Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

MULTI-ATTRIBUTE OPTIMIZATION UNDER PREFERENCE UNCERTAINTY

Bhavik A. Shah[1], Raul Astudillo[2], Peter I. Frazier[3]

[1]Google, Mountain View, CA, USA
[2]Department of Computing and Mathematical Sciences, Caltech, Pasadena, CA, USA
[3]School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA

ABSTRACT

We introduce multi-attribute optimization under preference uncertainty, a novel approach for optimization-based decision support when the decision-maker's preferences are uncertain. Here, each feasible design is associated with a vector of attributes, which is in turn assigned a utility by the decision-maker's utility function. This utility function has been incompletely estimated, and its remaining uncertainty is quantified by a Bayesian probability distribution. We develop an optimization-based formulation to generate a menu of diverse solutions, among which the decision-maker is expected to find a high-utility solution. The resulting optimization problem is challenging to solve in general, but we show that it can be approximately solved efficiently when the attributes and utility function are linear by reformulating it as a mixed-integer linear program, and using sample average approximation and submodular maximization. We also propose a posterior sampling approach that only requires optimizing individual samples of the user's utility function, supporting fast computation.

1 INTRODUCTION

As motivation, we consider the following example, which combines preference elicitation (Braziunas 2006; Abbas 2018) with classical multi-objective mixed-integer linear programming.

Consider a project between a local hospital and an external operations research (OR) team. The OR team's goal is to help the hospital administration create a weekly schedule for assigning nurses to hospital wards and shifts, which will 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; 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, θ is unknown.

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.

Such an 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 imply a preference for 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 recently enjoyed a 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, and allow the DM to choose. This would offer more value across a range of possible DM 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 solve the MILP for each such θ_ℓ and offer the set of solutions as a menu, allowing the DM to choose. This, however, is computationally infeasible and also overwhelming for the DM for large L.

To address the above challenges, we introduce a novel approach that acknowledges uncertainty on the DM's preferences but retains computational tractability. 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 her most preferred solution. She 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 with much less computation while also requiring much less effort from the human DM. Offering a menu dramatically outperforms optimizing based on a point estimate of the DM's utility function. Moreover, by leveraging probabilistic information about the DM's preferences, our method is more efficient than standard multi-objective linear programming.

Within this framework, we propose two approaches for solving this higher-level optimization problem using techniques from simulation. We focus on the practically important settings where the original known-preference problem is a mixed integer linear program (MILP). The first approach begins by defining an *optimal menu* given a constraint on menu size. Within this approach, we develop two tools for approximately computing the optimal menu, sample average approximation and submodular maximization, and provide performance guarantees on the expected reward they produce relative to the optimal menu. The second approach uses Thompson sampling to create a menu of options by repeatedly sampling from the posterior distribution on θ and appending the solution that would be preferred under preferences described by θ. We provide a performance guarantee on the gap between the reward achieved by this menu and that achieved by an infinitely large menu containing all solutions.

2 RELATED WORK

Our work is closely related to multi-objective MILP (Zionts 1979; Clímaco et al. 1997; Alves and Clímaco 2007). Multi-objective LPs have convex feasible regions, allowing points on the Pareto frontier to be recovered by minimizing a linear function of the objectives (Isermann 1974). Enumerating the Pareto frontier for multi-objective MILPs is more complex because their feasible regions are non-convex (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).

Among the broad literature on multi-objective MILP, our work is most closely related to *interactive* decision support for multi-objective LPs and MILPs, surveyed in Teghem and Kunsch (1986) and Alves and Clímaco (2007). In this literature, our work is especially close to *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 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. Our method is fundamentally different from these past methods through the fact that it builds a Bayesian probabilistic model over the implicit utility function and leverages this to build a menu with high expected utility. This allows our method to offer several key improvements over past methods in this space.

Also related is utility elicitation (Abbas 2018), which queries the DM to estimate her utility function. This estimate can be then used for decision-making (Keeney 1977). In practice, however, elicitation typically demands substantial time and cognitive effort from the DM, preventing complete estimation of the utility function. 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, preference elicitation has also been studied within computer science (Chajewska et al. 1998; Chajewska et al. 2000; Boutilier 2002; Boutilier et al. 2006). Unlike our work, this literature focuses on finding a single solution as opposed to a menu of high-quality solutions.

