A Preference Model for Over-Constrained Meeting Requests

Pauline M. Berry and **Melinda Gervasio** and **Bart Peintner** and **Neil Yorke-Smith***

Artificial Intelligence Center, SRI International, Menlo Park, CA 94025 USA
{berry,gervasio,peintner,nysmith}@AI.SRI.COM

Abstract

To have value for an individual tasked with arranging a meeting, a scheduling tool must actively account for the individual's scheduling preferences, especially when the meeting request must be relaxed. We develop a preference model designed to capture user scheduling preferences for over-constrained meeting requests between multiple people, and a methodology for preference elicitation to initially populate this model. The model is built around a 2-order Choquet integral representation. We explain a natural-language-based elicitation of the meeting request details and constraints, and outline the solving of the resulting constrained scheduling problem (with preferences). We then describe the display of solutions to the scheduling problem to the user, as candidate scheduling options with explanations, and detail unobtrusive learning of revisions to the preference model from the user's choices among the candidates. We report on initial assessment of the efficacy of such a preference model in terms of elicitation, learning, and reasoning.

Introduction

All too often, arranging meetings in an office environment is a tedious process. One common method is a series of emails to propose, reject, counter-propose, and eventually agree on a time. The effort consumed in such practices motivates the opportunity for automated assistance in scheduling.

Although a number of fully- or semi-automated scheduling systems have been developed, they have suffered from low adoption rates for two main reasons [12; 27; 31]: they fail to account for the personal nature of scheduling, or they demand too much control of an important aspect of an individual's working world.

The personal nature of scheduling is most directly seen in situations where a user's meeting request cannot be fully satisfied. For example, suppose that Alice requests a 90 minute meeting with Bob and Chris next Tuesday afternoon, but no time is available to all during that period. Some users may prefer the option of a shortened 60 minute meeting, while others (like Alice) may prefer the full 90 minute meeting Tuesday morning. Most users would like to be presented with both options (more generally, with all relevant options, of which there can be many in such *over-constrained* situations), but such that the options they most prefer have priority in the presentation. It is precisely in these difficult, over-constrained situations, in which there may be many competing factors and many possible *relaxations* of the meeting request, where scheduling assistance is most useful. But to be of real value here, an automated scheduling assistant must actively account for a user's scheduling preferences.

In this paper we consider representing and reasoning over the scheduling preferences of an individual. We present a preference model designed to balance the potentially competing loci of elicitation of preferences, refinement of the elicited model instance by machine learning, and reasoning over the preference model to provide options to over-constrained meeting requests. First, for elicitation, more expressive models better capture the nuances of user preferences, but require more effort to specify. Second, for learning, expressive models require significant training (and tend to overfit), while inexpressive models cannot distinguish between candidate schedule options that are distinguished by the user. Third, for automated searching for and ranking of candidate options (hereafter, constraint reasoning), it is more difficult to define and reason over complex objective functions, preferences, and constraints.

While prior research has looked at one or more of these aspects — modelling and eliciting, learning, and reasoning — the resulting systems have rarely sought to encompass all three. For example, representing and learning user preferences but not performing sophisticated reasoning to offer scheduling options [21], or representing preferences and performing constraint reasoning but not updating the preferences by learning [11].

Based on our preference model, the end-to-end semi-automated scheduling assistance enabled by the integration of these three aspects has been implemented within a deployed system called *PTIME* [2]. PTIME, a component of a larger cognitive assistant named *CALO* [25], manages the calendar and the scheduling requests of a CALO user in a mixed-initiative manner. Figure 1 shows the process. PTIME elicits scheduling preferences (step 0); elicits a meeting request (step 1); computes candidate schedules (possibly relaxations) in response to the request, by means of a Constraint Reasoner module, and displays a subset of the candidates to the user as options (step 2); and accepts the

*Contact author

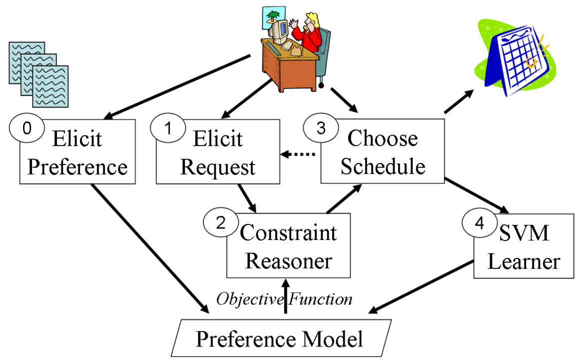

Figure 1: Use of the preference model in PTIME.

user's choice of the desired schedule option (step 3). Based on which option the user chooses for each request among the presented options, the learning module in PTIME updates the parameters of the preference model instance (step 4) [8]. The updated model becomes the basis of reasoning over candidate schedules for the next scheduling request.[1]

At present, the PTIME user organizing the meeting decides which meeting option to select, taking into consideration others participants' generic scheduling preferences. In the future, she will be able to also take others' meeting-specific preferences into consideration. In the simplest form of inter-agent negotiation supported by PTIME, the selected meeting is presented to invitees for inclusion or otherwise in their calendars.

