Mixed Integer Linear Programming Under Preference Uncertainty

Bhavik A. Shah

Raul Astudillo

Peter I. Frazier

We introduce mixed integer linear programming under preference uncertainty, a novel approach for supporting decision-making when the decision-maker's preferences are uncertain. Here, each feasible design is associated with a vector of attributes, and each vector is assigned a utility through the decision-maker's utility function, which has been incompletely estimated through preference learning, and whose remaining uncertainty is quantified by a Bayesian prior or posterior probability distribution. We develop new optimization-based algorithms with theoretical guarantees that provide a menu of diverse solutions among which the decision maker is likely to find a solution that performs well according to her preferences.

1. Introduction

As motivation, consider the following example, which marries preference learning/utility elicitation (Braziunas 2006, Abbas 2018) with classical multi-criterion mixed integer linear programming.

Consider a project between a local hospital and an external academic operations research (OR) team. The OR team's goal is to help the hospital administration create a weekly schedule assigning nurses to hospital wards and shifts, to be revised on an ongoing basis. The schedule should satisfy staffing levels while also prioritizing other attributes: ensuring that nurses get sufficient rest between shifts; satisfying soft constraints expressed by individual nurses on days when they are unavailable to work (understanding that not all soft preferences can be met); and distributing shifts on holidays and weekends fairly across the workforce. Moreover, the schedule should be able to be updated by a software tool that takes new information about required staffing levels in each ward, a changing set of nurses and their capabilities, and a changing set of requests from nurses for days off.

Suppose the hard constraints (minimum staffing levels in each ward; a nurse cannot work in two wards at the same time; a nurse can only work in wards for which he or she has the experience and skills to work) can be easily formulated as linear constraints, and each attribute can be formulated as a linear function of the decision variables. Let C be a known matrix such that Cx is a vector of attributes corresponding to any feasible solution x satisfying hard constraints $Ax \leq b$. Then, if we could find weights θ quantifying the hospital administration's preferences over attributes, we could simply solve a mixed integer linear program (MILP) to give an optimal schedule: $\max \theta^\top Cx$ subject to $Ax \leq b$. Unfortunately, however, it is unclear what weights to use.

If the attributes are all quantities that we want to be either as large or as small as possible, one approach is to use multi-objective linear programming: we solve $\max \theta^\top Cx$ subject to $Ax \leq b$ for all possible θ. This produces a Pareto frontier, which the hospital can then examine to select the schedule that has the attributes they most like.

Unfortunately, if there are more than 2 or 3 attributes, then it becomes difficult to visualize the Pareto frontier and select the best option from among it (Benson 1998). This is particularly problematic if each of 100 nurses enters requests for days off, which makes the Pareto frontier more than 100-dimensional. Also, when attributes are high-dimensional, we need to solve an exponentially large number of MILPs, which becomes computationally prohibitive. While this might be feasible for a single week's schedule, if a new schedule must be recomputed each week based on new input, then exploring the full Pareto frontier each week could be onerous.

Another approach is for the external OR team to engage in a sequence of meetings with the local hospital where an optimal schedule is found for many different θ and the hospital administration is asked to select among them the schedule they like the best. The corresponding θ that generated it can then be taken as providing a point estimate of the objective function. This can then be held fixed and used by the hospital in a software tool to generate schedules in future weeks. In this setting, we refer to $U(Cx;\theta) = \theta^\top Cx$ as the decision-maker's (DM's) utility function and $\widehat{\theta}$ our estimate of it.

This approach, however, ignores uncertainty in our estimate of θ. This uncertainty may arise due to noise in the responses or because it is based on feedback over a finite number of schedules: there may have been multiple θ that would have all corresponded to preferring the chosen schedule over the others offered. It may also arise because preferences change over time: in one week we may choose not to fulfill an employee's request for taking a particular day off because that employee has recently had a particularly desirable schedule, but then later on we may wish to prioritize requests from that employee more heavily. This approach also restricts choice, in that it offers only a single schedule to the DM in subsequent weeks.

A better approach would be to offer a collection of options, designed for several different likely θ, to allow the DM to choose. This would offer more value across a range of possible DM's preferences. Suppose, for example, we knew that the DM had a preference specified by one of finitely many weights, $\theta_1, \ldots, \theta_L$. Then, we could simply solve the MILP for each such θ_ℓ and offer the set of solutions as a menu, to allow the DM to choose. This, however becomes computationally infeasible and also practically overwhelming for the user as L grows large. To address this challenge, we design here a novel approach targeted specifically to mixed integer linear programming problems problems with many attributes. We build on recent work by the authors Astudillo and Frazier (2020) that focused on Bayesian optimization with many attributes.

This approach uses a similar sequence of conversations with the DM described above for calculating a point estimate $\widehat{\theta}$ and instead estimates a Bayesian posterior probability distribution over θ. Then, we solve a higher-level optimization problem that chooses a *menu* of solutions (schedules, in the example above) to the DM who selects his or her most preferred solution. He can either implement this solution or we can incorporate this choice into our probability distribution over θ to generate a new menu. In this way, we can quickly drive toward finding the DM's most preferred design on the Pareto frontier with much less computation while also requiring much less effort from the human DM. Offering a menu dramatically outperforms optimizing based on a single point estimate. By leveraging probabilistic information about the DM's preferences, our method is more efficient than standard multi-objective linear programming.

2. Literature Review

Within the broader field of multi-criterion optimization (Ehrgott 2005), multi-criterion combinatorial optimization (Ehrgott and Gandibleux 2000), and vector optimization (Jahn 2009), the most closely related work is on multi-objective mixed integer programs (Zionts 1979, Clímaco et al. 1997, Alves and Clímaco 2007).

A focus in this area is on efficiently computing points on the Pareto frontier. Multi-objective LPs have convex feasible regions, allowing points on the Pareto frontier to be recovered by minimizing a linear function of the attributes (Isermann 1974). Enumerating the Pareto frontier for multi-objective MILPs is more complex because their feasible regions are non-convex: some Pareto optimal points may be dominated by (infeasible) convex combinations of feasible points (Alves and Clímaco 2007). Methods for efficiently generating all or parts of a multi-objective MILP's Pareto frontiers are considered in Zionts (1979), Alves and Clímaco (2007), and De Loera et al. (2009).

Perhaps the most closely related work in this literature studies *interactive* decision support for multi-objective LPs and MILPs, surveyed in Teghem and Kunsch (1986) and Alves and Clímaco (2007). Within this literature, the most closely related are *implicit utility function* methods. These methods assume like ours that the DM has an implicit utility function. Like the interactive version of our method, these so-called implicit utility function methods learn from pairwise comparisons successively smaller sets of implicit utility functions consistent with the DM's responses, while also identifying successively smaller portions of the Pareto frontier that are optimal for implicit utility functions in this class. Implicit utility function methods restricted to biobjective problems include Aksoy (1990), Ramesh et al. (1990), and Shin and Allen (1994). More general implicit utility function methods appropriate for problems with more than two objectives include Gonzalez et al. (1985), Marcotte and Soland (1986), White (1985), and Lokman et al. (2016).

Our method is fundamentally different from these past methods through the fact that it builds a Bayesian machine learning model over the implicit utility function (rather than just a set of possible utility functions) and that it leverages this to build a menu with high expected utility. This fact allows our method to offer four key improvements over past methods in this space.

First, past methods require noise-free perfectly consistent responses from the DM. While our numerical experiments focus on noise-free responses, our menu selection can leverage Bayesian posteriors over preferences that can be learned from noisy inconsistent responses.

Second, most past methods make heavy use of linear utility functions while our methods allow general utility functions. One exception is Lokman et al. (2016), which allows quasiconcave utility functions.

Third, past methods start from a clean slate each time a new problem or a new DM is confronted. In contrast, we can leverage the full power of Bayesian preference learning to leverage data from past interactions with the same DM on similar problems, or other DMs on the same problem. For example, in the motivating example from the introduction, from schedules chosen in past weeks we might learn that DMs consistently give more preference to day-off-requests from nurses who have been denied in past weeks. This would allow us to automatically provide menus of solutions with this property, honing in on good solutions faster.

