Regret-based Utility Elicitation in Constraint-based Decision Problems

Craig Boutilier
Dept. of Computer Science
University of Toronto
cebly@cs.toronto.edu

Relu Patrascu
Dept. of Computer Science
University of Toronto
relu@cs.toronto.edu

Pascal Poupart
School of Computer Science
University of Waterloo
ppoupart@cs.uwaterloo.ca

Dale Schuurmans
Dept. of Computing Science
University of Alberta
dale@cs.ualberta.ca

Abstract

We propose new methods of preference elicitation for constraint-based optimization problems based on the use of minimax regret. Specifically, we assume a constraint-based optimization problem (e.g., product configuration) in which the objective function (e.g., consumer preferences) are unknown or imprecisely specified. Assuming a graphical utility model, we describe several elicitation strategies that require the user to answer only binary (bound) queries on the utility model parameters. While a theoretically motivated algorithm can provably reduce regret quickly (in terms of number of queries), we demonstrate that, in practice, heuristic strategies perform much better, and are able to find optimal (or near-optimal) configurations with far fewer queries.

1 Introduction

The development of automated decision software is a key focus within decision analysis [21; 16] and AI [7; 3]. To deal with different users, some form of preference elicitation must be undertaken in order to capture specific user preferences to a sufficient degree to allow an (approximately) optimal decision to be taken. In this work, we study the problem of preference elicitation in *constraint-based optimization (CBO)*. CBO provides a natural framework for specifying and solving many decision problems, such as configuration tasks [15], in which hard constraints capture options available to a customer and an objective (or utility) function reflects customer preferences. Explicit formulation as a mathematical program or using a soft constraint framework [17; 2] has been successfully used to model such problems.

Unfortunately, the requirement of complete utility information demanded by CBO is often problematic. For instance, users may have neither the ability nor the patience to provide full utility information to a system. Furthermore, in many if not most instances, an optimal decision (or some approximation thereof) can be determined with a very partial specification of the user's utility function. As such, it is imperative that preference elicitation procedures be designed that focus on the relevant aspects of the problem (e.g., by ignoring infeasible parts of utility space, or utility for outcomes provably dominated given current information).

Our framework is as follows. We assume a set of (hard) constraints together with a graphical utility model [1; 4] capturing user preferences. While the structure of the utility model is known, the parameters of this utility model are imprecise, given by upper and lower bounds. Adopting the minimax regret model of [5], a robust decision can be made with respect to this utility uncertainty, by choosing the *minimax optimal configuration*. This is the solution the user would regret the least should an adversary impose a utility function consistent with our knowledge of the user's preferences. If regret is unacceptably high, we query the user for more information about their preferences, until the worst-case error (regret) is small enough (zero if optimality is required).

In this work, we elicit preferences using *bound queries* (a local form of *standard gamble queries* [9])—that provide tighter upper or lower bounds on the utility parameters—since these are reasonably easy for users to assess and have been studied extensively in the decision analysis literature. We develop several new strategies for elicitation using bound queries whose aim is to reduce the worst-case error (i.e., get guaranteed improvement in decision quality) with as few queries as possible. These include strategies with good theoretical guarantees (related to polyhedral methods in conjoint analysis [18; 11]), as well as several heuristic methods that perform better empirically. We also show that one of these strategies is largely unaffected by computational approximation of the required minimax solutions.

2 Problem Formulation

We assume some system is charged with making or recommending a decision on behalf of a user, for example, configuring a (multiattribute) product for a consumer (e.g., the choice of a car and options). However, since user preferences vary, an appropriate configuration requires that the system interact with the user to determine enough about her preferences over feasible alternatives to make a good (or possibly optimal) choice. Thus, the system must engage in some form of *preference elicitation*. This basic problem lies at the heart of considerable work in multiattribute utility theory [12; 20; 14] and the theory of consumer choice (such as conjoint analysis [18; 11]). Our approach differs from classic approaches in several important respects, as we emphasize below. Most importantly, our decision model and elicitation strategies will be driven by the minimax regret criterion. To present our results, we first need to review previous work on minimax regret and the framework of [5] in more detail.