Our work is also related to dueling bandits (Yue et al. 2012) and preferential Bayesian optimization (Astudillo et al. 2023), both of which optimize based on preference feedback. These approaches query the DM to efficiently identify the utility function's maximizer. However, unlike our work, their focus is on finding a single optimal solution. In the dynamic setting described below, our strategy based on the optimal menu parallels the qEUBO sampling strategy for preference-based Bayesian optimization (Astudillo et al. 2023), whereas our Thompson sampling-inspired strategy parallels the dueling Thompson sampling strategy (Astudillo et al. 2023; Astudillo et al. 2024).

Finally, our work builds on previous research by the authors, which has introduced similar problem formulations in the contexts of multi-attribute ranking and selection (Frazier and Kazachov 2011) and multi-attribute Bayesian optimization (Astudillo and Frazier 2020).

3 PROBLEM SETTING

We assume that the space of designs is represented by a set $\mathbb{X} \subseteq \mathbb{R}^d$, and attributes are given by a function $f : \mathbb{X} \to \mathbb{R}^k$. We also assume that there is a DM whose preference over designs is characterized by a design's attributes, y, through a utility function, $U(\,\cdot\,;\theta) : \mathbb{R}^k \to \mathbb{R}$, where $\theta \in \Theta$ is a parameter vector characterizing the DM's preferences. Thus, of all the designs, the DM most prefers one in the set $\text{argmax}_{x\in\mathbb{X}} U(f(x);\theta)$. If θ were known to us, we could apply single-objective 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, denoted by 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.

An algorithm provides the DM with a menu of designs and receives as its reward the expected value of the utility obtained when she selects the best design in this menu according to her underlying utility function. Formally, the reward received from offering designs $x_1,\ldots,x_M$ is

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

where the expectation is over θ.

In principle, our framework is applicable as long as (1) can be maximized efficiently. However, this is challenging in general. In this work, we focus on a specific subclass of problems that allows us to maximize (1) in a computationally tractable way. Specifically, unless otherwise stated, we assume that $\mathbb{X} = \{x \in \mathbb{R}^I : Ax \leq b\}$, possibly intersected with $\mathbb{Z}^{I'} \times \mathbb{R}^{I-I'}$, where $A \in \mathbb{R}^{K\times I}$ and $b \in \mathbb{R}^K$, $f(x) = Cx$, where $C \in \mathbb{R}^{J\times I}$, and $U(y;\theta) = \theta^\top y$ for all $\theta \in \Theta \subseteq \mathbb{R}^k$.

In our experiments, we consider two settings: a *static* setting and a *dynamic* setting. In the static setting, we compute a menu and provide it to the DM. In the dynamic setting, this process is repeated multiple times. Each time we provide a menu to the DM, she selects her preferred option, and this information is used to update the (posterior) distribution over θ. In practice, we repeat this process as many times as necessary until the DM is satisfied with the menu options.

We derive performance guarantees in the static setting for algorithms relative to two benchmarks. The first is the maximum expected utility possible under a fixed menu size, $\max_{x_1,\ldots,x_M} \mathbb{E}[\max_{m=1,\ldots,M} U(f(x_m);\theta)]$. The second benchmark is the maximum utility possible if the DM selected from an infinite menu containing all items, $\max_{x\in\mathbb{X}} U(f(x);\theta)$. Since $\max_{x\in\mathbb{X}} U(f(x);\theta) \geq \max_{m=1,\ldots,M} U(f(x_m);\theta)$ for any $x_1,\ldots,x_m$, the expected value of the second benchmark dominates the first benchmark.

The following two sections develop two approaches. First, Section 4 develops an approach motivated by the optimal menu of a fixed size M, i.e., the collection of items $x_1,\ldots,x_M$ that maximize (1). This approach relies on two computational tools: sample average approximation and submodular maximization. We provide guarantees on the gap between the expected reward (1) and the expected reward of the optimal menu (the first benchmark) under both tools. Then, Section 5 develops a Thompson-sampling-based approach that directly increases the reward by adding items to the menu. We show that the probability of a large gap between the reward obtained and the second benchmark converges exponentially fast to 0.