Fourth, our algorithm is flexible: it can be used without any interaction to provide just a single menu with high expected utility, it can be used over just one or two rounds of interaction, or it can be used many time to hone in on precisely one optimal solution.

Another line of work related to ours is classical utility elicitation (Abbas 2018). The classical utility elicitation setting involves asking queries to the decision-maker that help to determine properties of her utility function and, ultimately, to obtain an accurate estimate of it, which can be then used for decision-making (Keeney 1977). In practice, however, this type of elicitation process is typically too demanding in terms of time and cognitive effort, specially in the presence of large design spaces or multiple attributes, and thus often a substantial amount of utility uncertainty remains. As mentioned earlier, our approach overcomes this challenge by acknowledging uncertainty in the DM's preferences.

More recently, due to the increasing interest in having artificial intelligence support decision-making and sometimes even make decisions on behalf of users, utility elicitation (sometimes also referred as preference elicitation) has also been studied within the computer science community (Chajewska et al. 1998, 2000, Boutilier 2002, Boutilier et al. 2006). Unlike in the classical utility elicitation setting, the aim of most of this work has been to develop algorithms that elicit the decision maker's utility function (or preferences) with the goal of finding a good decision given the utility uncertainty that remains, instead of trying to fully mitigate this uncertainty (Braziunas 2006). To the best of our knowledge, all these methods assume that the design space is finite, and thus they cannot be applied in our setting.

3. Problem Setup

We assume that the space of designs is represented by a convex polytope, $\mathbb{X} = \{x \in \mathbb{R}^I : Ax \leq b\}$, possibly intersected with $\mathbb{Z}^{I'} \times \mathbb{R}^{I-I'}$, where $A \in \mathbb{R}^{K \times I}$ and $b \in \mathbb{R}^K$, and attributes are given by a linear function, $f(x) = Cx$, where $C \in \mathbb{R}^{J \times I}$. We also assume that there is a DM whose preference over designs is characterized by a design's attributes, y, through a linear utility function, $U(y;\theta) = \theta^\top y$, where θ is the vector of weights representing her preferences. Thus, of all the designs, the DM most prefers one in the set $\arg\max_{x \in \mathbb{X}} U(f(x);\theta)$. If θ were known to us, we could apply single-objective mixed integer optimization to maximize $U(f(x);\theta)$. Instead, we assume that θ is unknown, and that we have access only to a probability distribution over θ, p, which may be obtained by a brief and incomplete utility elicitation exercise with the DM, and/or from choices made by this or other similar DMs in the past in problem contexts with the same attributes.

Our goal is to provide the DM with a menu of M designs so as to maximize the expected value of the utility obtained when she selects the best design in this menu according to her underlying utility function. Formally, we wish to solve

$$\max_{x_1,\ldots,x_M \in \mathbb{X}} \mathbb{E}\left[\max_{m=1,\ldots,M} U(f(x_m);\theta)\right] \tag{1}$$

$$\max_{x_1,\ldots,x_M \in \mathbb{X}} \sum_{\theta \in \Theta} p(\theta) \max_{m=1,\ldots,M} U(f(x_m);\theta) \tag{2}$$

With this method of finding a menu, we can then show the DM their options. Once the DM selects the best option from the menu, we use the classical preference learning / utility elicitation technique in which we ask the DM which option in the menu is most preferred. We can then use standard preference learning methods to form a posterior distribution from this preference information, and then repeat the process as many times as necessary until the DM is satisfied with the menu options.

4. Solution Approaches

We now introduce several approaches for generating menus to show a DM. For each of these approaches, we prove theoretical guarantees. The proofs of these results can be found in the appendix.

4.1. The Optimal Menu

In this section we describe an approach to solve Problem 1 by formulating it as a mixed integer linear program. Momentarily, we assume that the support of θ is finite; the end of this section, however, discusses how to relax this assumption.

Problem 1 is equivalent to the mixed integer linear program

$$\begin{aligned} \max_{w,x,z} \quad & \sum_{\theta\in\Theta} p(\theta) \sum_{m=1}^{M} w_{\theta,m} \\ \text{subject to} \quad & \sum_{m=1}^{M} z_{\theta,m} = 1 : \theta \in \Theta \\ & z_{\theta,m} \in \{0,1\} : \theta \in \Theta,\ m = 1,\ldots,M \\ & w_{\theta,m} \le (u_\theta - l_\theta) z_{\theta,m} : \theta \in \Theta,\ m = 1,\ldots,M \\ & w_{\theta,m} \le \theta^\top C x_m - l_\theta : \theta \in \Theta,\ m = 1,\ldots,M, \end{aligned} \tag{3}$$

where l_θ and u_θ are any constants satisfying $l_\theta \le U(f(x);\theta) \le u_\theta$ for all $x \in \mathbb{X}$. To see this equivalence, note that the first three constraints imply that, for each $\theta \in \Theta$, at most one $w_{\theta,m}$ is nonzero. Therefore, for any given $x_1,\ldots,x_M$, and for each $\theta \in \Theta$, the maximum achievable value of the sum $\sum_{m=1}^{M} w_{\theta m}$ is $\max_{m=1,\ldots,M} \theta^\top C x_m - l_\theta$, i.e., $\max_{m=1,\ldots,M} U(f(x_m);\theta) - l_\theta$. Thus, if $x_1,\ldots,x_M$ are optimal for Problem 3, then they are also optimal for Problem 1.

As mentioned earlier, the above approach assumes that θ has a finite support. However, this is often not true in practice. Moreover, even if θ has a finite support, Problem 1's dimensionality grows linearly with the support of θ, and thus it is desirable to consider approximations that are more computationally tractable in large-support regimes. To this end, we consider the following approach to approximately solve problem. Instead of using the full support of θ, this approach draws L i.i.d. samples from p_θ, $\theta_1,\ldots,\theta_L$, and replaces Problem 1 by

$$\max_{x_1,\ldots,x_M\in\mathbb{X}} \frac{1}{L} \sum_{\ell=1}^{M} \max_{m=1,\ldots,M} U(f(x_m);\theta_\ell). \tag{4}$$

The above approach is known in the literature as *sample average approximation (SAA)* (Kim et al. 2015). Despite the simplicity of this approach, we show that it possesses appealing theoretical guarantees. Theorem 1 below guarantees, for example, that any solution of Problem 2 converges almost surely to a solution of Problem 1 as L grows large.

Theorem 1. *Suppose that $\mathbb{X}$ is compact, and let v^* and $\widehat{v}^*(L)$ be the optimal values of Problems 1 and 4, respectively. Also let $X^* = \arg\max_{x_1,\ldots,x_M \in \mathbb{X}} \mathbb{E}\left[\max_{m=1,\ldots,M} U(f(x_m);\theta)\right]$ be the set of optimal solutions of Problem 1 and $\widehat{x}^*(L) = (\widehat{x}_1^*(L),\ldots,\widehat{x}_M^*(L))$ be any optimal solution of Problem 4. Then, $\widehat{v}^*(L) \to v^\star$ and $\operatorname{dist}(\widehat{x}^*(L), X^*) \to 0$ almost surely as $L \to \infty$.*

Next, we show that, under a mild additional regularity condition, any solution of Problem 4 not only converges almost surely to a solution of Problem 1 but this convergence is actually exponential. More specifically, we show the following.

Theorem 2. *Suppose that $\mathbb{X}$ is compact and that the moment generating function of $\|\theta\|_2$, $t \mapsto \mathbb{E}\left[e^{t\|\theta\|_2}\right]$ is finite in an open neighborhood of 0. Then, for all $\delta > 0$, there exist $G > 0$ and $\gamma > 1$ such that $\mathbb{P}(\operatorname{dist}(\widehat{x}^*(L), X^*) > \delta) \leq Ge^{-\gamma L}$.*

4.2. A Submodular Function Maximization Approach

The algorithm discussed in this section is inspired by the key observation that the function $V : 2^{\mathbb{X}} \to \mathbb{R}$ defined by $V(X) = \mathbb{E}[\sup_{x \in X} U(f(x);\theta)]$ is a monotone submodular set function. As we shall see later, this implies that a simple greedy algorithm has a $1 - 1/e$ approximation guarantee to the optimal solution. Before describing this algorithm in detail, we formally define what a monotone submodular function is.