2.1 Optimization with Graphical Utility Models

Following [5], we assume a finite set of *attributes* $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ with finite domains, characterizing a set of choices available to a decision maker (or consequences thereof). These might be, say, car options such as *make*, *engine size*, *transmission type*, etc. An assignment $\mathbf{x} \in Dom(\mathbf{X})$ is often referred to as a *state*. For ease of presentation, we assume attributes are boolean. We also have a set of *hard constraints* $\mathcal{C}$ over these attributes. Each constraint $\mathcal{C}_\ell$, $\ell = 1, ..., L$, is defined over a subset of attributes $\mathbf{X}[\ell] \subset \mathbf{X}$, and thus induces a set of legal configurations of attributes in the subset $\mathbf{X}[\ell]$ (e.g., the set of products that can be proposed). We assume that the constraints $\mathcal{C}_\ell$ are represented in some logical form and can be expressed compactly (e.g., $X_1 \wedge X_2 \supset \neg X_3$). For instance, we might have that model *Passat* and engine *2.8T* do not allow transmission *5Speed-Man*. We let *Feas*$(\mathbf{X})$ denote the subset of *feasible states* satisfying $\mathcal{C}$.

Suppose the system had access to the user's utility function $u : Dom(\mathbf{X}) \rightarrow \mathbf{R}$, representing a user's strength of preference for various configurations (e.g., we might view this as what they are willing to pay). The constraint-based optimization (CBO) problem is to find an optimal feasible state $\mathbf{x}^*$:

$$\mathbf{x}^* \in \arg \max_{\mathbf{x} \in \textit{Feas}(\mathbf{X})} u(\mathbf{x}).$$

For multiattribute problems of this type, one generally assumes some utility function structure [12; 8]. In this work, we adopt a *generalized additive independence (GAI)* model [8; 1]. Specifically, we assume that the utility function can be written as the sum of K local utility functions, or *factors*, over small sets of variables:

$$u(\mathbf{x}) = \sum_{k \leq K} f^k(\mathbf{x}[k]). \qquad (1)$$

Here each function f^k depends only on a local family of attributes $\mathbf{X}[k] \subset \mathbf{X}$. We denote by $\mathbf{x}[k]$ the restriction of state $\mathbf{x}$ to the attributes in $\mathbf{X}[k]$. This model is attractive due to its naturalness and generality (encompassing both linear models [12] and UCP-nets [4] as special cases). The problem of finding an optimal configuration can be formulated as an integer program (IP), and can often be solved very effectively.

2.2 Minimax Regret

While many approaches to elicitation focus on obtaining full utility information, it will often be infeasible, undesirable, or unnecessary to extract a complete utility function from the user [20]. As a consequence, a system must make decisions in the face of *incompletely* specified utility functions. Several approaches have been proposed for representing utility uncertainty, as well as making decisions given this uncertainty. For example, Bayesian methods quantify uncertainty about preferences probabilistically [7; 3]. Other techniques simply pose constraints on the set of possible utility functions and use various criteria to find or reduce the set of decisions or otherwise direct elicitation; for example, one can identify Pareto optimal [21] or dominant alternatives [20; 13; 14], or decisions that minimize regret [4; 16; 19; 5].

In this paper we adopt the minimax regret criterion, following the formulation of [5]. Unlike Bayesian methods, minimax regret recommends decisions given only a *set* of possible utility functions rather than a probabilistic prior. Thus it is suitable when reasonable priors are hard to formulate and does not require the computational approximations often needed in reasoning with complex priors [7; 3]. Minimax regret also allows recommendation of decisions that are robust in the face of utility function uncertainty since it provides tight bounds on the worst-case error, which is appealing in many circumstances (and unlike many schemes proffers a specific decision rather than, say, a Pareto optimal set). Previously this has been found to be an attractive criterion in real-world procurement settings [6]. As we will see, regret is also very effective in focusing elicitation effort even when priors are available [19].

Suppose the utility function for a CBO problem is unknown, but constraints on its parameters (e.g., in the form of bounds) are available and some decision must be recommended.[1] The minimax regret decision criterion suggests that one adopt the (feasible) assignment $\mathbf{x}$ that obtains minimum max-regret, where max-regret is the largest quantity by which one could "regret" choosing $\mathbf{x}$ (while allowing the utility function to vary within the bounds). More formally, let $\mathcal{U}$ denote the set of feasible utility functions. We refer to a pair $\langle \mathcal{C}, \mathcal{U} \rangle$, where $\mathcal{C}$ is a set of configuration constraints, as an *imprecise CBO problem*. The *pairwise regret* of state $\mathbf{x}$ with respect to $\mathbf{x}'$ over feasible utility set $\mathcal{U}$ is defined as

$$R(\mathbf{x}, \mathbf{x}', \mathcal{U}) \;=\; \max_{u \in \mathcal{U}} \{u(\mathbf{x}') - u(\mathbf{x})\}, \qquad (2)$$

which is the most our system could regret choosing $\mathbf{x}$ for the user instead of $\mathbf{x}'$ (e.g., if an adversary could impose any utility function in $\mathcal{U}$). The *maximum regret* of decision $\mathbf{x}$ is:

$$\textit{MR}(\mathbf{x}, \mathcal{U}) \;=\; \max_{\mathbf{x}' \in \textit{Feas}(\mathbf{X})} R(\mathbf{x}, \mathbf{x}', \mathcal{U}) \qquad (3)$$

