Multi-Criteria Evaluation in User-Centric Distributed Scheduling Agents

Pauline Berry, Melinda Gervasio, Bart Peintner, Tomás Uribe, Neil Yorke-Smith

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

Introduction

This position paper discusses the problem of locally evaluating and comparing candidate schedules, in the context of a distributed scheduling task operating in unbounded environments in which each agent selfishly serves the desires and preferences of its own user.

Distributed scheduling systems have used a variety of mechanisms to maximize the quality of the global solution. Techniques include negotiation frameworks based on market economies [Kis et al. 1996], game theoretic algorithms, and global or shared evaluation functions [Modi et al. 2005]. Our position is that these techniques do not adequately address the situation where self-interested cooperative agents maintain schedules on behalf of users operating in unbounded environments. Here, satisfaction is more important than optimality [Palen 1999], and personalized preferences are paramount to users [Berry et al. 2005a; Faulring and Myers 2005]. In our approach, global utility is secondary; the agent aims to maximize the utility of its user, but folds the positive utility of cooperating with others into that utility. Thus, we treat the utility of others as a single component of a personalized and context-dependent multi-criteria evaluation function, which the agent learns through interactions with its user.

PTIME

Our personalized time management system, *PTIME*, is part of the CALO project, a broader research program on cognitive assistance for the office environment [Mark & Perrault 2005]. A CALO agent's primary responsibility is to its user and, as such, PTIME aims to help a user manage her time and to optimize the time spent interacting with others. Issues of privacy, authority, cross-organizational scheduling and availability of meeting participants abound.

Tools and standards for representing, displaying, and sharing schedule information have become common, but the emphasis in the research community is automated meeting scheduling. PTIME is an *open* scheduling system [Ephrati et al. 1994] not operating within organizational boundaries or goals. Other examples include [Franzin et al. 2002; Payne et al. 2002; Modi et al. 2004; Faulring and Myers 2005].

The task that PTIME addresses is not a true distributed scheduling problem, as classically defined, but rather the loose interaction of many local scheduling problems, one for each user. Each local PTIME agent accounts for its user's preferences, workload and deadlines, by formulating the local problem as a soft Constraint Satisfaction Problem [Berry et al. 2005a,b]. Soft constraints extend the classical CSP to support the concept of preferences. Each PTIME may access only the availability of other agents, i.e., times of free timeslots and preferences over them; information beyond this, such as the reason for a user's non-availability, can only be obtained by explicitly querying the PTIME of that user.

Multi-Criteria Schedule Evaluation

The typical scheduling interaction in PTIME involves the user specifying constraints for one or more meetings to be scheduled, PTIME presenting the user with candidate schedules, and, possibly, the user requesting alternatives/ variants before finally accepting one schedule. PTIME aims to facilitate the process of scheduling the meeting(s) by presenting good alternatives. However, the decision whether one candidate schedule is superior to another hinges on multiple criteria. Aside from satisfying constraints and preferences on meetings being scheduled (e.g., "Meeting with Bob and Carol…I'd really like Bob to be there"), a user may have general preferences for when meetings are scheduled (e.g., "I prefer morning meetings") and for the nature of her overall schedule (e.g., "I prefer my meetings back-to-back"). In addition, there are preferences common to all users such as a preference for not moving existing meetings, as well as preferences that vary between the different participants of a meeting (e.g., Bob likes meetings early in the week but Carol prefers them later).

PTIME adopts a multi-criteria schedule evaluation framework. Suppose Alice is scheduling a new meeting of which she is to be the host. Let S be her existing schedule, M the new meeting to be scheduled, S' the resulting new schedule, and M' the actual placement of the meeting M in S'. Note that moving existing meetings opens the possibility of $S'–M' \neq S$, while constraint relaxation on the new meeting request opens the possibility of M being inconsistent with M'. PTIME's evaluation criteria are:

- f_1: satisfaction by M' of Alice's general preferences on meetings (e.g., day of week, time of day)
- f_2: satisfaction by S' of Alice's general preferences on schedules (e.g., degree of fragmentation, #meetings/day)
- f_3: satisfaction by M' of meeting request constraints

- f_4: satisfaction by S' of schedule constraints
- f_5: stability of S'
- f_6: perturbation between S and S'
- f_7: evaluation of S' by other meeting participants

f_1 and f_2 capture respectively how well S' satisfies Alice's general preferences over meetings and schedules, while f_3 and f_4 capture respectively how well S' satisfies constraints on the meeting request and on the overall schedule. f_5 evaluates S' in terms of its ability to accept future changes while f_6 measures perturbation from the current schedule S. Criteria f_1–f_6 all measure the quality of schedule S' only with respect to Alice, while f_7 captures the other participants' assessment of the proposed schedule S' (i.e., its satisfaction of their own preferences and constraints).