Definition 1. *A set valued function $V : 2^{\Omega} \to \mathbb{R}$ is said to be monotone if $A \subset B \subset \Omega$ implies $V(A) \leq V(B)$; and V is said to be submodular if $V(A) + V(B) \geq V(A \cup B) + V(A \cap B)$ for all $A, B \subset \Omega$.*

Formally, we consider the algorithm that builds the menu to be shown to the DM iteratively as follows. For $N = 1, \ldots, M$ and having chosen the first $(N-1)$-th items of the menu, the N-th item of the menu, x_N^*, is chosen as an optimal solution of

$$\max_{x_N \in \mathbb{X}} \mathbb{E}\left[\max\left\{\max_{m=1,\ldots,N-1} U(f(x_m^*);\theta), U(f(x_N);\theta)\right\}\right], \tag{5}$$

i.e., we choose the best possible item given the items that we have chosen so far.

As mentioned earlier, this simple algorithm has an appealing approximation guarantee. This guarantee is formally stated in Theorem 3 below, whose proof follows directly from the classical result in Nemhauser et al. (1978).

Theorem 3. *Suppose that $x_1^*, \dots, x_M^*$ are iteratively chosen as*

$$x_N^* \in \arg\max_{x_N \in \mathbb{X}} \mathbb{E}\left[\max\left\{\max_{m=1,\dots,N-1} U(f(x_m^*);\theta), U(f(x_N);\theta)\right\}\right], \; N = 1,\dots,M.$$

Then,

$$\mathbb{E}\left[\max_{m=1,\dots,M} U(f(x_m^*);\theta)\right] \ge \left(1 - \frac{1}{e}\right) \max_{x_1,\dots,x_M \in \mathbb{X}} \mathbb{E}\left[\max_{m=1,\dots,M} U(f(x_m);\theta)\right].$$

Problem 5 can be formulated as a MILP in an analogous way as we did before with Problem 1. By leveraging the the fact that it is enough to distinguish whether the new item to be included menu is better than previous ones, this MILP reformulation can be simplified to

$$\begin{aligned} \max_{x_N} \; & \sum_\theta p(\theta)((1 - z_\theta)U_{N-1}^*(\theta) + w_\theta) \\ \text{subject to } & w_\theta \le (u_\theta - l_\theta) z_\theta : \theta \in \Theta \\ & w_\theta \le U(f(x_i);\theta) - l_\theta : \theta \in \Theta \\ & z_{\theta,n} \in \{0,1\} : \theta \in \Theta \end{aligned} \tag{6}$$

where $U_{N-1}^*(\theta) = \max_{m=1,\dots,N-1} U(f(x_m);\theta)$ and, as before, l_θ and u_θ are any constants satisfying $l_\theta \le U(f(x);\theta) \le u_\theta$ for all $x \in \mathbb{X}$. As with the previous approach, we can use SAA to approximately solve the above problem when the support of θ is too large.

4.3. A Thompson Sampling Inspired Approach

Thompson sampling is an algorithm for sequential decision-making under uncertainty (Russo et al. 2018), known for balancing exploitation and exploration. In each step, it selects the design to evaluate, x^*, by drawing a sample from the probability of it being the optimum. In our setting, a sample from this distribution, x^*, can be obtained by drawing a sample θ from $p(\theta)$, and choosing $x^* \in \arg\max_{x \in \mathbb{X}} U(f(x);\theta)$. (This will make our proposed Thompson-sampling-based algorithm easy-to-implement in practice if one already has code for solving a single-objective version of the desired MILP.)

While our setting is not sequential, we take inspiration from this algorithm and consider the following strategy to build the menu of designs to be shown to the DM: let $\theta_1, \ldots, \theta_M$ be iid samples from p_θ; the menu to be shown is chosen as $\{x_1^*, \ldots, x_M^*\}$, where $x_m^* \in \arg\max_{x \in \mathbb{X}} U(f(x); \theta_m)$.

Drawing parallels with the exploitation and exploration balance that Thompson sampling provides in the sequential setting, a menu built under the approach described above is likely to contain designs that are both high quality individually and diverse when considered jointly. In our experiments, we show that this simple algorithm performs well in practice. Moreover, we show that a menu built following this approach achieves an optimal performance exponentially fast as the size of the menu grows. More specifically, we prove the two theorems below. Theorem 4 show that this convergence holds when the true utility function is in the support of the prior distribution, and Theorem 5 show the same when the true utility function drawn from the prior distribution.

Theorem 4. *Suppose that $\mathbb{X}$ is compact and that $t_0 \in \Theta$ is such that $\mathbb{P}(\|\theta - t_0\|_2 < \delta) > 0$ for all $\delta > 0$. Also suppose that $\theta_1, \ldots, \theta_M \overset{iid}{\sim} p_\theta$ and let $x^*(\theta_m) \in \arg\max_{x \in \mathbb{X}} U(f(x); \theta_m)$, $m = 1, \ldots, M$. Then, for any $\epsilon > 0$,*

$$\mathbb{P}\left(\max_{x \in \mathbb{X}} U(f(x); t_0) - \max_{m=1,\ldots,M} U(f(x^*(\theta_m)); t_0) > \epsilon\right)$$

converges to 0 exponentially fast as $M \to \infty$.

Theorem 5. *Suppose that both $\mathbb{X}$ and Θ are compact, and let $\theta_0, \theta_1, \ldots, \theta_M \overset{iid}{\sim} p_\theta$, $x^*(\theta_m) \in \arg\max_{x \in \mathbb{X}} U(f(x); \theta_m)$, $m = 1, \ldots, M$. Further suppose that the following two conditions hold*

1. *The set $\Theta' = \{t \in \Theta : \mathbb{P}(\|\theta - t\|_2 < \delta) > 0 \text{ for all } \delta > 0\}$ has probability 1 under p_θ.*

2. *For any $\delta > 0$, $\inf_{t \in \Theta'} \mathbb{P}(\|\theta - t\|_2 < \delta) > 0$.*

Then, for any $\epsilon > 0$,

$$\mathbb{P}\left(\max_{x \in \mathbb{X}} U(f(x); \theta_0) - \max_{m=1,\ldots,M} U(f(x^*(\theta_m)); \theta_0) > \epsilon\right)$$

converges to 0 exponentially fast as $M \to \infty$.

We note that, while conditions 1 and 2 in Theorem 2 seem rather technical, they hold in two standard settings: when the distribution over θ has finite support, and when it admits a strictly positive density over Θ.

5. Numerical Experiments

We conduct two numerical experiments to asses the performance of our algorithms and showcase some practical applications of our framework.

5.1. Experimental Setups

We consider three different experimental setups. The first two are static and the third one is dynamic, in the sense that preference information is gathered from the DM to update the distribution on θ.

Setup 1: The first setup uses a prior on θ whose support is small enough to be fully included into the scenario sets for the optimal menu and submodular approaches. To evaluate an algorithm for a given menu size in this setting, we do the following:

1. We draw a small number of preference scenarios $\theta_1, \ldots, \theta_L$ uniformly at random from the probability simplex. Then we set the prior distribution on θ as uniform over $\Theta = \{\theta_1, \ldots, \theta_L\}$.

2. We provide the point estimate, submodular, and optimal menu algorithms the entire set of preference scenarios, Θ. Thompson sampling samples independently (with replacement) from Θ.

3. For each preference scenario $\theta \in \Theta$, we calculate the utility of the best item in the offered menu, and then take the average across all preference scenarios in Θ. This provides the expected utility of the offered menu under the prior.

4. We then average this expected utility across many replications of this process to get an unbiased estimate of an algorithm's quality. This averages over uncertainty about both the preferences of the DM and also the scenario set used.