The *minimax regret* of feasible utility set $\mathcal{U}$ is:

$$\textit{MMR}(\mathcal{U}) \;=\; \min_{\mathbf{x} \in \textit{Feas}(\mathbf{X})} \textit{MR}(\mathbf{x}, \mathcal{U}) \qquad (4)$$

If the only information we have about a user's utility function is that it lies in the set $\mathcal{U}$, then a minimax-optimal decision $\mathbf{x}^*$ (i.e., $\mathbf{x}^*$ s.t. $\textit{MR}(\mathbf{x}^*, \mathcal{U}) = \textit{MMR}(\mathcal{U})$) minimizes the worst-case loss w.r.t. possible realizations of utility $u \in \mathcal{U}$.

Computation of minimax regret in CBO problems requires care; the explicit minimization in Eq. 4 is infeasible. Fortunately, one can formulate it in a manner that exploits the graphical structure of the utility model, thereby admitting (in practice) computationally tractable solution [5]. The procedure of [5] assumes an imprecise CBO problem with factors f^k, $k \leq K$, defined over local families $\mathbf{X}[k]$. The parameters of this utility function are denoted by $u_{\mathbf{x}[k]} = f^k(\mathbf{x}[k])$, where $\mathbf{x}[k]$ ranges over $Dom(\mathbf{X}[k])$. Upper and lower bounds on each of these parameters are assumed, denoted by $u_{\mathbf{x}[k]}\!\uparrow$ and $u_{\mathbf{x}[k]}\!\downarrow$, respectively. Effective computation of pairwise regret, max regret and minimax regret is possible by exploiting structure in the constraints and graphical utility model.

[1] These constraints reflect knowledge of the user's utility function, generally obtained through elicitation as we elaborate below.

Input: imprecise CBO problem, worst-case error tolerance τ.

1. Compute minimax regret value *mmr*
2. Repeat until $mmr < \tau$
   (a) Ask bound query q about some utility parameter $u(\mathbf{x}[k])$.
   (b) If $u(\mathbf{x}[k]) \leq q$ then lower $u_{\mathbf{x}[k]}\!\uparrow$ to q.
   (c) Otherwise raise $u_{\mathbf{x}[k]}\!\downarrow$ to q
   (d) recompute *mmr*

Table 1: General form of the interactive elicitation procedure.

In particular, the minimax optimization in Eq. 4 is rewritten as as minimization with an infinite number of constraints. A constraint generation procedure is used to generate constraints incrementally until all (finitely many) active constraints are enumerated. The mixed integer programs (MIPs) required to solve both the main minimization and the generation of the most violated constraint are compact, with a number of variables linear in the size (number of parameters) of the GAI model. An important property of this procedure is that it generates both an optimal solution $\mathbf{x}^*$ and an adversarial *witness* $\mathbf{x}^w$ for the current $\mathcal{U}$: $\mathbf{x}^w$ is the assignment that the maximizes regret of $\mathbf{x}^*$ in Eq. 4 (as the $\mathbf{x}'$ variable in Eq. 3). We refer to [5] for further algorithm details. The procedure was shown to handle significant practical problems, generally generating very few constraints, with solution times ranging from fractions of a second to 1000 seconds [5].

In practice, since minimax regret will be computed between elicitation queries, it is critical that minimax regret be estimated in a relatively short period of time (say 5 seconds). For this reason, we propose several improvements to the procedure of [5] that can speed up regret computation with elicitation in mind: (1) The constraint generation procedure for solving the MIP can be accelerated by simply finding a feasible $\mathbf{x}$ given the current set of constraints, rather than explicitly searching for a minimax optimal $\mathbf{x}$ given current constraints. (2) Since minimax regret is computed incrementally by generating constraints, it has an anytime nature and early stopping can be used. This has the effect that some violated constraints may not have been generated, but the "early" solution provides a lower bound on true minimax regret. We can also compute an upper bound by computing the max regret of the $\mathbf{x}$ found for the last MIP solved. These bounds are often tight enough to provide good elicitation guidance. (3) The minimax regret problem solved after receiving a response to one query is very similar to that solved before posing the query. As such, one can "seed" the minimax procedure invoked after a query with the constraints generated at the previous step. In this way, typically, only a few extra constraints are generated during each minimax computation.