We begin by describing our first user studies and the requirements on the preference model that we derived from them. We then describe a model, based on Multi-Attribute Utility Theory (MAUT) [15] designed to balance expressiveness with amenability for learning and reasoning. Next, we describe the interfaces and design choices for eliciting both an initial instance of the model (*general preference elicitation*), and the specification of and constraints on meeting requests (*problem-specific preference elicitation*). The following sections describe the presentation of schedule options and the learning based on user choices. We report on initial assessment of the suitability of such a preference model in terms of elicitation, learning, and reasoning, and on initial studies of the overall user experience of the PTIME system, noting some of the challenges found in this evaluation.

Development of a Preference Model

An earlier model of user scheduling preferences in the PTIME system, reported in [8], sought to capture temporal preferences (day and time of events) for under-constrained meeting requests. It was composed as a weighted linear sum of features such as meeting start and end times. Building a constraint problem representation and reasoning over it, to find preferred schedule options according to the model, was straightforward. Refining a model instance, likewise, proved effective using *support vector machine* learning techniques [14]. Despite these positive aspects, the model could not capture non-temporal preferences, such as for meeting participants, and its expressiveness was too limited to capture how temporal and non-temporal criteria interact. Compared to the under-constrained case, these two aspects have a significant role for over-constrained meeting requests.

User Studies on Scheduling Habits

We began by investigating the criteria that commonly influence scheduling decisions. We conducted a series of structured interviews focused on how individuals prefer to schedule their meetings and free time, and how they deal with scheduling conflicts. The latter questions were designed to give insight into the trade-offs between different criteria. The study group included managers, administrative staff, and researchers in our organization. We further asked the participants to keep a record of their schedules over a period of a week, and to record the meeting requests they made and received, and how they negotiated meetings with others.

The fivefold foci of the study were *event characteristics* (e.g., one-on-one vs. group meetings), *scheduling processes* (e.g., iterated refinement of a time), *scheduling needs*, *decision factors* (e.g., relationship to meeting host), and *preferences*. In the interest of space, we omit all but the last two.

Factors We found that the subjects take into consideration a handful of factors that are a subset of a broadly common list, when scheduling a meeting and when deciding whether to accept a meeting request. We group the factors into seven categories: (1) importance (of the meeting; its importance to you, your importance to the meeting); (2) urgency/criticality; (3) interest; (4) relationships (host, relationship to host; other participants, relationship to other participants [31]); (5) perturbation (effect on other meetings); (6) stability (often characterized as "number of times the meeting has been rescheduled"); and (7) preferences (of the individual and of others). The relative importance of the features varied considerably between subjects.

Preferences The subjects explicitly indicated preferences for or against some specific features. We group them into four categories: (1) general meeting-specific preferences (e.g., time of day); (2) general calendar-wide preferences (e.g., fragmentation, density, number of meetings per day); (3) preferences over how to relax meeting request constraints (e.g., proximity to specified time, proximity to specified duration, attendance of high-priority participants); and (4) preferences over how to relax calendar-wide constraints (e.g., no overlaps, no Friday p.m. meetings because of childcare situations, no late meetings on carpool days).

Among other studies of user scheduling habits and (semi-)automated calendaring tools, Palen [27], for instance, examined the use in situ of group calendaring software at Sun Microsystems. Several researchers report that people are reluctant to invest in accurately and fully informing a scheduling system of their preferences (to the extent that they can articulate them) unless either (1) the process is not

[1]The dashed arrow from step 3 back to step 2 indicates the user's ability to reject all the presented schedule options and revise her scheduling request. She might do so because she finds none of the options satisfactory, or because seeing the options has stimulated her to refine her request or explore an alternative.

burdensome and they are persuaded of the benefit; or (2) they are mandated to do so [12; 31]. Other studies support our finding that even when people are confident in the behavior and the decisions of a (semi-)automated system, they seek transparency into its reasoning [1; 27].

Our own study and prior work both demonstrate that evaluation of scheduling options hinges on multiple criteria and their interaction [16; 4]. Accounting for the relative importance of these features is crucial if we are to offer the user desirable relaxations for over-constrained requests. Whereas [11; 8; 22] and others assume no interaction between the criteria in their preference model — which simplifies the learning and solving aspects — this choice limits the expressiveness of the model.

A set of requirements for a preference model results from these various ethnographic investigations. First, the model must be populated from intuitive (and thus likely qualitative) statements expressed in terms of concepts that the user is familiar with in the domain, i.e., events, people, and calendars. Second, the expressiveness of the model must be such that it can capture enough of user scheduling preferences to enable the personalized scheduling task. Third, the preference model must be explainable to the user, again in terms of familiar, domain-relevant concepts [21]. Fourth, the preference model must be able to express multiple criteria that factor into the user's decisions, and the interaction between the criteria, such as between the meeting duration and the temporal preferences of other participants. Finally, as we have argued, the model must be tractable for reasoning and learning, if a practical scheduling assistant is to be built.