Setup 2 The second setup uses a prior probability distribution on θ with infinite support: the uniform distribution over the simplex. It evaluates an algorithm as follows:

1. We use the algorithm to generate a menu. For algorithms requiring a small scenario set (optimal menu, submodular) we subsample a scenario set of a specified size uniformly from the prior (uniform

over the probabilty simplex) and the algorithm approximates the prior by assuming it is uniform over that scenario set.

2. We then sample a large number of new preference scenarios independently from the prior (probability simplex). We then follow the same methodology from steps (3) and (4) from Setup 1 over many replications to generate an unbiased estimate of the expected utility.

We plot the expected utility versus menu size (*Setup 2a*) and versus the the number of preference scenarios used in the optimal menu and submodular approaches (*Setup 2b*).

To reduce variance in comparisons, we induce some correlation in both the preference scenario sets and evaluation sets used: (1) We use the same scenario set for the optimal menu and submodular approaches, when they are evaluated for the same menu size and the same number of preference scenarios; (2) As we increase the number of scenarios in Setup 2b, each replication retains the set previously used adding new elements as we go; (3) However, we use an independent and much larger scenario set to provide a point estimate. And we also do not use a scenario set for Thompson sampling; (4) When we perform evaluation, for each replication, we use the same evaluation set over all algorithms and menu sizes / scenario set sizes in both the utility graph and the time graph.

Setup 3 Here we study a dynamic, interactive, setting. We hold the menu size fixed at 3 vary the number of interactions with the DM. In each replication:

1. We first sample a θ from the prior, which is uniform over the probability simplex, and hold it out, not showing it to the algorithms.

2. We then use the algorithm to generate a menu.

3. We then simulate showing the menu to the DM, who selects the best item in the offered menu according to the held out θ. The algorithm observes the selected item.

4. We then update the prior distribution on θ to get a posterior distribution given the preference information expressed by the DM, i.e., that the selected item is better than all others in the menu.

5. We then use the algorithm to generate another menu, using the current posterior, and repeat the above steps. The value generated from the algorithm after m menus shown is the value of the best item shown among the first m menus, according to the held out θ.

We then average across many replications. We plot the average value of the best item in the first m menus vs. m, average along with the standard error.

For algorithms that require a small set of preference scenarios (submodular and optimal menu), we sample a fixed number of scenarios independently from the current posterior each time the algorithm is asked to generate a menu.

To sample from the posterior (required to form scenario sets for submodular and optimal menu, to form point estimate for the point estimate method, and used directly by Thompson sampling), we use acceptance rejection sampling: we sample from the prior, check whether it satisfies all of the constraints given by past selections from the DM, accept it if it does and reject it if it does not. We repeat this until we get the required number of samples.

5.2. Problem Descriptions

5.2.1. Doctor Scheduling The motivation behind this experiment is a fictional doctor who has a number of appointments that he wishes to schedule. There are A patients and B time-slots in which each patient can be scheduled. Patient a assigns a score $c_{a,b}$ to time-slot b, representing how much she likes this time slot. These scores are drawn independently across patients and time-slots from a uniform distribution over $[0,1]$ and fixed throughout the experiment. Each patient is also assigned a priority depending on how critical her appointment is. We partition the patients into $J = 5$ disjoint subsets $P_1, \dots, P_J$ uniformly at random and fix this partition throughout the experiment. A binary decision variable, $x_{a,b} \in \{0,1\}$, indicates whether patient a was scheduled in time-slot b so that $x_{a,b} = 1$ if patient n was scheduled in slot b, and $x_{a,b} = 0$ otherwise. Attributes here are given by the cumulative scores of patients with the same priority: $f_j(x) = \sum_{a \in P_j} \sum_{b=1}^{B} c_{a,b} x_{a,b}$

5.2.2. Intensity-Modulated Radiation Therapy Based on Chu et al. (2005), we consider a cancer patient undergoing intensity-modulated radiation therapy (IMRT). This therapy passes beams of radiation through the patient's body from a variety of angles. IMRT aims to modulate the intensity of each beam (or each portion of each beam, called a “beamlet”) so that the tumor receives enough radiation to initiate remission while the surrounding non-cancerous tissues receive a small enough dose to avoid the most significant side effects.

We divide the cross-sectional area of the patient's body irradiated by IMRT into N voxels. We let $w_v \geq 0$ be the radiation dosage at voxel v and $x_b \geq 0$ be the intensity of beamlet b.

The radiation dosage at voxel v is modeled as being a linear function of the beamlet intensities, $w_v = \sum_b D_{v,b} x_b$, where the $D_{v,b} \geq 0$ are known. We partition the voxels into J sets, $S_j : t = 1 \ldots J$, where each set represents a different type of tissue within the body: the tumor; healthy organs; and other surrounding tissues. We will support a physician's choice of x_b variables to control the w_v within each tissue to best achieve medical outcomes.

Attributes here are given by the average dose of radiation given to a tissue type (or its negation), where, under the assumption that $j = 1$ denotes the tumor, $f_1(x) = \frac{1}{|S_1|} \sum_{v \in S_1} w_v$, and $f_j(x) = -\frac{1}{|S_j|} \sum_{v \in S_j} w_v$, $\forall j \neq 1$. This is a linear function of the w_v and thus also the x_b. We also constrain the maximum dose in each tissue type (while avoiding large amounts of radiation focused on individual voxels is obviously desirable in non-cancerous tissue, it is also desirable in the tumor since such doses can cause cell death to occur in undesirable ways) and the minimum dose in the tumor. Although we do not do so here, our framework allows including the maxima and minima as additional attributes over which the DM can have uncertain preferences.

5.3. Results

Figures 1 through 4 below show results for each experimental setup with the doctor scheduling problem in the top row and IMRT in the bottom. The left column shows expected quality of the best item found and the right column shows the computational time used, including the time to sample the requisite number of preference scenarios. Computational time is reported for a 2.5 GHz Quad-Core Intel Core i7 with 16 GB of RAM.

The three novel algorithms that we propose significantly outperform the status quo point estimate method. The results show the clear trade-off between an algorithm's utility and its computational cost. While the optimal menu algorithm tends to offer the best solution quality, it does so at a