While we focus on the use of upper and lower bounds on utility parameters, the procedures described here can be adapted to problems with arbitrary linear constraints over utility parameters. Handling such constraints is important when *comparison queries* are used (see below). With this background in place, we can now turn to elicitation.

3 Elicitation Strategies

We consider an interactive process in which the decision software queries the user for information about her utility function, refining initial bounds on the parameters, until minimax regret reaches an acceptable level τ.[2] Table 1 summarizes the general form of the interactive elicitation procedure.

3.1 Bound Queries

The types of queries we consider are *bound queries* in which we ask the user whether one of her utility parameters lies above a certain value. A positive response raises the lower bound on that parameter, while a negative response lowers the upper bound: in both cases, uncertainty is reduced.[3]

While users often have difficulty assessing numerical parameters, they are typically better at comparing outcomes [12; 9]. However, a bound query can be viewed as a local form of a *standard gamble query (SGQ)*, commonly used in decision analysis; these, in fact, ask for comparisons. An SGQ for a specific state $\mathbf{x}$ asks the user if she prefers $\mathbf{x}$ to a gamble in which the best outcome $\mathbf{x}_\top$ occurs with probability l and the worst $\mathbf{x}_\bot$ occurs with probability $1-l$ [12]. A positive response puts a lower bound on the utility of $\mathbf{x}$, and a negative response puts an upper bound. Calibration is attained by the use of common best and worst outcomes across all queries (and numerical assessment is restricted to evaluating probabilities). The foundations of such queries can be made precise using results of Fishburn [8]; we defer details for space reasons (see [10] for more on elicitation in GAI networks).[4]

Our query strategies rely on the following definitions.

Defn 1 Let $\langle \mathcal{C}, \mathcal{U} \rangle$ be an imprecise CBO problem. An *optimistic state* $\mathbf{x}^o$, a *pessimistic state* $\mathbf{x}^p$, and a *most uncertain state* $\mathbf{x}^{mu}$ are any states satisfying:

$$\mathbf{x}^o \in \arg\max_{\mathbf{x}\in Feas(\mathbf{X})} \max_{u\in\mathcal{U}} u(\mathbf{x})$$
$$\mathbf{x}^p \in \arg\max_{\mathbf{x}\in Feas(\mathbf{X})} \min_{u\in\mathcal{U}} u(\mathbf{x})$$
$$\mathbf{x}^{mu} \in \arg\max_{\mathbf{x}\in Feas(\mathbf{X})} \max_{u,u'\in\mathcal{U}} \{u(\mathbf{x}) - u'(\mathbf{x})\}$$

An optimistic state is a feasible state with the greatest upper bound on utility. A pessimistic state has the greatest lower bound on utility. A most uncertain state has the greatest difference between its upper and lower bounds. Each of these states can be computed in a single optimization by setting the parameters of the utility model to their upper bounds, their lower bounds, or their difference, and solving the corresponding (precise) CBO problem.

In this framework, the goal of an elicitation strategy is to reduce minimax regret using as few queries as possible. The challenge is to select such queries efficiently—avoiding intractabilities inherent in outcome enumeration and lookahead, while nevertheless reducing minimax regret effectively.

[2]We could insist that regret reaches zero (i.e., a provably optimal solution), or stop when regret reaches a point where further improvement is outweighed by the cost of further interaction.

[3]If the user's true value is close to the query point, she may feel "roughly indifferent;" in this case we could impose a constraint that the true value is "close" (e.g., within some ε) to this point.

[4]While we focus on bound queries, other queries are quite natural. Comparison queries ask if one state $\mathbf{x}$ is preferred to another $\mathbf{x}'$. A response imposes a linear constraint on utility parameters. Regret computation must then take the general form alluded to above.

3.2 The Halve Largest Gap Strategy

The first query strategy we consider is the *halve largest gap (HLG)* strategy. It asks a query at the midpoint of the largest interval, i.e., corresponding to the parameter $\mathbf{x}[k]$ with the largest gap between its upper and lower bounds. This is motivated by theoretical considerations based on simple worst-case bounds on minimax regret. Define, respectively, the *gap* of a utility parameter $u(\mathbf{x}[k])$, the *span* of factor f^k and *maxspan* of a utility model $\mathcal{U}$ as follows:[5]

$$\mathit{gap}(\mathbf{x}[k]) = u_{\mathbf{x}[k]}\!\uparrow - u_{\mathbf{x}[k]}\!\downarrow \tag{5}$$

$$\mathit{span}(f^k) = \max_{\mathbf{x}[k] \in Dom(\mathbf{X}[k])} \mathit{gap}(\mathbf{x}[k]) \tag{6}$$

$$\mathit{maxspan}(\mathcal{U}) = \sum_k \mathit{span}(f^k) \tag{7}$$

The quantity *maxspan* measures the largest gap between the upper and lower utility bound, regardless of feasibility. We can show that this quantity bounds minimax regret:

Proposition 1 *For any $\langle \mathcal{C}, \mathcal{U} \rangle$, MMR$(U) \leq$ maxspan$(\mathcal{U})$.*

Since $\mathit{MMR}(U) \leq \mathit{MR}(\mathbf{x}^o, \mathcal{U})$ and for any optimistic state $\mathbf{x}^o$ we have $\mathit{MR}(\mathbf{x}^o, \mathcal{U}) \leq \mathit{maxspan}(\mathcal{U})$, the result follows.[6]

This suggests an obvious query strategy, the HLG method, in which a bound query is asked of the parameter p with the largest gap, at the midway point of its interval, $(p\!\uparrow - p\!\downarrow)/2$. This method ensures rapid reduction in max regret:

Proposition 2 *Let $\mathcal{U}$ be an uncertain utility model with n parameters and let m = maxspan$(\mathcal{U})$. After $n \log(m/\varepsilon)$ queries by HLG, minimax regret is no greater than ε.*

In the worst case, no query strategy can reduce regret more quickly than HLG. Furthermore, there are classes of utility functions for which the bound is tight, so worst-case $\mathcal{U}$ and configuration constraints $\mathcal{C}$ exist that ensure regret will never be reduced to zero in finitely many queries.[7] This strategy is similar to heuristically motivated polyhedral methods in conjoint analysis used in product design and marketing [18; 11]). In fact, HLG can be viewed as a special case of the method of [18] in which our polyhedra are hyper-rectangles.

3.3 The Current Solution Strategy

While HLG allows one to provide strong worst-case guarantees on regret improvement, it is “undirected” in that considerations of feasibility play no role in determining which queries to ask. An alternative strategy is to focus attention on parameters that participate in defining the max regret, namely, the minimax optimal $\mathbf{x}^*$ and the adversarial witness $\mathbf{x}^w$ for the current $\mathcal{U}$ (recall that the witness maximizes the regret of $\mathbf{x}^*$). The *current solution (CS) query strategy* asks about the utility parameter in the set $\{\mathbf{x}^*[k] : k \leq K\} \cup \{\mathbf{x}^w[k] : k \leq K\}$ with largest *gap*$(\mathbf{x}[k])$ and queries the midpoint of the corresponding utility interval. Intuitively, should the answer to a query raise the lower bound on some $u(\mathbf{x}^*[k])$ or lower the upper bound on some $u(\mathbf{x}^w[k])$, then the pairwise regret $R(\mathbf{x}^*, \mathbf{x}^w)$ will be reduced, and usually minimax regret as well. Of course, if the answer lowers the upper bound on some $u(\mathbf{x}^*[k])$ or raises the lower bound on some $u(\mathbf{x}^w[k])$, then pairwise regret $R(\mathbf{x}^*, \mathbf{x}^w)$ remains unchanged. (Note that minimax regret might still be reduced.)

We have also experimented with a variant of the CS strategy in which regret is computed approximately to ensure fast interactive response by imposing a time bound on computation (as described above). While we can’t be sure we have the minimax optimal solution with early termination, the solution may be good enough to guide the querying process. Furthermore, since we can compute the max regret of the anytime solution, we know the quality of the solution, and we have an upper bound on minimax regret which can be used as a natural termination criterion for the querying process.

3.4 Alternative Strategies

Finally, we have experimented with several other strategies, which we describe briefly. The *optimistic query strategy* computes an optimistic state $\mathbf{x}^o$ and queries (at the midpoint of the interval) the utility parameter in $\mathbf{x}^o$ with the largest gap. Intuitively, an optimistic $\mathbf{x}^o$ is a useful adversarial choice, so refining information about it can help reduce regret. The *pessimistic query strategy* is analogous, relying on the intuition that a pessimistic choice is useful in preventing the adversary from making us regret our decision too much. The *optimistic-pessimistic (OP) strategy* combines the two: it queries the parameter that has the largest gap among both states. These strategies are computationally appealing since they require only standard CBO, not minimax optimization.