Choquet Integral Model of Scheduling Preferences

The preferences literature is rich with qualitative and quantitative models of varying expressive power and computational tractability; for surveys, we refer to [26; 9]. The reasoning and learning methods we wish to bring to bear on scheduling problems are quantitative, and the features of the preferences are not independent. Suitable models can be based on, for instance, Generalized Additive Utility (GAI) [6] or Multi-Attribute Utility Theory (MAUT) [15].

To balance the competing aspects of expressiveness and elicitation, learning, and reasoning, and to meet the above requirements, we chose to adopt the global utility function approach of MAUT. By providing a single function by which the system can rate alternative schedules, a MAUT approach is in principle amenable to the schedule evaluation and preference learning components. The challenge of adopting any approach is to build a model suited to the scheduling domain, and to adapt reasoning and learning algorithms to it.

A MAUT model is specified by a set of n criteria, a set $u_1, \ldots, u_n$ of (partial) utility functions that make the criteria commensurate, and an *aggregation function* F. An instance of the model, i.e., a capturing of the preferences of an individual, is specified by the coefficients of the aggregation function. For example, if $z_j = u_j(x)$, where x is item being evaluated, then $F(z_1, \ldots, z_n) = \sum_i a_i z_i$ for coefficients a_k is aggregation by a linear weighted sum.

A weighted sum cannot express interaction between criteria; it assumes that all criteria are preferentially independent [15]. An aggregation function that avoids this assumption and that satisfies certain desirable properties is the *Choquet integral* [10; 17]. The Choquet integral subsumes a weighted sum, and is able to express multi-criteria trade-offs such as Pareto optimal decisions that a weighted sum cannot represent. We now explain how we define a scheduling preference model based on this representation.

From our first user study reported above, augmented by features suggested in prior work in the literature (e.g., [16; 21; 4]), we identified seven criteria consistent across different users: (1) scheduling windows for the requested meetings; (2) durations of meetings; (3) overlaps, ordering constraints, and conflicts between requested and existing meetings; (4) locations of meetings; (5) participants in meetings; (6) time or duration changes for existing meetings; and (7) preferences of others participating in new meetings or rescheduled existing meetings [7].

We deliberately chose criteria expressed in terms of concepts familiar to the user in the domain to facilitate elicitation of instances of the preference model and explanation of learned preferences. Below we describe the formulation of the commensurate utility functions u_i. Other criteria, including the stability of a candidate schedule (how stable it is with respect to new meetings and meetings that run long) and how much the candidate schedule perturbs the existing schedule, would add richness to the preference model. However, as we will describe, we found that the richer model would not be amenable to constraint solving.

With the criteria specified, we next define the Choquet integral aggregation function simplified to this context.

Definition 1. *Let $z_j = u_j(x)$, let N be the set of criteria, and let $\wedge$ denote conjunction. The* Choquet integral *is*

$$F(z_1, \ldots, z_n) = \sum_{I \subseteq N} a_I \bigwedge_{j \in I} z_j \tag{1}$$

where a_I is a coefficient representing the degree and type of interaction of the criteria in I. □

Over n criteria, specification of the general Choquet integral requires 2^n coefficients. A *k-additive* or *k-order* Choquet integral considers only the interactions of criteria sets with k or fewer criteria. It trades expressiveness of the model for its easier specification. Practical applications indicate that the 2-order case is usually sufficient [20; 18]. Only $n + \binom{n}{2} = \frac{n(n+1)}{2}$ coefficients are required to specify the integral. In this case, (1) can be written as

$$F(z_1, \ldots, z_n) = \sum_{i \in N} a_i z_i + \sum_{\{i,j\} \subseteq N} a_{ij}(z_i \wedge z_j) \tag{2}$$

where the coefficients a_i, $i \in N$, and a_{ij}, $\{i, j\} \subseteq N$, fully specify the model. Details of the derivation are given in [20].

In the 2-order case, the coefficient $a_i \in [0, 1]$ describes the relative importance of criterion i (a greater value indicates greater relative importance), while $a_{ij} \in [-1, 1]$ describes the interaction between criteria i and j. $a_{ij} > 0$ indicates that i and j are complementary criteria, while $a_{ij} < 0$ indicates that they are substitutive; $a_{ij} = 0$ indicates no correlation between the two.

We make the hypothesis that a 2-order model trades off expressiveness (being unable to express interaction effects among three or more criteria) with ease of model specification and explanation in a way suitable for (1) a constraint-based representation of the scheduling problem, (2) reasoning over this problem representation, and (3) learning revisions to the model. With $n = 7$ criteria, an instance of the model is specified by $\frac{1}{2}7(7+1) = 28$ coefficients. Note, however, that the reasoning and learning to be described do not depend on a 2-order model, other than their complexity increases as the number of Choquet coefficients increases.