4 APPROXIMATE COMPUTATION OF THE OPTIMAL MENU

Our first approach is based on direct (approximate) computation of the optimal menu, i.e., the menu that maximizes the objective in Equation (1). Formally, this menu is the solution to

$$\max_{x_1,\ldots,x_M\in\mathbb{X}} \mathbb{E}\left[\max_{m=1,\ldots,M} U(f(x_m);\theta)\right]. \tag{2}$$

Solving this problem is challenging due to the stochastic nature of the problem induced by the expectation over θ and the combinatorial structure of the problem resulting from maximizing $x_1,\ldots,x_M$ over $\mathbb{X}$.

To develop tools addressing these challenges, we begin by observing that, if the support of θ is finite, Problem (2) is equivalent to the mixed integer linear program

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

where l_θ and u_θ are any constants satisfying $l_\theta \leq U(f(x);\theta) \leq u_\theta$ 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 2.

If θ has large or infinite support, or the menu size M is large, solving (3) may not be computationlly tractable. We thus offer two tools for addressing this challenge: sample average approximation (4.1) and submodular maximization (4.2). They can be applied separately or jointly. Sample average approximation is appropriate when θ has large or infinite support. Submodular maximization is appropriate when the menu size is large. They can also be applied jointly by using sample average approximation to solve the subproblem considered by submodular maximization, as discussed in Section 4.2.

4.1 Sample Average Approximation

The above reformulation (3) is only possible when θ has finite support. However, θ often has infinite support in practice. Moreover, even if θ has finite support, Problem (3)'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 solve this problem approximately. Instead of using the full support of θ, this approach draws L i.i.d. samples from p, denoted by $\theta_1, \ldots, \theta_L$, and replaces Problem (2) by

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

This approach is known in the literature as *sample average approximation* (Kim et al. 2015). Despite its simplicity, it possesses appealing theoretical guarantees, as formalized by Theorem 1. The proof of this result can be found in Appendix A.

Theorem 1 Suppose that $\mathbb{X}$ is compact, and let v^* and $\widehat{v}^*(L)$ be the optimal values of Problems 2 and 4, respectively. Also let $X^* = \text{argmax}_{x_1,\ldots,x_M \in \mathbb{X}} \mathbb{E}\left[\max_{m=1,\ldots,M} U(f(x_m); \theta)\right]$ be the set of optimal solutions of Problem 2 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 $\text{dist}(\widehat{x}^*(L), X^*) \to 0$ almost surely as $L \to \infty$.

4.2 Submodular Maximization

To address computational challenges arising from large menu sizes, we make 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 an appealing 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 $V(A) \le V(B)$; and V is said to be submodular if $V(A) + V(B) \ge V(A \cup B) + V(A \cap B)$ for all $A, B \subset \Omega$.

We consider the approach that builds the menu to be shown to the DM iteratively as follows. For each $N = 1, \ldots, M$ 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. The approximation guarantee of this approach is formalized in Theorem 2 below, whose proof follows directly from the classical result in Nemhauser et al. (1978). For this result, we assume that $U(f(x); \theta) \ge 0$ for all $x \in \mathbb{X}$ and $\theta \in \Theta$, which can be done without loss of generality in most cases by subtracting an appropriate constant.

Theorem 2 Suppose that $x_1^*, \ldots, x_M^*$ are chosen as described above. Then,

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

Problem 5 can be formulated as a MILP analogous to (3). 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

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

where $U^*_{N-1}(\theta) = \max_{m=1,\ldots,N-1} U(f(x_m);\theta)$ and, as before, l_θ and u_θ are any constants satisfying $l_\theta \leq U(f(x);\theta) \leq u_\theta$ for all $x \in \mathbb{X}$. As before, we can use sample average approximation to approximately solve the above problem when the support of θ is large.

5 A THOMPSON-SAMPLING-INSPIRED APPROACH

Our second approach for ensuring a large value for the reward (1) is inspired by Thompson sampling, an algorithm for optimization-based sequential decision-making under uncertainty (Russo et al. 2018). While our setting is not necessarily sequential, we take inspiration from this algorithm and consider the following strategy to build the menu of designs to be shown to the DM. Notably, this approach only requires maximizing individual samples from the utility function, making it straightforward to implement in scenarios where software for optimizing point estimates of the utility function is available. Specifically, this approach draws M i.i.d. samples from p, denoted by $\theta_1,\ldots,\theta_M$, and builds the menu as $x^*_1,\ldots,x^*_M$, where $x^*_m \in \text{argmax}_{x\in\mathbb{X}} U(f(x);\theta_m),\ m = 1,\ldots,M$.

