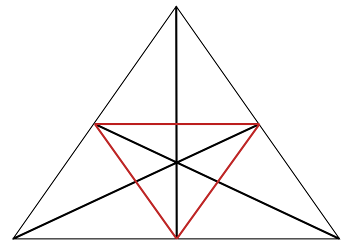

Fig. 3. The unit simplex Δ^2 and the queries of $\mathcal{Q}_{3,1}$ represented by bold and red lines.

k	$\hat{\mu}_{3,k}$	$\hat{\mu}_{4,k}$	$\hat{\mu}_{5,k}$	$\hat{\mu}_{6,k}$	$\hat{\mu}_{7,k}$	$\hat{\mu}_{8,k}$
1	$\hat{\mu}_{3,1} = 0.88 \pm 0.020$	$\hat{\mu}_{4,1} = 0.86 \pm 0.022$	0.85 ± 0.022	0.84 ± 0.023	0.84 ± 0.023	0.84 ± 0.023
2	$\hat{\mu}_{3,2} = 0.77 \pm 0.026$	$\hat{\mu}_{4,2} = 0.75 \pm 0.027$	0.74 ± 0.028	0.76 ± 0.026	0.77 ± 0.026	0.78 ± 0.026
3		$\hat{\mu}_{4,3} = 0.76 \pm 0.026$	0.77 ± 0.026	0.79 ± 0.025	0.80 ± 0.025	0.80 ± 0.025
4			0.80 ± 0.026	0.82 ± 0.024	0.82 ± 0.024	0.83 ± 0.023
5				0.83 ± 0.023	0.84 ± 0.023	0.85 ± 0.022
6					0.86 ± 0.022	0.87 ± 0.021
7						0.90 ± 0.019

TABLE I
ESTIMATES OF $\mu_{n,k}$ FOR DIFFERENT VALUES OF n AND k WITH 95% CONFIDENCE INTERVALS. EACH ESTIMATE WAS THE AVERAGE OVER 1000 I.I.D SAMPLES.

$$\begin{aligned} &\theta_1 \geq \theta_2?, \theta_2 \geq \theta_3?, \theta_1 \geq \theta_3? \\ &\theta_1 \geq \theta_2 + \theta_3?, \theta_2 \geq \theta_1 + \theta_3?, \theta_3 \geq \theta_1 + \theta_2? \end{aligned} \tag{8}$$

We can visualize this scenario in Figure 3. The black bold lines represent queries corresponding to questions in the first line of equation (8) while the red ones correspond to the ones in the second line. To compute $\mu_{3,1}$ we proceed as follows: Fix $\theta \in \Theta$, that is take any point inside the triangle of Figure 3. We distinguish two cases:

- θ is in the middle triangle delimited by the red lines. In this case, asking any red query will lead to a posterior set of mass $3/4$, while any black query will lead to a posterior set of mass $1/2$. We therefore have that none of the 6 queries will beat the $1/2$ guarantee given by OQS-n.
- θ is in one of the 3 triangles formed by a red line and the half of two sides of Δ^2. In this case, the red query corresponding to this red line will lead to a posterior set of mass $1/4$ while the 2 other red queries will yield a posterior set of mass $3/4$. The black queries will lead to a posterior set of mass $1/2$. Therefore, only 1 of the 6 queries beats the $1/2$ guarantee.

The middle triangle has mass $1/4$, and so taking expectations over Θ gives us that:

$$\mu_{3,1} = \frac{1}{4} \cdot \frac{6}{6} + \frac{3}{4} \cdot \frac{5}{6} = \frac{7}{8}.$$

In Table I, we present estimates $\hat{\mu}_{n,k}$ of $\mu_{n,k}$ for n ranging from 3 to 8 and k from 1 to $n-1$. We observe that all of the estimates for $\mu_{n,k}$'s are quite high (above 0.7), telling us that a random strategy only rarely beats the 2^{-k} guarantee on the reduction of uncertainty given by OQS-n. We see that for a fixed k, the estimates $\hat{\mu}_{n,k}$ are roughly equal for different values of n, whereas for a fixed n, $\hat{\mu}_{n,k}$ increases with k for

$k \geq 2$. This seems to indicate that it gets increasingly more challenging to do better than OQS-n at reducing uncertainty as we consider longer query sequences. This may be due to the fact that in the large pool of possible query sequences, very few actually have a chance of capturing a given fixed θ more accurately than OQS-n. For instance, the red triangle of Figure 3 was a "dead zone" because no queries were able to give a smaller posterior set if θ was inside it. Such dead zones contribute to making $\mu_{n,k}$ larger and so it may be that they are more numerous as we consider longer query sequences. These estimates comfort us in the idea that OQS-n is not only a robust method of querying, but also that it rarely pays off to adopt another strategy to reduce uncertainty.