Preference and Problem Elicitation

Both preference and problem elicitation share the common challenges of eliciting preference information from a user. First, information that the user does know and can express she may not be willing to input to the system, especially if the elicitation process is burdensome. Second, the user may not fully know the information or be able to express it (at least via the elicitation mechanism offered). Moreover, the very process of elicitation itself is known to reshape — and at worst bias — the information provided; preferences can even be created by the elicitation process [28; 32].

Preference elicitation One approach to elicitation of general scheduling preferences is to derive rankings of schedule options by showing the user specific examples as part of an elicitation phase. *groupTime* [4] is a scheduling system that takes such an example-driven approach. Viappiani et al. [32] show that presenting users with carefully chosen examples can help stimulate their preferences, a technique known as *example critiquing*.

An alternative approach to elicitation has the user enter the parameters of the preference model directly. The scheduling system of Hayes et al. [11] takes such a model-driven approach. The advantage of this approach is that the elicitation period can be shorter, because the information elicited has greater entropy. The significant disadvantage is that users must comprehend the model, rather than examples that are possibly easier to comprehend.

The coefficients of a Choquet integral, the goal of our elicitation, can be derived from statements over examples (*"I rate this example more highly than that"*) or over the model (*"overlap is more important for me than duration"*) [20].

We chose to explore a form of model-driven preference specification for three reasons. First, we considered that the user's intuition of the scheduling domain reduces the cognitive effort to understand the model, in contrast to the need to digest an artificial example before giving meaningful feedback based upon it. This is especially true for over-constrained examples, where a perception of the calendars and preferences of involved participants is necessary to make an informed judgment over scheduling options. Second, the use of online learning in PTIME updates the model from examples (real-life examples for which the user already has in mind the meeting request, and participants' calendars and preferences, to some degree). Third, we considered it advantageous to enable the user to employ PTIME without an extended elicitation phase, by developing a one-shot, model-driven elicitation.

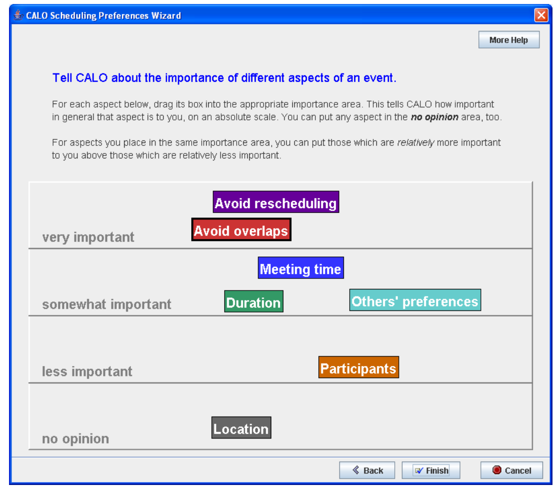

Figure 2: One panel of the preference elicitation interface.

Hence, our approach to preference elicitation combines an interactive, visual interface and a series of simple elicitation *invitations*. Each invitation is presented as a panel, such as in Figure 2, that asks the user to provide some statements of her general scheduling preferences. The panels are presented in a 'wizard' interface that allows the user to freely step forward and backward through them, and to ignore any invitation (i.e., provide no information for it).

From the information entered by the user, we infer qualitative preference statements on and between the criteria, and compile these statements into quantitative coefficients in the Choquet representation. This compilation is performed by solving a linear program (LP); for the details we refer to [20]. The preference statements we infer encompass information regarding both the relative importances of the criteria (i.e., the a_i Choquet coefficients), and the trade-offs between criteria pairs (i.e., the a_{ij} coefficients).[2] In the panel of Figure 2, the classification of the criteria into the four buckets speaks to the first aspect (a_i), and the pairwise relative positions of the criteria speak to the second aspect (a_{ij}).

This two-stage approach of interactive elicitation followed by compilation is designed to blend intuitive, lightweight elicitation for the user, in qualitative, domain-dependent terms familiar to her, with the eventual derivation of quantitative coefficients required by our model. Moreover, the process is robust to the amount of information, falling back to an uninformative, default instantiation of the model in the case of no information.[3]

[2] Since it is well known (e.g., [15]) that people do not act rationally and often do not have consistent desires, the preference statements made may result in conflicts. In these cases, we iteratively relax or remove conflicting statements until the resulting LP is consistent. We judged that any benefit from including the user in this process would be outweighed by the burden of possible added confusion. Relaxation of inconsistent Choquet models is a research topic beyond our scope here [19].

[3] Equal importance ($a_i = 1/n$) to each criterion; no interaction

Problem elicitation Like preference elicitation, our approach to problem elicitation is based on an interactive interface. Common office calendaring tools, such as Microsoft Outlook, obtain meeting details by allowing the user to select a block of time on a calendar interface, and then filling in details in a form. Recent tools, such as Google Calendar, allow the user to specify the meeting details by entering natural language (NL) sentences. In both cases, such tools are not seeking to elicit information in order to formulate and solve a problem of presenting schedule options, but simply to fix a meeting in the user's calendar.