The simplest aggregation of these criteria, a weighted sum, is inadequate because it cannot express tradeoffs users have in practice. PTIME aggregates the criteria using methods from multi-criteria decision theory, such as finding Pareto optima. Thus, Alice is presented a set of partially-ranked candidate schedules and, through the user interface, she can refine or relax her input constraints (i.e., both M and S), to explore alternative solutions or select a preferred candidate. A similar user-centric approach is advocated by [Faulring and Myers 2005]. This exploration is supported by an explanation facility that provides visibility into why constraints may have been violated and how a new variant differs from the original.

In deciding which candidates to present to the user, PTIME has multiple objectives: (1) present desirable schedules the user is likely to accept, (2) present qualitatively different schedules to help the user explore the space of possibilities, and (3) support faster learning of preferences over criteria. We are now exploring search techniques that can produce not only a single optimal solution, but also a good set of qualitatively different solutions to meet these objectives.

The Opinion of Others

Our informal user studies confirm that other participants' constraints and preferences often influence the meeting organizer's decisions [Palen 1999]. Within our multi-criteria evaluation, these concerns are captured by f_8, which is key to the distributed aspects of the scheduling problem.

The formation of f_7 is context-dependent. At its simplest, f_7 is the sum of the other user's assessments of S', weighted according to the importance Alice attaches to their opinion: $f_8(S') = \Sigma_{i \in M} \lambda_i \, v_i(S_i')$, where v_i is the assessment of user i. and S_i' is user i's new schedule.

Clearly, a requirement here is for Alice's PTIME to have knowledge of the values $v_i(S_i')$. This requires communication with the other PTIME agents, and their willingness to provide this information. In turn, this leads to wider CALO issues of privacy and distributed communication (e.g., another user has his CALO switched off). If Alice's PTIME cannot ascertain one $v_i(S_i')$, it by default falls back to estimation of the value. Such issues are a further hindrance to classical distributed scheduling approaches to this scheduling situation.

Given a single candidate schedule, selected by its user, or proposed by another PTIME agent, PTIME provides a numerical value that expresses the combined evaluation of the criteria on this schedule. Hence, PTIME can respond when queried by another agent for its value v_i(S'_i).

Learning

Critical to the mixed-initiative goals of PTIME is the ability to use the learned knowledge of user preferences within the underlying CSP. Our initially deployed system focused on learning an individual's preferences over temporal constraints. For example, does Alice prefer morning meetings? Our current work is focused on learning a user's preferences between criteria—does Alice lend more weight to the preferences of her manager than her desire to keep her mornings free?

Acknowledgments. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010.

References

Berry, P. M.; Gervasio, M.; Uribe, T.; and Yorke-Smith, N. 2005a. Mixed-Initiative Issues for a Personalized Time Management Assistant. In *Proc. of the ICAPS'05 Workshop on Mixed Initiative Planning and Scheduling*.

Berry, P. M.; Gervasio, M.; Uribe, T.; and Yorke-Smith, N. 2005b. *Multi-Criteria Constraint Solving and Relaxation for Personalized Systems*, Technical Report, SRI International.

Ephrati, E.; Zlotkin, G.; and Rosenschein, J.S. 1994. A non-manipulable meeting scheduling system. In *Proc. of the 13th Intl. Distributed Artificial Intelligence Workshop*.

Faulring, A., and Myers, B. A. 2005. Enabling Rich Human-Agent Interaction for a Calendar Scheduling Agent. In *CHI'05*.

Franzin, M. S.; Freuder, E. C.; Rossi, F.; and Wallace, R. 2002 Multi-agent meeting scheduling with preferences. In *Proc. of the AAAI 2002 Workshop on Preference in AI and CP*.

Kis, T.; Va'ncz, J.; and Ma'rkus, A. 1996. Controlling Distributed Manufacturing Systems by a Market Mechanism. In *Proc. of ECAI 96.*

Mark, B., and Perrault, R. C. 2005. *CALO: Cognitive Assistant that Learns and Organizes.* http://www.ai.sri.com/project/CALO

Modi, P. J.; Veloso, M.; Smith, S.; and Oh., J. 2004. CMRadar: A Personal Assistant Agent for Calendar Management. In *Agent Oriented Information Systems*.

Modi, P. J.; Shen, W.; Tambe, M.; and Yokoo, M. 2005. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence* 161:149–180.

Palen, L. 1999. Social, Individual and Technological Issues for Groupware Calendar Systems. In *Proc. of CHI'99*.

Payne, T. R.; Singh, R.; and Sycara, K. 2002. Rcal: A case study on semantic web agents. In *Proc. of AAMAS'02.*