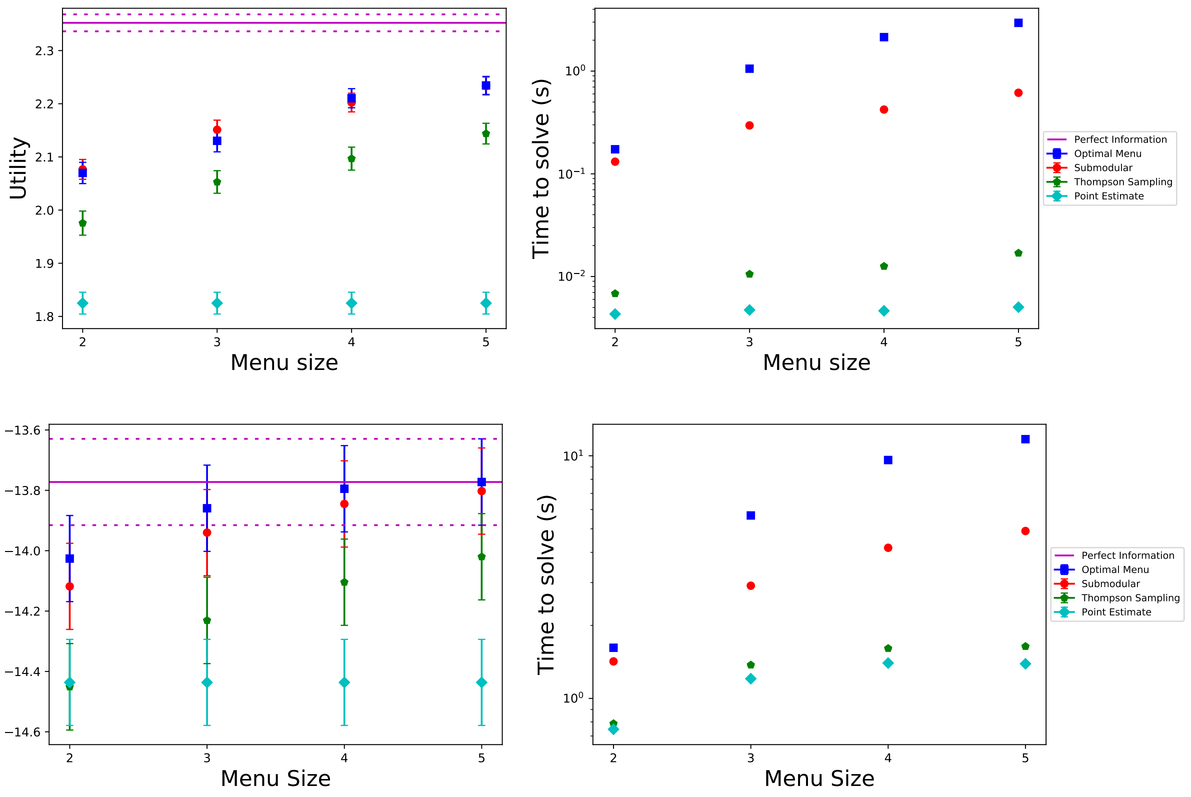

Figure 1 Setup 1, where the prior uses a small number of scenarios which are also used to compute the given menus. Doctor scheduling uses 8 distinct preference scenarios and 100 replications while IMRT uses 5 scenarios and 2500 replications.

greater computational cost, growing much faster in its time to solve with the menu size and the number of preference scenarios used. The submodular algorithm tends to offer a solution quality that is almost as good, trailing just underneath the optimal menu algorithm in nearly all instances. It is able to do this with much better computational scalability. Then, we have Thompson sampling, which offers a different tradeoff between performance and speed, being much faster than submodular and the optimal menu, and offering much better performance than the point estimate method. Thus, we offer 3 choices with different tradeoffs between solution quality and computational cost from which a user can select based on their circumstances.

An interesting point where IMRT differs from Doctor Scheduling is that in Figure 1, we see that, when we have a menu size of five, the optimal menu algorithm performs exactly as well as when we have perfect information. This is due to us generating a menu size of five when there are only five

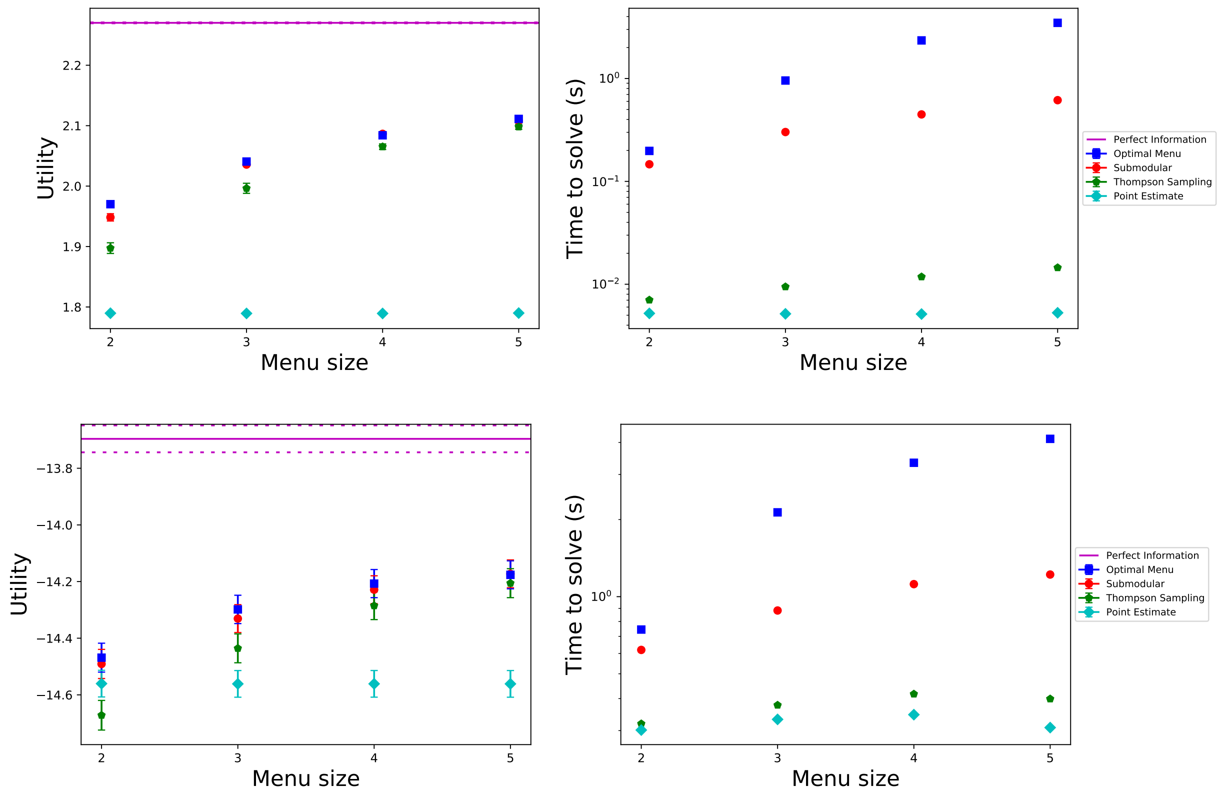

Figure 2 Setup 2a, where we use a sample of 1000 scenarios from the prior as the ground truth. The menu size is varied from 2 to 5. Doctor Scheduling gives 8 preference scenarios to submodular/optimal menu algorithms while IMRT uses 5. Both use 100 replications

ground truth values, and all are being given to the algorithm. Another point is that in Figure 3 for IMRT, the submodular approach sometimes outperforms the optimal menu algorithm. This is due the case that sometimes an optimal solution using a set of size $L \ll 1000$ preference scenarios might not be optimal for the ground truth of size 1000.

Moreover, we note that across all static experiments (Figures **??** through **??**), the standard errors tend to be small. Standard errors also tend to decline with menu size and are smaller for methods with better overall expected solution set quality. We believe that this is because methods with good expected solution set quality tend to contain near-optimal solutions for each θ in their offered set regardless of randomness introduced by sampling over the thetas used to construct the menu. This reduces the variance of a single evaluation of solution quality, thus reducing overall standard errors.