The information for our scheduling task can be both broader (*"An afternoon next week"*) and more specific (*"Bob is an optional participant"*) than that required by calendaring tools. We are seeking to elicit not the details of the meeting itself, but details of the meeting request — a potentially rich set of soft constraints that will be used to formulate a constrained scheduling problem instance. Since the constraints are relaxable, problem elicitation can be seen as elicitation of problem-specific (in contrast to generic scheduling) preferences.

Thus, the deficiencies of common approaches are magnified. On one hand, form-based elicitation varies from restrictive, since the user is limited to the fields in the form, to intimidating, if the form contains enough fields to allow expression of rich constraints. Further, users are observed to feel they must fill in a field simply because of its presence [28]. On the other hand, unrestricted NL can leave the user unsure of what to enter, especially once the illusion that the system really understands everything entered is (inevitably) broken. Moreover, while forms place restrictions or requirements on the user, NL provides little guidance: for example, that the user can specify optional participants.

Figure 3 shows our problem elicitation interface. In the top section of the window we see a system-user dialogue, following DiamondHelp [30]. In the bottom section of the window we see dynamic, context-specific content — here the meeting request interface. Based around an NL input mechanism (centre), the system summarizes the information the user has entered already (bottom), and stimulates her with 'example' lines (top). The design choices and rationale behind them are discussed in [2]. Fundamentally, the scheduling domain scopes the statements the user wishes to input so that restricted NL is practicable, and directs the statements, so that it is intuitive.

Because of the NL interface, the user can readily specify *soft* constraints (i.e., constraints that may be satisfied only to a degree, in contrast to *hard* constraints that must be satisfied exactly), such as preferred times (e.g., *"prefer early"*, or *"prefer 3pm"*), and optional and preferred locations and participants. The user can also flexibly and succinctly specify time windows (e.g., *"tues afternoon"*). Transparency and predictability are enhanced by the system reporting what it understood ("You entered:"); auto-completion reduces the burden and possible mis-spelling of entering locations, participant names, and so on.

The NL interface provides a mechanism for specifying

($a_{ij} = 0$) between each pair of criteria.

Figure 3: Meeting request (problem) elicitation interface.

preferences that is designed to be superior to a form-based direct manipulation input interface. Nonetheless, since easy access to direct manipulation is among the principles for mixed-initiative user interfaces [13], and recognizing that some users find a form-based interface more familiar, we provide an option to "Switch to Form View" that allows direct input of a subset of the possible constraints.

Our interface is suited to specification of straightforward temporal preferences for the meeting, such as a combined preference for the morning over the afternoon, and later in the morning over earlier. It is less suited for specification of finely grained preferences, such as *"later between 10–11am, or as soon as possible after noon, but not between 11–11:30am"*. As the last sentence exemplifies, such highly detailed preferences are cumbersome to describe in text. Rather, a dedicated direct-manipulation interface for such preference would be preferable [5]. We have not implemented this complementary kind of interface, because our first user studies indicated that meetings are rarely requested with such detailed preferences. We think it better for the user to retain these preferences and use them to guide among relaxations of the meeting request, if warranted, when the schedule options are presented by the system.

Altogether, the assisted NL interface was favourably received by users in the restricted setting of our calendaring domain, relative to the direct manipulation interface.

Constraint Reasoning

The aim of constraint reasoning is to generate candidate options in response to a scheduling problem instance that results from problem elicitation (recall Figure 1). This reasoning must account for the preference model of the user, the constraints and preferences in the scheduling problem, and both the current schedule and day/time preferences for all involved participants.

To find desirable solutions according to our chosen preference model, the PTIME Constraint Reasoner must search for candidate schedules using the Choquet integral as the objective function. Thus we have two questions to address: (1) how to represent the hard and soft temporal constraints in a form that translates into the criteria we have chosen; and (2) how to extend the temporal reasoning beyond the simple objective functions found in the literature to the more complicated Choquet integral.

Both questions arise from the core issue of balancing the need for an expressive preference model with difficulty of learning and reasoning about it. In previous quantitative constraint-based temporal optimization work, objective functions were based on simple aggregations of *local preference values* — values indicating how well each constraint is satisfied (e.g., [29]). These objectives do not easily extend to encompass a preference model based on a Choquet integral. First, our model is based on abstract (domain-relevant) criteria, not individual constraints. Second, the model contains interactions between criteria, something not readily expressible in existing frameworks that can accommodate the constraints in our problem, even for the case in which each constraint is considered a distinct criterion [23].

The first facet of our reasoning approach is to map each constraint to a subset of the criteria, based on the origin of each constraint. For example, a soft constraint expressing allowed durations maps to the duration criterion, and a constraint expressing that a person cannot attend two meetings at once maps to the overlap criterion. The mapping is one to many: a constraint can map to multiple criteria.