The *most uncertain state (MUS) strategy* is a variant of HLG that accounts for feasibility: we compute a most uncertain state $\mathbf{x}^{mu}$ and query (at the midpoint) the parameter in $\mathbf{x}^{mu}$ with the largest gap. Finally, the *second-best (SB) strategy* is based on the following intuition: suppose we have the optimistic state $\mathbf{x}^o$ and the second-best optimistic state $\mathbf{x}^{2o}$ (i.e., that state with the second-highest upper bound—this is computable with a single optimization). If we could ask a query which reduced the upper bound utility of $\mathbf{x}^o$ to lower than that of $\mathbf{x}^{2o}$, we ensure that regret is reduced (since the adversary can no longer attain this most optimistic value); if the lower bound of $\mathbf{x}^o$ were raised to the level of $\mathbf{x}^{2o}$’s upper bound, then we could terminate—knowing that $\mathbf{x}^o$ is *optimal*. Thus we would like to query $\mathbf{x}^o$ at $\mathbf{x}^{2o}$’s upper bound: a negative response will reduce regret, a positive response ensures $\mathbf{x}^o$ is optimal. The SB method “distributes” this query across the relevant factor parameters, asking several bound queries.

The *myopically optimal (MY) strategy* computes the average regret reduction of the midpoint query for *each* utility parameter by solving the minimax problem for each response to each query; it then asks the query with the largest average reduction. This approach is generally infeasible, but we test

[5] We denote the upper and lower bounds of any parameter p by $p\!\uparrow$ and $p\!\downarrow$ respectively.

[6] The definition of *maxspan* can be tightened to account for logical consistency of the assignments to different factors, or by restricting attention to feasible states (w.r.t. $\mathcal{C}$). The result still holds.

[7] The bound is not generally tight if there is overlap in factors; but is tight if *maxspan* accounts for logical consistency.

it on small problems to see how the other methods compare.[8]

4 Empirical Results

We tested the effectiveness of our query strategies on a variety of problems. For each problem we tested: halve largest gap (HLG), current solution (CS), current solution with a computation-time bound of five seconds per query (CS-5), optimistic-pessimistic (OP), second-best (SB), and most uncertain state (MUS). We also compared these against the more computationally demanding myopically optimal method (MY) on small problems. We implemented the constraint generation approach of [5] and used CPLEX as the generic IP solver. Our experiments considered two realistic domains—car rentals and real estate—as well as randomly generated synthetic problems.

First, we tested small synthetic problems to allow a comparison of our heuristics with the MY strategy. Fig. 1 reports the average minimax regret over 45 small synthetic problems constructed by randomly setting the utility bounds and the variables on which each utility factor depends. Each problem has 10 attributes that can take at most 4 values and 10 factors that depend on at most 3 attributes. We simulate user responses by drawing a random utility function u for each trial, consistent with the bounds, representing a user's preferences. Responses to queries are generated using u (assuming the user accurately answers queries relative to u). Results are shown for two cases: utility parameters drawn from a uniform distribution over each interval, and those drawn from a (truncated) Gaussian centered at the midpoint of the interval (reflecting that a user is somewhat more likely to have a true value near the middle of the range).[9] In both cases, we observe that the OP, CS and CS-5 heuristics decrease minimax-regret at a rate very close to MY. This suggests that OP, CS and CS-5 are computationally feasible, yet promising alternatives to the computationally prohibitive MY strategy.

We report on further experiments using all strategies except MY with larger synthetic problems, a real-estate problem and a car rental problem taken from [5], drawing users from uniform distributions (Gaussian results are very similar both in shape and magnitude). As above, all results are averaged over 45 trials. The car-rental problem is modeled with 26 (multivalued) variables that specify various attributes of a car relevant to typical rental decisions, resulting in 61,917,360,000 possible configurations. The factored utility model consists of 36 local factors, each defined on at most five variables, with 150 utility parameters. Performance of the various query strategies is depicted in Fig. 2(a). Initial utility bounds are set to give minimax regret of roughly 18% of the optimal solution. Both CS and CS-5 perform extremely well: regret is reduced to almost zero within 160 queries on average.[10] More importantly, these methods show excellent anytime performance: after only 80 queries, average minimax regret has dropped from 18% to under 2%. Interestingly, the time bound of 5 seconds imposed by CS-5, while leading to approximately minimax optimal solutions, does not affect query quality: the approximate solutions give rise to queries that are virtually as effective as the optimal solutions. The CS strategy requires on average at most 83s per query. The OP strategy works very well too, and requires less computation time (0.1s per query) since it does not need to solve minimax problems (except to verify termination "periodically," which is not reflected in query computation time). However, both OP and CS-5 are fast enough to be used interactively on problems of this size. MUS, HLG, and SB do not work nearly as well. Note the HLG performs poorly since it fails to account for the feasibility of options, thus directing its attention to parts of utility space for which no product exists (hence polyhedral methods alone [18; 11] will not offer reasonable elicitation in our setting). MUS significantly outperforms HLG for just this reason.