By drawing an analogy to the balance of exploitation and exploration provided by Thompson sampling in sequential settings, a menu constructed using our approach is likely to include designs that are both individually high-quality and collectively diverse. Our experiments demonstrate that this straightforward algorithm performs effectively in practice. Furthermore, we show that a menu built following this approach achieves an optimal performance exponentially fast as the size of the menu grows. This result is formalized in Theorem 3, whose proof can be found in Appendix A.

Theorem 3 Suppose that both $\mathbb{X}$ and Θ are compact, and let $\theta_0,\theta_1,\ldots,\theta_M \overset{iid}{\sim} p$, $x^*(\theta_m) \in \text{argmax}_{x\in\mathbb{X}} U(f(x);\theta_m)$, $m = 1,\ldots,M$. 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.
2. For any $\delta > 0$, $\inf_{t\in\Theta'} \mathbb{P}(\|\theta - t\|_2 < \delta) > 0$.

Then, for any $\varepsilon > 0$, $\mathbb{P}(\max_{x\in\mathbb{X}} U(f(x);\theta_0) - \max_{m=1,\ldots,M} U(f(x^*(\theta_m));\theta_0) > \varepsilon)$ converges to 0 exponentially fast as $M \to \infty$.

We note that while conditions 1 and 2 in Theorem 3 seem 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.

6 NUMERICAL EXPERIMENTS

We conduct numerical experiments demonstrating the practical application of our framework. We compare the performance of the optimal menu, computed approximately via sample average approximation as described in Section 4.1 (Optimal Menu), the variant of this approach that leverages submodular maximization from Section 4.2 (Submodular), and the Thompson sampling approach described in Section 5 (Thompson Sampling). We also compare against a point-estimate approach (Point Estimate) that provides a single solution to the DM by optimizing $U(f(x);\hat{\theta})$ using a single a point estimate $\hat{\theta}$. This point estimate is the expected value of θ under the prior. Finally, we report the performance of an idealized algorithm that knows the DM's true utility (Perfect Information).

6.1 Experimental Settings

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

Static setting The first setting uses a prior on θ whose support is small enough to be fully included in 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 for 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.

Dynamic setting In the second setting,we hold the menu size fixed at 3, varying the number of interactions with the DM. In each replication, we follow these steps:

1. Sample a θ from the prior, which is uniform over the probability simplex, and hold it out, not showing it to the algorithms.
2. Use the algorithm to generate a menu.
3. 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. 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. 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.

Using many replications of the above process, we report the average value of the best item in the first n menus vs. n, along with the standard error.

For algorithms that require a small set of preference scenarios (Optimal Menu ane Submodular), 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, 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.

6.2 Problem Descriptions

Doctor Scheduling This experiment considers a fictional doctor wishing to schedule appointments. There are A patients and B time-slots in which each patient can be scheduled. Patient a assigns a desirability score $c_{a,b}$ to time-slot b. These scores are drawn independently across patients and time-slots from a uniform($[0,1]$) distribution and fixed throughout the experiment. We partition the patients into $J = 5$ disjoint subsets $P_1, \ldots, P_J$ uniformly at random and fix this partition throughout the experiment. Patients in P_j have priority j. A binary decision variable, $x_{a,b}$, indicates whether patient a was scheduled in time-slot b. Attribute j is the cumulative score $f_j(x) = \sum_{a \in P_j} \sum_{b=1}^{B} c_{a,b} x_{a,b}$ of patients with priority j.

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 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 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. Our goal is to support a physician's choice of x_b to control 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 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.

6.3 Results

Figures 1 and 2 below show results for the static and dynamic settings with the doctor scheduling problem in the top row and IMRT in the bottom row. The left column shows the expected quality of the best item found, and the right column shows the computational time used, including the time to sample preference scenarios. Computational time is reported for a 2.5 GHz Quad-Core Intel Core i7 with 16 GB of RAM.

The three proposed algorithms 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 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 Optimal Menu, and offering much better performance than the Point Estimate method. Thus, we offer 3 choices with different trade-offs between solution quality and computational cost from which a user can select based on their circumstances.

In the static setting (Figure 1), 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. An interesting point where IMRT differs from Doctor Scheduling is 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 ground truth values, and all are being given to the algorithm.

In the dynamic setting (Figure 2), 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 performs close to optimal against the true preferences of the DM. Moreover, we note that even a cheap algorithm such as Thompson Sampling, can perform close to optimal after a few interactions with the DM, as it can learn the preferences well enough. 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 finding a menu with higher utility.