Formally, we model the constraints as a *Disjunctive Temporal Problem with Preferences* (DTPP) [29]. Adding the mapping from constraints to criteria transforms the DTPP into a *Multi-Criteria DTPP* (MC-DTPP) [23]. To optimally solve an MC-DTPP, a solution must be found that maximizes the value of the Choquet integral equation (2). To optimize over the full integral (which, recall, is a sum of as many as 2^n terms) would be extremely challenging. Every search point would require significant time to calculate the objective value and, more important, it would be difficult to develop effective heuristics or pruning strategies.

This difficulty highlights the trade-off between model expressiveness and tractable reasoning. It motivates our choice of the 2-order Choquet integral, which captures at least one level of criteria interaction, but can be represented using the sum of only $\frac{1}{2}n(n+1)$ terms. It also guides our choice of criteria: criteria such as stability and perturbation mentioned earlier are functions of an entire schedule, and thus do not easily incorporate with the standard DTPP scheme of aggregating over *local* preference values.

The second facet of our reasoning is thus an effective algorithm for solving an MC-DTPP to obtain the scheduling options. The algorithm augments a leading branch-and-bound DTPP solver [24] with special bounding logic and additional heuristics. We leave the details to [23].

Schedule Presentation and Learning

From the many solutions generated by the Constraint Reasoner we want to present a subset — the candidate scheduling options — to the user. On one hand, the system must not overwhelm the user with too many options, but on the other hand, it must present enough options so that the user can select one that is acceptable. In general, we want to present the most desirable solutions first (according to the elicited preference model). However, the model will generally be an imperfect approximation of the user's 'true' preferences, so we also want to present additional solutions that are not necessarily rated highly but that are qualitatively different from those thought most preferred. This approach presents desirable solutions and also enables the user to explore the solution space. With each option that relaxes the meeting request, we provide an explanation in the form of a simplified, natural language summary of the relaxed constraints.

While the elicited preferences provide a starting point for presenting solutions tailored to the user, an effective scheduling assistant should refine its preference model over time. Thus, we employ machine learning techniques. It would be disruptive to the user experience to interrupt with learning questions; therefore, in a deployed setting, learning must happen online using only feedback obtained through a user's natural interaction with the system — i.e., through the user's selections from among the candidates presented; these are the training examples for the learner.

Recall that in our earlier work, the PTIME Preference Learner acquired user preferences over temporal schedule features (i.e., day/time preferences) using *support vector machine* (SVM) learning techniques for ranking [14]. The learner acquired the weights of a linear schedule evaluation function that could be used to rank candidate schedules [8].[4]

Our shift to multi-criteria schedule evaluation fundamentally changes the learning task. While the task remains to learn a schedule evaluation function, the features of the function are no longer boolean features representing temporal properties of a schedule, but instead are real-valued features representing the degree of satisfaction of higher-level criteria u_i, each of which is itself a function of lower-level schedule features, as described earlier. Moreover, the function being learned is a 2-order Choquet integral, rather than a linear weighted sum. This adds the constraints that the coefficients obey $a_i \in [0, 1]$ and $a_{ij} \in [-1, 1]$, as mandated by the Choquet model [17; 20].

Observe that the 2-order Choquet integral can be viewed as a linear weighted sum of a *new* set of features (criteria) comprising affine combinations of the importance coefficients a_i and interaction coefficients a_{ij}. Based on this linearization, we can apply the same SVM learning techniques as earlier to learn the coefficients (weights) for the resulting transformed function, subject to the constraints on the values of the coefficients.

[4]In that work, our initial focus was on learning preferences in under-constrained situations, although the same approach could have been used in principle to learn preferences in the more difficult over-constrained situations as well, by adding features to represent the degree of satisfaction of the different scheduling constraints.

Since both the elicited and learned preferences have the same functional form, we use a simple weighting scheme to combine them into the refined schedule evaluation function

$$F'(Z) = \alpha \times A \cdot Z + (1 - \alpha) \times W \cdot Z \qquad (3)$$

where A is the vector of coefficients (weights) for Z representing the elicited preferences and W are the learned weights for Z. By decaying α over time, we can dynamically modify this relative weighting to give more consideration to the learned weights as the user employs the system.

Evaluation and Ongoing Work

Personalization is a key requirement for adoption of automated scheduling technology. A model of the user's preferences, on one hand, must be expressive enough to capture the salient features that distinguish one scheduling option from another in the user's judgment. On the other hand, the model must be amenable to elicitation to populate instances of it, to explanation, and to reasoning over scheduling requests to derive candidate schedules. Moreover, if the system is to adapt itself by learning, the model must further be amenable to the machine learning techniques employed.

We have implemented the multi-criteria preference model described in this paper within a deployed semi-automated scheduling assistant, *PTIME*. The implementation necessitated a user interface design that exposes the richness of the model without overbearing the user, novel constraint-based reasoning to find optimal schedules according to the model, and adaption of earlier work in non-obtrusive online learning to update the model as the user employs the system.