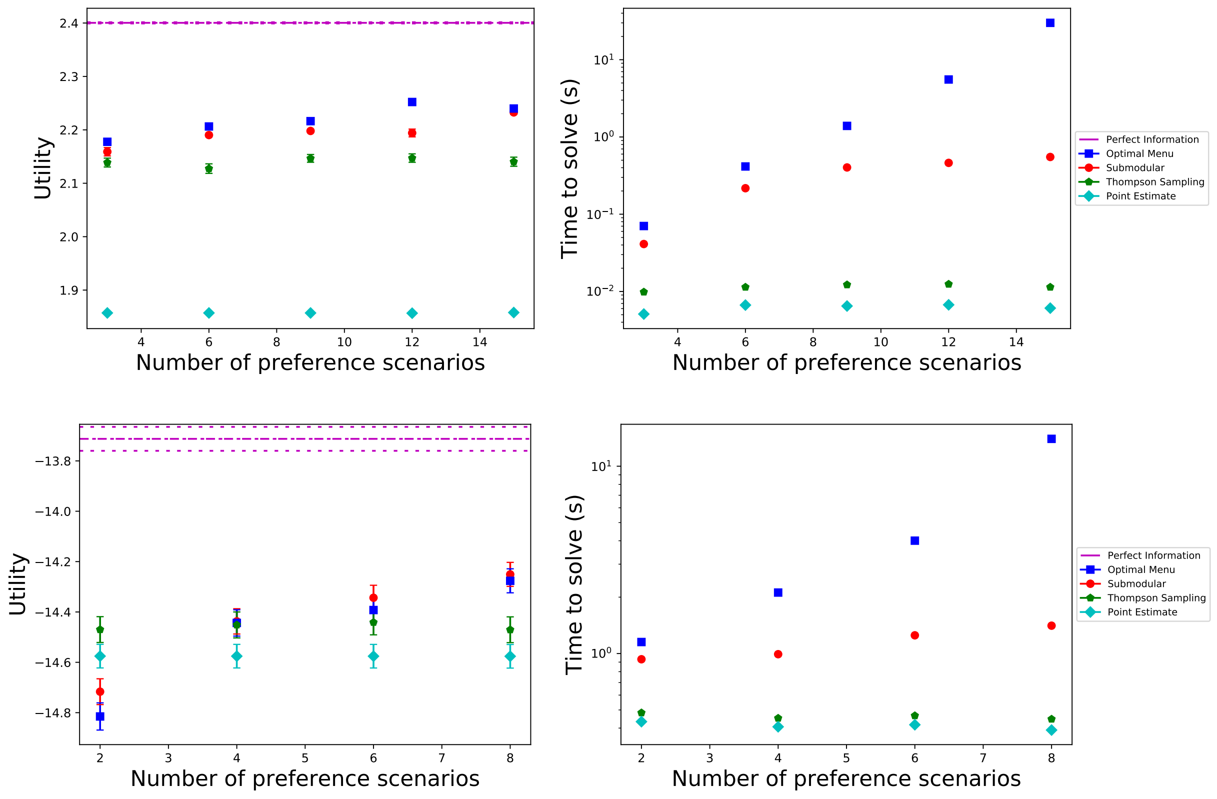

Figure 3 Setup 2b, where we use a sample of 1000 scenarios from the prior as the ground truth. We vary the number of theta given to optimal menu/submodular algorithms. The menu size is held constant at 3. Doctor Scheduling provides $\{3, 6, 9, 12, 15\}$ scenarios and IMRT provides $\{2, 4, 6, 8\}$.Both use 100 replications.

In the dynamic experiment (Figure 4), the algorithms perform well at their task of learning the DM's preferences. After a few iterations of presenting menus to the DM, we can see that the utility of the menus provided perform close to optimal against the true preferences of the DM. Moreover, we note that even a cheap algorithm, that does not perform as well in a non-dynamic setting such as Thompson sampling can perform close to optimal after a few interactions with the DM, as it can learn the preferences well enough anyway. Thus, we can see that in the case that the DM can spare enough bandwidth to interact with the user, it might be fine to use an inexpensive algorithm. However, in the case that the DM can afford few interactions, it is better to use a more computationally expensive algorithm, which would be better at giving us something with higher utility.

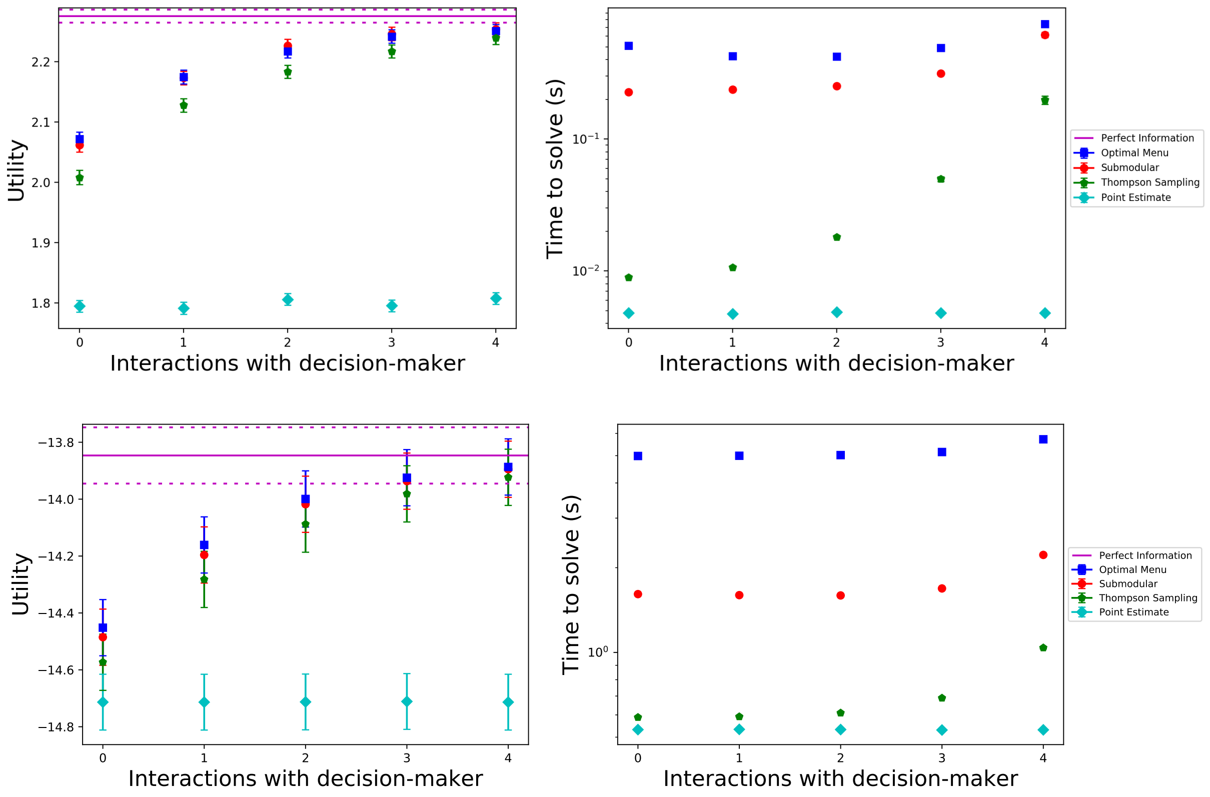

Figure 4 Setup 3, optimizing with a menu size of 3. We modulate the number of interactions with the DM and update the prior on preference scenarios after each interaction. We use 1000 replications in Doctor Scheduling and 25000 in IMRT.

6. Conclusion

We introduced a novel approach to support MILP-based decision-making when there are multiple attributes and the DM's preferences over these attributes are uncertain. By acknowledging uncertainty in the DM's preferences, our approach is more flexible than the point estimate approach, which provides a single suggested solution. By leveraging preference information, our approach provides solutions better tailored to the DM's preferences than generating the whole Pareto front. By constructing the menu using a decision-theoretic analysis and a Bayesian prior distribution over utility functions, our approach is significantly more flexible than existing interactive multi-objective integer linear programming methods: it can warm-start the first menu using an informative prior distributions; it can tolerate noisy inconsistent DM's responses; and it can leverage selections made in other related problems or by other DMs.

We proposed three algorithms that achieve different levels of solution quality at different levels of computational cost, and proved theoretical guarantees for each of them. Our numerical experiments show that these algorithms significantly outperform the traditional approach of using a point estimate of the decision-maker's utility function.

References

Abbas, A. E. (2018). *Foundations of multiattribute utility.* Cambridge University Press.

Aksoy, Y. (1990). An interactive branch-and-bound algorithm for bicriterion nonconvex/mixed integer programming. *Naval Research Logistics (NRL)*, 37(3):403–417.

Alves, M. J. and Clímaco, J. (2007). A review of interactive methods for multiobjective integer and mixed-integer programming. *European Journal of Operational Research*, 180(1):99–115.

Astudillo, R. and Frazier, P. (2020). Multi-attribute bayesian optimization with interactive preference learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4496–4507, Online. PMLR.

Benson, H. P. (1998). An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *Journal of Global Optimization*, 13(1):1–24.

Boutilier, C. (2002). A pomdp formulation of preference elicitation problems. In *AAAI/IAAI*, pages 239–246.

Boutilier, C., Patrascu, R., Poupart, P., and Schuurmans, D. (2006). Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artificial Intelligence*, 170(8-9):686–713.

Braziunas, D. (2006). Computational approaches to preference elicitation. Technical report, University of Toronto, Department of Computer Science.

Chajewska, U., Getoor, L., Norman, J., and Shahar, Y. (1998). Utility elicitation as a classification problem. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 79–88. Morgan Kaufmann Publishers Inc.

Chajewska, U., Koller, D., and Parr, R. (2000). Making rational decisions using adaptive utility elicitation. In *AAAI/IAAI*, pages 363–369.