V. Conclusion and Discussion

Our main contribution is to provide a general framework to deal with the problem of user uncertainty in automated negotiation. We consider a situation where the true utility is assumed to come from a parametric family and where the agent is allowed to query the user for information about this true utility. By introducing the notion of information potential of a query, we derive an optimal sequential problem in Problem 1. An optimal querying mechanism is then one that maximizes the information potential of queries. This gives us an objective way to quantify what it means to optimally reduce uncertainty in an offline active elicitation setting. We apply the framework to a specific form of utilities called linear additive utilities. Based on successive bisections of the standard unit simplex, we derive OQS-n, an optimal querying algorithm which provides a maximal guarantee on the reduction of uncertainty. OQS-n is optimal in our framework provided that the prior belief Π on the true parameter is uniformly distributed on the unit simplex. We relax the strong assumptions that OQS-n relies on to be able to produce an optimal query sequence of arbitrarily length and consider assumptions applicable to most negotiations on multi-issue domains. Namely, we show that under the sole knowledge of the best and worst outcomes, OQS-n can generate maximally optimal query sequences up to length $n-1$. We then demonstrate experimentally that the guarantee on the reduction of uncertainty provided by OQS-n can hardly be beaten by using a random exploratory querying strategy.

Our contribution is to provide a formal manner of using offline active elicitation to tackle the problem of user uncertainty in automated negotiation. Nevertheless, the relative importance of reducing user uncertainty still needs to be put to the test with respect to the performance of the agent. A future direction to evaluate this could be to conduct a meta analysis of negotiating agents by treating the elicitation mechanism of an agent as a component just like the acceptance strategy or the opponent model, and to assess how modifying those components affects the performance of the agent. Such work has been done previously but without considering elicitation as a component of the agent [17]. We have nonetheless established a framework that can be used in future works to make objective formal statements about reduction of uncertainty in automated negotiations under user uncertainty.

References

[1] F. L. de Alba, A. González-Briones, P. Chamoso, T. Pinto, Z. Vale, and J. M. Corchado, "A p2p electricity negotiation agent systems in urban smart grids," in *Distributed Computing and Artificial Intelligence, Special Sessions, 17th International Conference*, S. Rodríguez González, A. González-Briones, A. Gola, G. Katranas, M. Ricca, R. Loukanova, and J. Prieto, Eds. Cham: Springer International Publishing, 2021, pp. 97–106.

[2] T. Baarslag, K. Fujita, E. Gerding, K. Hindriks, T. Ito, N. Jennings, C. Jonker, S. Kraus, R. Lin, V. Robu, and C. Williams, "Evaluating practical negotiating agents: Results and analysis of the 2011 international competition," *Artificial Intelligence*, vol. 198, p. 73–103, 05 2013.

[3] T. Baarslag, M. Hendrikx, K. Hindriks, and C. Jonker, "Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques," *Autonomous Agents and Multi-Agent Systems*, 09 2016.

[4] T. Baarslag, M. Kaisers, E. H. Gerding, C. M. Jonker, and J. Gratch, "When will negotiation agents be able to represent us? the challenges and opportunities for autonomous negotiators," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, ser. IJCAI'17. AAAI Press, 2017, p. 4684–4690.

[5] T. Baarslag and M. Kaisers, "The value of information in automated negotiation: a decision model for eliciting user preferences," 05 2017.

[6] Y. Mohammad and S. Nakadai, *FastVOI: Efficient Utility Elicitation During Negotiations: 21st International Conference, Tokyo, Japan, October 29-November 2, 2018, Proceedings*, 10 2018, pp. 560–567.

[7] T. Baarslag and E. Gerding, "Optimal incremental preference elicitation during negotiation," 01 2015.

[8] H. Leahu, M. Kaisers, and T. Baarslag, "Preference learning in automated negotiation using gaussian uncertainty models," 03 2019.

[9] ——, "Automated negotiation with gaussian process-based utility models," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 421–427. [Online]. Available: https://doi.org/10.24963/ijcai.2019/60

[10] D. Tsimpoukis, T. Baarslag, M. Kaisers, and N. Paterakis, *Automated Negotiations Under User Preference Uncertainty: A Linear Programming Approach*, 04 2019, pp. 115–129.

[11] R. Aydoğan, T. Baarslag, K. Fujita, J. Mell, J. Gratch, D. de Jonge, Y. Mohammad, S. Nakadai, S. Morinaga, H. Osawa, C. Aranha, and C. M. Jonker, "Challenges and main results of the automated negotiating agents competition (anac) 2019," in *Multi-Agent Systems and Agreement Technologies*, N. Bassiliades, G. Chalkiadakis, and D. de Jonge, Eds. Cham: Springer International Publishing, 2020, pp. 366–381.

[12] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole, "Cp-nets: A tool for representing and reasoning withconditional ceteris paribus preference statements," *Journal of Artificial Intelligence Research*, vol. 21, p. 135–191, Feb 2004. [Online]. Available: http://dx.doi.org/10.1613/jair.1234

[13] R. Aydoğan, T. Baarslag, K. Hindriks, C. Jonker, and P. Yolum, "Heuristics for using cp-nets in utility-based negotiation without knowing utilities," *Knowledge and Information Systems*, vol. 45, no. 2, pp. 357–388, Nov. 2015.

[14] N. Nisan, "Bidding languages," http://www.eecs.harvard.edu/cs286r/courses/fall09/papers/bidding.pdf, 2009.

[15] J. Žilinskas, "Branch and bound with simplicial partitions for global optimization," *Mathematical Modelling and Analysis*, vol. 13, pp. 145–159, 03 2008.

[16] G. Aparicio, L. G. Casado, E. Hendrix, B. G.-Tóth, and I. García Fernandez, "On the minimum number of simplex shapes in longest edge bisection refinement of a regular n-simplex," *Informatica*, vol. 26, pp. 17–32, 03 2015.

[17] T. Baarslag, "What to bid and when to stop," Ph.D. dissertation, 09 2014.