The real-estate problem is modeled with 20 (multivalued) variables, with 47,775,744 possible configurations. The factored utility model consists of 29 local factors, giving rise to 100 utility parameters. Query performance is shown in Fig. 2(b), using the same regime as above. Again, both CS and CS-5 perform best, and the time bound of CS-5 has no effect on the quality of the CS strategy. Interestingly, OP performs almost identically to these, with somewhat lower computational cost (CS takes 14s/query, CS-5 5s, and OP 0.1s). Each of these methods reduces minimax regret from 40% of optimal to under 5% in about 120 queries. As above, SB fails to make progress, while HLG and MUS do somewhat better. We also tested the query strategies on larger randomly generated problems (with 25 variables of domain size no more than four, and 20 utility factors with no more than three variables each). The same performance patterns as in real-estate emerge, with CS, CS-5 and OP all performing much better than the others (see Fig. 2(c)). Although OP performs slightly better than CS/CS-5, the difference is not significant.

5 Concluding Remarks

We have developed a number of query strategies for eliciting bounds on the parameters of graphical utility models for the purpose of solving imprecise constraint-based optimization problems. The most promising of these strategies, CS and OP, perform extremely well, requiring very few queries (relative to the model size) to provide dramatic reductions in regret. We have shown that using approximation of minimax regret reduces interactive computation time to levels required for real-time response without a noticeable effect on the performance of CS. OP also can be executed in real-time, since it does not require the same intensive minimax computation.

We plan to extend this research a number of directions. We would like to consider additional query types, such as comparisons of outcomes and tackle the associated computational

[8] By doing lookahead of this type for k stages, we could in fact compute the optimal query plan of k-steps.

[9] All experiments show a reasonably small variance so we exclude error bars for legibility.

[10] Though this may seem like a lot of queries, recall that the problem is very large, with a utility model with 150 parameters. We intentionally choose problems this large to push the computational boundaries of regret-based elicitation. Furthermore, while 160 queries may be large for typical consumer choice problems, it is more than reasonable for high stakes configuration applications.

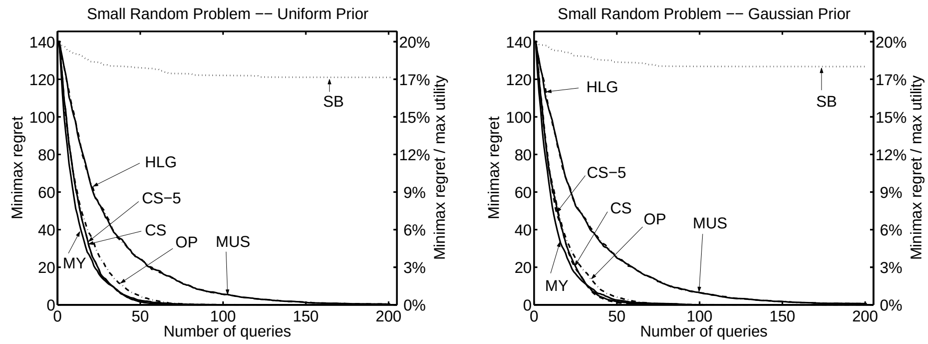

Figure 1: Avg. max regret on small random problems (45 instances) as a function of number of queries given (a) uniform and (b) Gaussian distributed utilities.

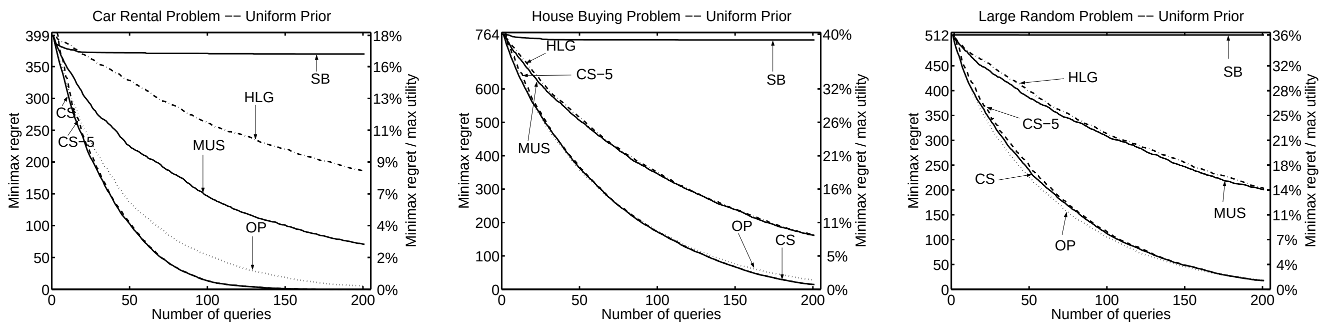

Figure 2: Avg. max regret (45 instances, uniform density) as a function of number of queries: (a) car; (b) real estate; (c) large random problems.