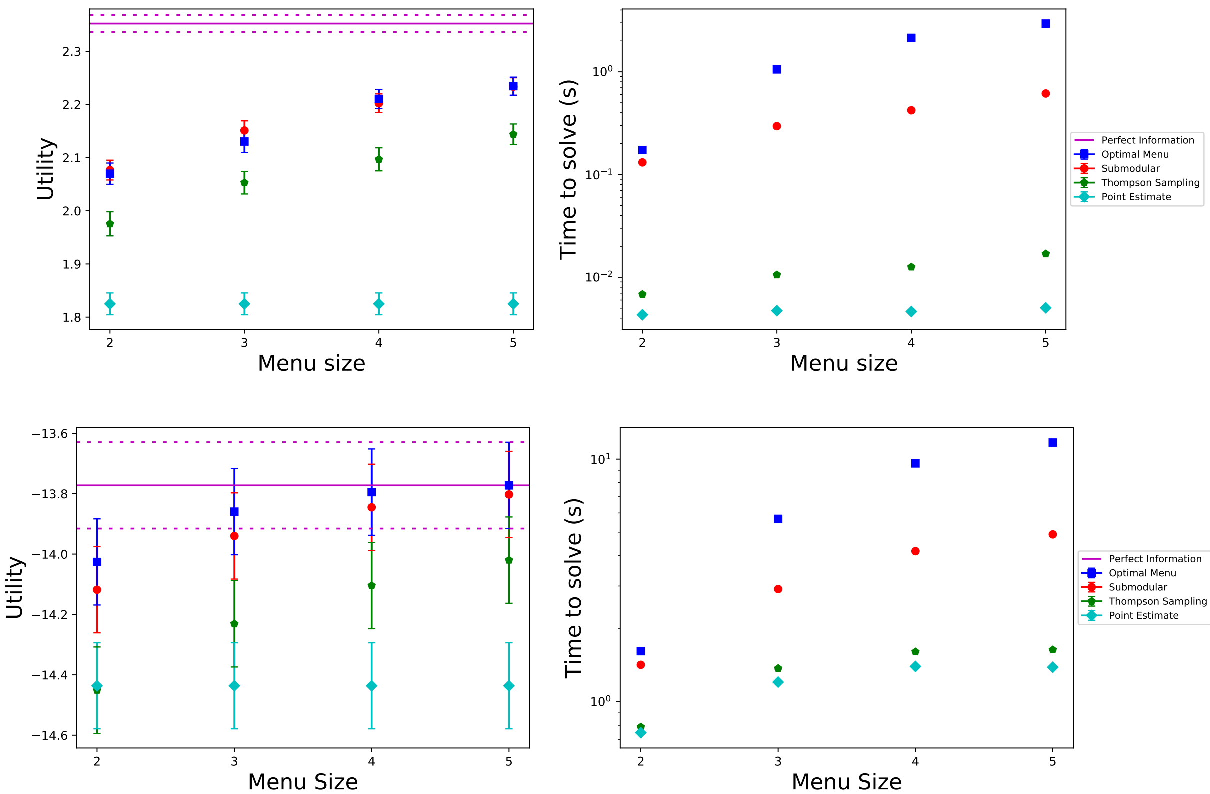

Figure 1: Static setting: Doctor scheduling (top row) uses 8 distinct preference scenarios and 100 replications, while IMRT (bottom row) uses 5 scenarios and 2500 replications.

7 CONCLUSION

We introduced a novel approach to support multi-attribute optimization-based decision-making when the DM's preferences over the 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. Finally, 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 optimization methods.

We proposed a scheme to approximately compute the optimal menu in the mixed-integer linear setting and a Thompson-sampling-inspired approach that only requires maximizing multiple samples from the utility function, making it practical in situations where optimization routines to maximize point estimates of the utility function are available. Our numerical experiments show that these algorithms significantly outperform the traditional point-estimate approach.

A PROOFS OF THEORETICAL RESULTS

Proof. (Proof of Theorem 1) 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 $K \to \infty$.