Chu, M., Zinchenko, Y., Henderson, S. G., and Sharpe, M. B. (2005). Robust optimization for intensity modulated radiation therapy treatment planning under uncertainty. *Physics in Medicine and Biology*, 50:5463–5477.

Clímaco, J., Ferreira, C., and Captivo, M. E. (1997). Multicriteria integer programming: An overview of the different algorithmic approaches. In *Multicriteria analysis*, pages 248–258. Springer.

De Loera, J. A., Hemmecke, R., and Köppe, M. (2009). Pareto optima of multicriteria integer linear programs. *INFORMS Journal on Computing*, 21(1):39–48.

Ehrgott, M. (2005). *Multicriteria optimization*, volume 491. Springer Science & Business Media.

Ehrgott, M. and Gandibleux, X. (2000). A survey and annotated bibliography of multiobjective combinatorial optimization. *OR-Spektrum*, 22(4):425–460.

Gonzalez, J. J., Reeves, G. R., and Franz, L. S. (1985). An interactive procedure for solving multiple objective integer linear programming problems. In *Decision Making with Multiple Objectives*, pages 250–260. Springer.

Homem-de Mello, T. (2008). On rates of convergence for stochastic optimization problems under non-independent and and identically distributed sampling. *SIAM Journal on Optimization*, 19(2):524–551.

Isermann, H. (1974). Proper efficiency and the linear vector maximum problem. *Operations Research*, 22(1):189–191.

Jahn, J. (2009). *Vector optimization.* Springer.

Keeney, R. L. (1977). The art of assessing multiattribute utility functions. *Organizational behavior and human performance*, 19(2):267–310.

Kim, S., Pasupathy, R., and Henderson, S. G. (2015). A guide to sample average approximation. In *Handbook of simulation optimization*, pages 207–243. Springer.

Lokman, B., Köksalan, M., Korhonen, P. J., and Wallenius, J. (2016). An interactive algorithm to find the most preferred solution of multi-objective integer programs. *Annals of operations research*, 245(1-2):67–95.

Marcotte, O. and Soland, R. M. (1986). An interactive branch-and-bound algorithm for multiple criteria optimization. *Management Science*, 32(1):61–75.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294.

Ramesh, R., Karwan, M. H., and Zionts, S. (1990). An interactive method for bicriteria integer programming. *IEEE transactions on systems, man, and cybernetics*, 20(2):395–403.

Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., Wen, Z., et al. (2018). A tutorial on Thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96.

Shin, W. S. and Allen, D. B. (1994). An interactive paired comparison method for bicriterion integer programming. *Naval Research Logistics (NRL)*, 41(3):423–434.

Teghem, J. and Kunsch, P. (1986). Interactive methods for multi-objective integer linear programming. In *Large-scale modelling and interactive decision analysis*, pages 75–87. Springer.

White, D. (1985). A multiple objective interactive lagrangean relaxation approach. *European journal of operational research*, 19(1):82–90.

Zionts, S. (1979). A survey of multiple criteria integer programming methods. In *Annals of Discrete Mathematics*, volume 5, pages 389–398. Elsevier.

Appendix

A.1. Proof of Theorems 1 and 2

Theorem 1. *Suppose that $\mathbb{X}$ is compact, and let v^* and $\widehat{v}^*(K)$ be the optimal values of problems 1 and 4, respectively. Also let X^* be the set of optimal solutions of 1 and $\widehat{x}^*(K) = (\widehat{x}_1^*(K), \ldots, \widehat{x}_M^*(K))$ be any optimal solution of problem 4. Then, $\widehat{v}^*(K) \to v^\star$ and $\mathrm{dist}(\widehat{x}^*(K), X^*) \to 0$ almost surely as $K \to \infty$.*

Proof: This theorem is a direct consequence of Proposition 2.2 in Homem-de Mello (2008). It suffices to verify that the two conditions below are satisfied.

1. For any $x_1, \ldots, x_M \in \mathbb{X}$,

$$\frac{1}{K}\sum_{k=1}^{K} \max_{m=1,\ldots,M} U(f(x_m);\theta_k) \to \mathbb{E}\left[\max_{m=1,\ldots,M} U(f(x_m);\theta)\right]$$

almost surely as $K \to \infty$.

2. There exists a function $\ell : \Theta \to \mathbb{R}$ such that $\mathbb{E}[\ell(\theta)] < \infty$ and

$$\left|\max_{m=1,\ldots,M} U(f(x_m);\theta) - \max_{m=1,\ldots,M} U(f(x'_m);\theta)\right| \le \ell(\theta)\,\|(x_1,\ldots,x_M) - (x'_1,\ldots,x'_M)\|_2\,.$$

Condition 1 follows from the law of large numbers. Thus, it only remains to verify condition 2. Let $n_* \in \arg\max_{m=1,\ldots,M} U(f(x_m);\theta)$ and $n'_* \in \arg\max_{m=1,\ldots,M} U(f(x'_m);\theta)$. Note that

$$\begin{aligned}
&\left|\max_{m=1,\ldots,M} U(f(x_m);\theta) - \max_{m=1,\ldots,M} U(f(x'_m);\theta)\right| \\
&= \max\{U(f(x_{n_*});\theta) - U(f(x'_{n'_*});\theta), U(f(x'_{n'_*});\theta) - U(f(x_{n_*});\theta)\} \\
&\le \max\{U(f(x_{n_*});\theta) - U(f(x'_{n_*});\theta), U(f(x'_{n'_*});\theta) - U(f(x_{n'_*});\theta)\} \\
&= \max\{\theta^\top C x_{n_*} - \theta^\top C x'_{n_*}, \theta^\top C x'_{n'_*} - \theta^\top C x_{n'_*}\} \\
&\le \|\theta\|_2 \|C\|_2 \max\{\|x_{n_*} - x'_{n_*}\|, \|x_{n'_*} - x'_{n'_*}\|\} \\
&\le \|\theta\|_2 \|C\|_2\, \|(x_1,\ldots,x_M) - (x'_1,\ldots,x'_M)\|_2\,.
\end{aligned}$$

Therefore, it is enough to take $\ell(\theta) = \|C\|_2 \|\theta\|_2$.

Theorem 2. *Suppose that $\mathbb{X}$ is compact and that the moment generating function of $\|\theta\|_2$, $t \mapsto \mathbb{E}\left[e^{t\|\theta\|_2}\right]$ is finite in an open neighborhood of 0. Then, for all $\delta > 0$, there exist $G > 0$ and $\gamma > 1$ such that $\mathbb{P}(\text{dist}(\widehat{x}^*(K), X^*) > \delta) \leq Ge^{-\gamma K}$.*

Proof: Again, this is a direct consequence of Theorem 2.3 in Homem-de Mello (2008). It suffices to verify that the following two conditions are satisfied.

1. The moment generating function of $\ell(\theta)$, $t \mapsto \mathbb{E}\left[e^{t\ell(\theta)}\right]$, where ℓ is defined as in the proof of Theorem 1, is finite in an open neighborhood of 0.

2. For any $x_1, \ldots, x_M \in \mathbb{X}$, the moment generating function of $\max_{m=1,\ldots,M} U(f(x_m);\theta)$, $t \mapsto \mathbb{E}\left[e^{t \max_{m=1,\ldots,M} U(f(x_m);\theta)}\right]$, is finite in an open neighborhood of 0.

Recall that $\ell(\theta) = \|C\|_2\|\theta\|_2$, and thus condition 1 follows directly from the hypothesis that the moment generating function of $\|\theta\|_2$ is finite in an open neighborhood of 0. Condition 2 follows easily from this hypothesis as well since $|\max_{m=1,\ldots,M} U(f(x_m);\theta)| \leq \|\theta\|_2\|C\|_2 \max_{m=1,\ldots,M} \|x_m\|_2$.