problems. We also plan to explore new query strategies, including those that exploit probabilistic information to compute queries yet still use (distribution-free) regret to make decisions [19]. Non-myopic models are also of considerable interest. Finally, human factors must be addressed, including framing and presentation, and dealing with the fact that users often "construct" preferences during analysis, rather than "reveal" existing preferences—this can often lead to inconsistency. While our current approach will never ask a query for which a response could be inconsistent with prior responses, allowing a user to backtrack still may be important.

Acknowledgements: This research was supported by the the Institute for Robotics and Intelligent Systems (IRIS) and the Natural Sciences and Engineering Research Council (NSERC). Poupart was supported by a scholarship provided by Precarn Incorporated through IRIS.

References

[1] F. Bacchus and A. Grove. Graphical models for preference and utility. *UAI-95*, pp.3–10, Montreal, 1995.

[2] S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based constraint satisfaction and optimization. *JACM*, 44:201–236, 1997.

[3] C. Boutilier. A POMDP formulation of preference elicitation problems. *AAAI-02*, pp.239–246, Edmonton, 2002.

[4] C. Boutilier, F. Bacchus, and R. I. Brafman. UCP-Networks: A directed graphical representation of conditional utilities. *UAI-01*, pp.56–64, Seattle, 2001.

[5] C. Boutilier, R. Patrascu, P. Poupart, D. Schuurmans. Constraint-based optimization with the minimax decision criterion. *CP-2003*, pp.168–182, Kinsale, 2003.

[6] C. Boutilier, T. Sandholm, and R. Shields. Eliciting bid taker non-price preferences in (combinatorial) auctions. *AAAI-04*, pp.204–211, San Jose, CA, 2004.

[7] U. Chajewska, D. Koller, and R. Parr. Making rational decisions using adaptive utility elicitation. *AAAI-00*, pp.363–369, Austin, TX, 2000.

[8] P. C. Fishburn. Interdependence and additivity in multivariate, unidimensional expected utility theory. *Intl. Econ. Rev.*, 8:335–342, 1967.

[9] S. French. *Decision Theory*. Halsted, New York, 1986.

[10] C. Gonzales and P. Perny. GAI networks for utility elicitation. *KR-04*, pp.224–234, Whistler, BC, 2004.

[11] V. S. Iyengar, J. Lee, and M. Campbell. Q-Eval: Evaluating multiple attribute items using queries. *ACM EC-01*, pp.144–153, Tampa, FL, 2001.

[12] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. Wiley, NY, 1976.

[13] C. W. Kirkwood, R. K. Sarin. Ranking with partial information: A method and an application. *Op. Res.*, 33:38–48, 1985.

[14] A. M. Mármol, J. Puerto, F. R. Fernández. The use of partial information on weights in multicriteria decision problems. *J. Multicrit. Dec. Anal.*, 7:322–329, 1998.

[15] D. Sabin and R. Weigel. Product configuration frameworks—a survey. *IEEE Intel. Sys. and their Appl.*, 13(4):42–49, 1998.

[16] A. Salo and R. P. Hämäläinen. Preference ratios in multiattribute evaluation (PRIME)–elicitation and decision procedures under incomplete information. *IEEE SMC*, 31(6):533–545, 2001.

[17] T. Schiex, H. Fargie, and G. Verfaillie. Valued constraint satisfaction problems: Hard and easy problems. *IJCAI-95*, pp.631–637, Montreal, 1995.

[18] O. Toubia, J. Hauser, and D. Simester. Polyhedral methods for adaptive choice-based conjoint analysis. TR 4285-03, Sloan School of Mgmt., MIT, 2003.

[19] T. Wang and C. Boutilier. Incremental utility elicitation with the minimax regret decision criterion. *IJCAI-03*, pp.309–316, Acapulco, 2003.

[20] M. Weber. Decision making with incomplete information. *Eur. J. Op. Res.*, 28:44–57, 1987.

[21] C. C. White, III, A. P. Sage, and S. Dozono. A model of multiattribute decisionmaking and trade-off weight determination under uncertainty. *IEEE SMC*, 14(2):223–229, 1984.