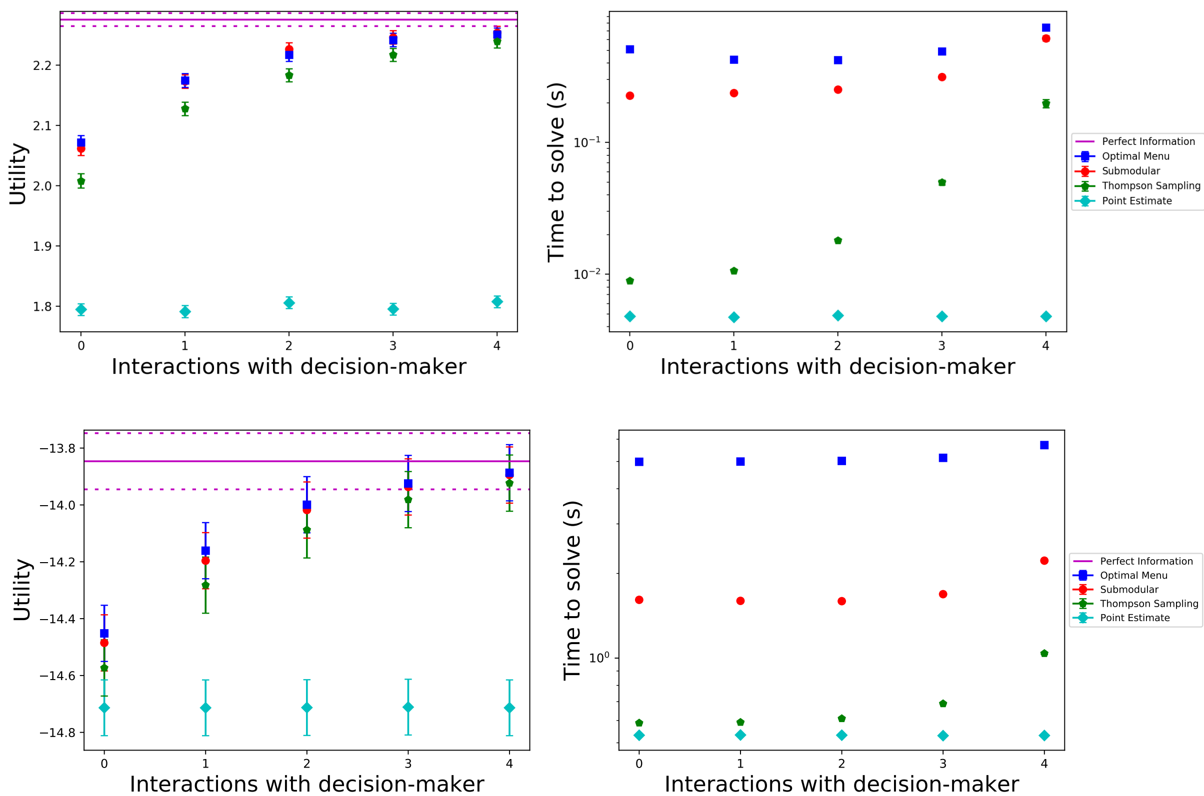

Figure 2: Dynamic setting (with a menu of size 3): We modulate the number of interactions with the DM and update the prior on preference scenarios after each interaction. We use 1000 replications in Doctor Scheduling and 25000 in IMRT.

2. There exists a function $\ell : \Theta \to \mathbb{R}$ such that $|\max_{m=1,\dots,M} U(f(x_m);\theta) - \max_{m=1,\dots,M} U(f(x'_m);\theta)| \leq \ell(\theta) \, \|(x_1,\dots,x_M) - (x'_1,\dots,x'_M)\|_2$ and $\mathbb{E}[\ell(\theta)] < \infty$.

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

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

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

Proof. (Proof of Theorem 3) Note that $\max_{x \in \mathbb{X}} U(f(x);t) = \max_{x \in \mathbb{X}} t^\top C x$ 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 $\varepsilon > 0$, there exists $\delta' > 0$ such that if $\|t - t_0\|_2 < \delta'$, then $|\max_{x \in \mathbb{X}} U(f(x);t) - \max_{x \in \mathbb{X}} U(f(x);t_0)| < \varepsilon/2$.