To assess the value of our approach to the trade-off between representation, reasoning, and learning, we designed a set of experiments and user studies to investigate the following questions: (1) will the learning algorithm converge to a given preference model given consistent feedback; (2) how well do users' elicited preferences match their 'true' preferences; (3) how well does the learner perform using feedback from real users on interesting scheduling instances; (4) does the Constraint Reasoner provide optimal solutions in an adequate amount of time; and (5) how well does the entire system perform in real-world situations?

Space precludes detailed description of our initial evaluation. In brief, we find that (1) the learning converges in theoretical experiments; (2) there is a weak correlation between elicited and 'true' preferences; (3) a lack of data hinders evaluation of the learning in practice; (4) reasoning performance is acceptable with real-life meeting requests, but could be improved; (5) users express satisfaction with the system behaviour. For a detailed discussion of the user studies and computational experiments, their results, and an analysis, we refer to [3].

The inconclusive results of our evaluation to date point to the difficulty of evaluating a model and learning process. Positively, user interviews indicate that our model is expressive enough to capture user scheduling preferences, to a degree sufficient for the types of meeting requests made by knowledge workers in a typical office setting. Although we find that learning obtains an inexact representation of the schedule ranking function (as measured by Spearman's ρ), the refined preference model obtained by the learning becomes adept at suggesting the most preferred schedules.

Users found that the implemented system provides reasonable scheduling options from its first use and exhibits increasing trustworthiness over time — complementary aspects found essential if scheduling technology is to be adopted in practice [21]. Integration with the user's existing calendaring systems and workflow combine with the low demand of the preference elicitation and learning, to further support adoption of the system [27; 2]. This indicates that the model formulation is successful, and that, for the typical use case of PTIME, even poor model instantiations do not degrade user satisfaction.

These positive points said, the experiments undertaken were not able to evaluate the success of the our paradigm of lightweight elicitation of an initial model, and its subsequent non-intrusive refinement. While users found the system's behaviour satisfactory, the evaluation with human subjects did not demonstrate that an elicited instantiation of the preference model is superior to a random instantiation, nor that online learning converges to the user's true model in practical settings (although in artificial settings it does).

Experiments found that an acceptable schedule was usually in the first three presented. Thus, there is limited scope for learning to improve performance. Further, the subjects may have simply selected the first acceptable option, rather than the 'best' option. This hypothesis is supported by [27], which claims that scheduling is often more about "satisficing" instead of "optimizing". Acquiring a model instantiation that reflects the user's preferences is still valuable, since if the first presented option is consistently acceptable, the user may eventually trust the system enough to delegate greater autonomy to it for her scheduling decisions.

In sum, our experiments to date have been unable to determine the correlation or otherwise between the seen satisfactory performance of the system and the model, elicitation, and learning approach underlying it. The reasons for this inability include the small number of subjects in the CALO evaluations, a lack of data for learning to be visible, the difference between "interesting" (roughly, over-constrained) and "uninteresting" (under-constrained) problem instances, and the difficulty of uncovering the user's true preferences or capturing them by approximate metrics.

A central aspect, therefore, of our ongoing work is to design experiments able to answer the above questions. Toward this goal, we have developed a *test harness* to allow rapid gathering of new data and experimentation with various preference models and learning algorithms. The harness will enable more realistic validation and will help characterize how well different model and algorithm combinations work for different types of scheduling instances.

In ongoing work we are considering the use of multiple complementary preference models. For instance, deploying a simple 1-order Choquet model that is more readily elicited and refined, for under-constrained problem instances, in conjunction with the existing 2-order model that is able to capture criteria trade-offs, for critically- and over-constrained instances where the interaction between criteria is a much more significant factor.

On the offline learning side, we are investigating the merits of more direct, albeit heavyweight, elicitation of the Choquet complementarity and substitutability degrees between criteria, via methods from decision theory. On the online learning side, we are investigating hierarchical learning over the criteria within the Choquet u_i functions as well as over the aggregated F function. We are also exploring adaptive presentation of the candidate set of schedule options, including learning what options to present (e.g., varying the set's diversity according to measures informational entropy [33]).

Acknowledgments The authors thank their colleagues from the CALO project, at SRI and elsewhere. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA), or the Department of Interior-National Business Center (DOI-NBC).

References

[1] D. Beard, M. Palaniappan, A. Humm, D. Banks, A. Nair, and Y.-P. Shan. A visual calendar for scheduling group meetings. In *Proc. of CSCW'90*, 1990.

[2] P. Berry, K. Conley, M. Gervasio, B. Peintner, T. Uribe, and N. Yorke-Smith. Deploying a personalized time management agent. In *Proc. of AAMAS-06*, 2006.

[3] P. M. Berry, M. Gervasio, B. Peintner, and N. Yorke-Smith. Balancing the needs of personalization and reasoning in a user-centric scheduling assistant. Technical Note 561, Artificial Intelligence Center, SRI International, Feb. 2007.