A.2. Proofs of Theorems 4 and 5

Theorem 4. *Suppose that $\mathbb{X}$ is compact and that $t_0 \in \Theta$ is such that $\mathbb{P}(\|\theta - t_0\|_2 < \delta) > 0$ for all $\delta > 0$. Also suppose that $\theta_1, \ldots, \theta_M \overset{iid}{\sim} p_\theta$ and let $x^*(\theta_m) \in \arg\max_{x\in\mathbb{X}} U(f(x);\theta_m)$, $m = 1, \ldots, M$. Then, for any $\epsilon > 0$,*

$$\mathbb{P}\left(\max_{x\in\mathbb{X}} U(f(x);t_0) - \max_{m=1,\ldots,M} U(f(x^*(\theta_m));t_0) > \epsilon\right)$$

converges to 0 exponentially fast as N goes to ∞.

Proof: Note that $\max_{x\in\mathbb{X}} U(f(x);t) = \max_{x\in\mathbb{X}} t^\top Cx$ is simply the value of the support function of $\mathbb{X}$ at $C^\top t$. Since $\mathbb{X}$ is convex and compact, its support function is continuous. Therefore, given $\epsilon > 0$, there exists $\delta' > 0$ such that if $\|t - t_0\|_2 < \delta'$,

$$\left|\max_{x\in\mathbb{X}} U(f(x);t) - \max_{x\in\mathbb{X}} U(f(x);t_0)\right| < \frac{\epsilon}{2}.$$

On the other hand, if $\|t - t_0\| < \epsilon/2\,(\max_{x\in\mathbb{X}} \|Cx\|_2 + 1)$, then we have

$$|U(f(x^*(t));t) - U(f(x^*(t));t_0)| = \left|t^\top Cx^*(t) - t_0^\top Cx^*(t)\right|$$

$$\begin{aligned}
&\leq \|t-t_0\|_2 \|Cx^*(t)\|_2 \\
&< \frac{\epsilon}{2\left(\max_{x\in\mathbb{X}}\|Cx\|_2+1\right)}\|Cx^*(t)\|_2 \\
&< \frac{\epsilon}{2}.
\end{aligned}$$

It follows that, if $\|t-t_0\| < \delta := \min\{\delta', \epsilon/2\left(\max_{x\in\mathbb{X}}\|Cx\|_2+1\right)\}$, then

$$\begin{aligned}
&\max_{x\in\mathbb{X}} U(f(x);t_0) - U(f(x^*(t));t_0) \\
&= \left|\max_{x\in\mathbb{X}} U(f(x);t_0) - \max_{x\in\mathbb{X}} U(f(x);t) + U(f(x^*(t));t) - U(f(x^*(t));t_0)\right| \\
&\leq \left|\max_{x\in\mathbb{X}} U(f(x);t) - \max_{x\in\mathbb{X}} U(f(x);t_0)\right| + |U(f(x^*(t));t) - U(f(x^*(t));t_0)| \\
&< \epsilon.
\end{aligned}$$

Hence, the event $\{\max_{x\in\mathbb{X}} U(f(x);t_0) - U(f(x^*(\theta));t_0) \geq \epsilon\}$ is contained in the event $\{\|\theta - t_0\| \geq \delta\}$, and thus

$$\begin{aligned}
\mathbb{P}\left(\max_{x\in\mathbb{X}} U(f(x);t_0) - U(f(x^*(\theta));t_0) \geq \epsilon\right) &\leq \mathbb{P}\left(\|\theta - t_0\| \geq \delta\right) \\
&= 1 - \mathbb{P}\left(\|\theta - t_0\| < \delta\right).
\end{aligned}$$

Finally,

$$\left\{\max_{x\in\mathbb{X}} U(f(x);t_0) - \max_{m=1,\ldots,M} U(f(x^*(\theta_m));t_0) \geq \epsilon\right\} = \bigcap_{n=1}^{M}\left\{\max_{x\in\mathbb{X}} U(f(x);t_0) - U(f(x^*(\theta_m));t_0) \geq \epsilon\right\},$$

and thus

$$\begin{aligned}
&\mathbb{P}\left(\max_{x\in\mathbb{X}} U(f(x);t_0) - \max_{m=1,\ldots,M} U(f(x^*(\theta_m));t_0) \geq \epsilon\right) \\
&= \mathbb{P}\left(\bigcap_{n=1}^{M}\left\{\max_{x\in\mathbb{X}} U(f(x);t_0) - U(f(x^*(\theta_m));t_0) \geq \epsilon\right\}\right) \\
&= \prod_{n=1}^{M}\mathbb{P}\left(\max_{x\in\mathbb{X}} U(f(x);t_0) - U(f(x^*(\theta_m));t_0) \geq \epsilon\right) \\
&\leq \left(1 - \mathbb{P}\left(\|\theta - t_0\| < \delta\right)\right)^M,
\end{aligned}$$

which concludes the proof since $\mathbb{P}\left(\|\theta - t_0\| < \delta\right) > 0$.

Theorem 5. *Suppose that both* $\mathbb{X}$ *and* Θ *are compact, and let* $\theta_0, \theta_1, \ldots, \theta_M \overset{iid}{\sim} p_\theta$, $x^*(\theta_m) \in \arg\max_{x \in \mathbb{X}} U(f(x); \theta_m)$, $m = 1, \ldots, M$. *Further suppose that the following two conditions hold*

1. *The set* $\Theta' = \{t \in \Theta : \mathbb{P}(\|\theta - t\|_2 < \delta) > 0 \text{ for all } \delta > 0\}$ *has probability 1 under* p_θ.
2. *For any* $\delta > 0$, $\inf_{t \in \Theta'} \mathbb{P}(\|\theta - t\|_2 < \delta) > 0$.

Then, for any $\epsilon > 0$,

$$\mathbb{P}\left(\max_{x \in \mathbb{X}} U(f(x); \theta_0) - \max_{m=1,\ldots,M} U(f(x^*(\theta_m)); \theta_0) > \epsilon\right)$$

converges to 0 *exponentially fast as* $M \to \infty$.

Proof: Since Θ is compact, the support function of $\mathbb{X}$ is not only continuous but uniformly continuous over $C^\top \Theta$, and thus δ' in the proof of Theorem 1 can chosen independently of t_0, which in turn implies that δ can also be chosen independently of t_0. Therefore, given $\epsilon > 0$, there exists $\delta > 0$ such that

$$\begin{aligned}\mathbb{P}\left(\max_{x \in \mathbb{X}} U(f(x); t) - \max_{m=1,\ldots,M} U(f(x^*(\theta_m)); t) \geq \epsilon\right) &\leq \left(1 - \mathbb{P}\left(\|\theta - t\| < \delta\right)\right)^M \\ &\leq \left(1 - \inf_{t \in \Theta'} \mathbb{P}\left(\|\theta - t\| < \delta\right)\right)^M\end{aligned}$$

for all $t \in \Theta'$.

Since Θ' has probability 1 under p_θ, it follows that

$$\mathbb{E}\left[\mathbb{P}\left(\max_{x \in \mathbb{X}} U(f(x); \theta_0) - \max_{m=1,\ldots,M} U(f(x^*(\theta_m)); \theta_0) \geq \epsilon \mid \theta_0\right)\right] \leq \left(1 - \inf_{t \in \Theta'} \mathbb{P}\left(\|\theta - t\| < \delta\right)\right)^M,$$

where the expectation is over θ_0; i.e.,

$$\mathbb{P}\left(\max_{x \in \mathbb{X}} U(f(x); \theta_0) - \max_{m=1,\ldots,M} U(f(x^*(\theta_m)); \theta_0) \geq \epsilon\right) \leq \left(1 - \inf_{t \in \Theta'} \mathbb{P}\left(\|\theta - t\| < \delta\right)\right)^M,$$

which finishes the proof since $\inf_{t \in \Theta'} \mathbb{P}\left(\|\theta - t\| < \delta\right) > 0$.

A.3. POMDP Formulation in the Dynamic Setting

We now provide a formulation of the dynamic version of our problem as a partially observable Markov decision process (POMDP). The set of actions is given by all possible menus of size M which, in principle, are given by all subsets of $\mathbb{X}$ of size M. Due to optimization amenability, however, we momentarily ignore the order and duplicity of items in the menu and represent the set of actions by $\mathbb{X}^M$. For now we let the set of belief states be given by all possible distributions over Θ. At every time step, the