Moreover, if $\|t-t_0\| < \varepsilon/2\left(\max_{x\in\mathbb{X}}\|Cx\|_2+1\right)$, then we have

$$\begin{aligned}|U(f(x^*(t));t)-U(f(x^*(t));t_0)| &= \left|t^\top Cx^*(t)-t_0^\top Cx^*(t)\right| \\ &\le \|t-t_0\|_2\|Cx^*(t)\|_2 \\ &< \frac{\varepsilon}{2\left(\max_{x\in\mathbb{X}}\|Cx\|_2+1\right)}\|Cx^*(t)\|_2 \\ &< \varepsilon/2.\end{aligned}$$

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

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

Hence, $\{\max_{x\in\mathbb{X}} U(f(x);t_0)-U(f(x^*(\theta));t_0)\ge\varepsilon\}\subseteq\{\|\theta-t_0\|\ge\delta\}$, and thus

$$\mathbb{P}\left(\max_{x\in\mathbb{X}} U(f(x);t_0)-U(f(x^*(\theta));t_0)\ge\varepsilon\right)\le\mathbb{P}\left(\|\theta-t_0\|\ge\delta\right)=1-\mathbb{P}\left(\|\theta-t_0\|<\delta\right).$$

Finally,

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

and thus

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

Now observe that, since Θ is compact, the support function of $\mathbb{X}$ is not only continuous but uniformly continuous over $C^\top\Theta$, and thus δ' above can be chosen independently of t_0, which in turn implies that δ can also be chosen independently of t_0. Therefore, given $\varepsilon>0$, there exists $\delta>0$ such that, for all $t\in\Theta'$,

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

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

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

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

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

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

REFERENCES

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

Alves, M. J. and J. Clímaco. 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 P. Frazier. 2020. "Multi-attribute Bayesian optimization with interactive preference learning". In *International Conference on Artificial Intelligence and Statistics*, 4496–4507. Online: PMLR.

Astudillo, R., K. Li, M. Tucker, C. X. Cheng, A. D. Ames and Y. Yue. 2024. "Preferential multi-objective Bayesian optimization". *arXiv preprint arXiv:2406.14699*.

Astudillo, R., Z. J. Lin, E. Bakshy, and P. Frazier. 2023. "qEUBO: A decision-theoretic acquisition function for preferential Bayesian optimization". In *International Conference on Artificial Intelligence and Statistics*, 1093–1114. 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, 239–246. Austin: AAAI Press.

Boutilier, C., R. Patrascu, P. Poupart, and D. Schuurmans. 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, Department of Computer Science, University of Toronto.

Chajewska, U., L. Getoor, J. Norman, and Y. Shahar. 1998. "Utility elicitation as a classification problem". In *Conference on Uncertainty in Artificial Intelligence*, 79–88. San Francisco: Morgan Kaufmann Publishers Inc.

Chajewska, U., D. Koller, and R. Parr. 2000. "Making rational decisions using adaptive utility elicitation". In *Proceedings of the AAAI Conference on Artificial Intelligence*, 363–369. Austin: AAAI Press.

Chu, M., Y. Zinchenko, S. G. Henderson, and M. B. Sharpe. 2005. "Robust optimization for intensity modulated radiation therapy treatment planning under uncertainty". *Physics in Medicine Biology* 50(23):5463.

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

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

Frazier, P. I. and A. M. Kazachov. 2011. "Guessing preferences: A new approach to multi-attribute ranking and aelection". In *Winter Simulation Conference (WSC)*, 4311–4323 https://doi.org/10.1109/WSC.2011.6148119.

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.

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

Kim, S., R. Pasupathy, and S. G. Henderson. 2015. "A guide to sample average approximation". In *Handbook of Simulation Optimization*, 207–243. Springer.

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

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

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

Yue, Y., J. Broder, R. Kleinberg, and T. Joachims. 2012. "The k-armed dueling bandits problem". *Journal of Computer and System Sciences* 78(5):1538–1556.

Zionts, S. 1979. "A survey of multiple criteria integer programming methods". In *Annals of Discrete Mathematics*, Volume 5, 389–398. Elsevier.

AUTHOR BIOGRAPHIES

BHAVIK A. SHAH is a software engineer at Google. His e-mail address is shah.bhavik627@gmail.com.

RAUL ASTUDILLO is a Postdoctoral Scholar at Caltech. His research focuses on efficient sequential information acquisition. His e-mail address is rastudil@caltech.edu. His website is https://raulastudillo.netlify.app/.

PETER I. FRAZIER is the Eleanor & Howard Morgan Professor of Operations Research & Information Engineering at Cornell and a Staff Data Scientist at Uber. His email address is pf98@cornell.edu. His website is https://people.orie.cornell.edu/pfrazier/.