[4] M. Brzozowski, K. Carattini, S. R. Klemmer, P. Mihelich, J. Hu, and A. Y. Ng. groupTime: preference-based group scheduling. In *Proc. of CHI-06*, 2006.

[5] A. Faulring and B. A. Myers. Availability bars for calendar scheduling. In *Proc. of CHI-06*, 2006.

[6] P. Fishburn. Interdependence and additivity in multivariate, unidimensional expected utility theory. *Intl. Economic Review*, 8, 1967.

[7] L. Garrido, R. Brena, and K. Sycara. Multiple negotiation among agents for a distributed meeting scheduler. In *Proc. of ICMAS 2000*, 2000.

[8] M. T. Gervasio, M. D. Moffitt, M. E. Pollack, J. M. Taylor, and T. E. Uribe. Active preference learning for personalized calendar scheduling assistance. In *Proc. of IUI'05*, 2005.

[9] R. Goebel, R. Greiner, and D. Lin, editors. *Computational Intelligence: Special Issue on Preferences*, volume 20. 2004.

[10] M. Grabisch, T. Murofushi, and M. Sugeno, editors. *Fuzzy Measures and Integrals*. Physica-Verlag, Heidelberg, 2000.

[11] T. Haynes, S. Sen, N. Arora, and R. Nadella. An automated meeting scheduling system that utilizes user preferences. In *Proc. of Agents'97*, 1997.

[12] K. Higa, B. Shin, and V. Sivakumar. Meeting scheduling: an experimental investigation. In *Proc. of IEEE International Conference on Systems, Man, and Cybernetics 1996*, 1996.

[13] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proc. of CHI-99*, 1999.

[14] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. of KDD'02*, 2002.

[15] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives*. Cambridge University Press, 1993.

[16] R. Kozierok and P. Maes. A learning interface agent for scheduling meetings. In *Proc. of IUI'93*, 1993.

[17] C. Labreuche and M. Grabisch. The Choquet integral for the aggregation of interval scales in multicriteria decision making. *Fuzzy Sets and Systems*, 137(1), 2003.

[18] C. Labreuche and F. L. Huédé. MYRIAD: a tool for preference modeling application to multi-objective optimization. In *Proc. of CP'05 Workshop on Preferences and Soft Constraints*, 2005.

[19] C. Labreuche and F. L. Huede. Inconsistencies in the determination of a capacity. In *Proc. of ECAI'06 Multidisciplinary Workshop on Advances in Preference Handling*, 2006.

[20] J.-L. Marichal and M. Roubens. Determination of weights of interacting criteria from a reference set. *European J. of Operational Research*, 124(3), 2000.

[21] T. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski. Experience with a learning personal assistant. *Comm. of the ACM*, 37(7), 1994.

[22] P. J. Modi, M. M. Veloso, S. F. Smith, and J. Oh. CMRadar: A personal assistant agent for calendar management. In *Proc. of AOIS'04*, 2004.

[23] M. D. Moffitt, B. Peintner, and N. Yorke-Smith. Multi-criteria optimization of temporal preferences. In *Proc. of CP'06 Workshop on Preferences and Soft Constraints*, 2006.

[24] M. D. Moffitt and M. E. Pollack. Temporal preference optimization as weighted constraint satisfaction. In *Proc. of AAAI'06*, 2006.

[25] K. Myers, P. Berry, J. Blythe, K. Conley, M. Gervasio, D. McGuinness, D. Morley, A. Pfeffer, M. Pollack, and M. Tambe. An intelligent personal assistant for task and time management. *AI Magazine*, 28, 2007. To appear.

[26] M. Oztürk, A. Tsoukiàs, and P. Vincke. Preference modelling. In *State of the Art in Multiple Criteria Decision Analysis*. Springer-Verlag, 2005.

[27] L. Palen. Social, individual and technological issues for groupware calendar systems. In *Proc. of CHI-99*, 1999.

[28] J. Payne, J. Bettman, and E. Johnson. *The Adaptive Decision Maker*. Cambridge University Press, 1993.

[29] B. Peintner and M. E. Pollack. Low-cost addition of preferences to DTPs and TCSPs. In *Proc. of AAAI'04*, 2004.

[30] C. Rich, C. Sidner, N. Lesh, A. Garland, S. Booth, and M. Chimani. DiamondHelp: A collaborative interface framework for networked home appliances. In *IEEE Distributed Computing Systems Workshops*, 2005.

[31] J. Tullio, J. Goecks, E. Mynatt, and D. Nguyen. Augmenting shared personal calendars. In *Proc. of UIST'02*, 2002.

[32] P. Viappiani, B. Faltings, and P. Pu. Evaluating preference-based search tools: a tale of two approaches. In *Proc. of AAAI-06*, 2006.

[33] J. S. Weber and M. E. Pollack. Entropy-driven online active learning for interactive calendar management. In *Proc. of IUI'07*, 2007.