A Formal Study of Distributed Meeting Scheduling

Sandip Sen
Department of Mathematical & Computer Sciences
University of Tulsa
600 South College Avenue
Tulsa, OK 74104-3189
sandip@kolkata.mcs.utulsa.edu

Edmund H. Durfee
Department of EECS
University of Michigan
1101 Beal Avenue
Ann Arbor, MI 48109
durfee@umich.edu

Running head: Distributed Meeting Scheduling

A Formal Study of Distributed Meeting Scheduling

Sandip Sen
Department of Mathematical & Computer Sciences
University of Tulsa
Tulsa, OK 74104-3189
sandip@kolkata.mcs.utulsa.edu

Edmund H. Durfee
Department of EECS
University of Michigan
Ann Arbor, MI 48109
durfee@umich.edu

Abstract

Automating routine organizational tasks, such as meeting scheduling, requires a careful balance between the individual (respecting his or her privacy and personal preferences) and the organization (making efficient use of time and other resources). We argue that meeting scheduling is an inherently distributed process, and that negotiating over meetings can be viewed as a distributed search process. Keeping the process tractable requires introducing heuristics to guide distributed schedulers' decisions about what information to exchange and whether or not to propose the same tentative time for several meetings. While we have intuitions about how such heuristics could affect scheduling performance and efficiency, verifying these intuitions requires a more formal model of the meeting schedule problem and process. We present our preliminary work toward this goal, as well as experimental results that validate some of the predictions of our formal model. We also investigate scheduling in overconstrained situations, namely, scheduling of high priority meetings at short notice, which requires cancellation and rescheduling of previously scheduled meetings. Our model provides a springboard into deeper investigations of important issues in distributed artificial intelligence as well, and we outline our ongoing work in this direction.

Keywords: meeting scheduling, intelligent agents, distributed AI, probabilistic model, discrete event simulation

1 Introduction

Computer networks that support human organizations provide an infrastructure for improving group performance through an array of collaboration tools, such as electronic mail systems and shared file systems. While such tools help people share, access, and manipulate more information, they can also impair human performance through overuse or abuse from the propagation of unnecessary information. Techniques from artificial intelligence (AI) can introduce "intelligent agents" into organizational computing systems, where these agents use knowledge about the interests and priorities of people to perform routine organizational tasks such as automatically screening, directing, and even responding to information (Hewitt and Inman 1991; Malone *et al.* 1987; Pan and Tenenbaum 1991).

Meeting scheduling is a good candidate for automation because it is often tedious, iterative, and time-consuming for people. Even when everyone involved in a meeting has available times to meet, the process of searching for a commonly available time in the presence of communication delays (either through electronic mail or in contacting by phone), and in the presence of other meetings being scheduled concurrently, can be frustrating and lead to suboptimal solutions. Automating meeting scheduling is important, therefore, not only because it can save time and effort on the part of humans, but also because this may lead to more efficient schedules and to changes in how information is exchanged within organizations (Feldman 1987). Past efforts (Greif 1982; Malone *et al.* 1987) in developing automated meeting schedulers have met with limited success, although they are available in a number of office software systems (Grudin 1987; Kinkaid, Dupont, and Kaye 1985). Ideas borrowed from experimental studies on how humans handle scheduling (Kelley and Chapanis 1982) may suggest strategies to be used in more efficient, popular automated meeting schedulers.

Our work is directed toward developing intelligent agents that can negotiate over scheduling options on behalf of their associated humans (Sen 1993) . Under routine circumstances, these agents should converge on acceptable meetings unaided, while in more complex situations (requiring cancellations or complicated tradeoffs) the agents should aid their associated human users to schedule meetings better and faster. Having evolved out of a tradition of work in Distributed AI (DAI), our approach views meeting scheduling as a distributed task where a separate calendar management process is associated with each person in order to increase reliability and exploit inherent parallelism. Moreover, giving each person his or her own process enhances privacy and permits personal tailoring of preference parameters for scheduling meetings. But because the information about available times is distributed among processes that wish to minimize how much information they reveal, arriving at meeting times involves selective, distributed search (Durfee and Montgomery 1991). Our research is concerned with how to implement and control that search, and in this paper we develop formal tools for analyzing alternative heuristic strategies that affect the performance and efficiency of the overall search process.

Before going further, we should also make clear what our research, to date, is *not* about. Our DAI emphasis on studying coordination among artificially intelligent agents that are solving collective problems ignores crucial human factors issues concerning how to transfer the preferences and constraints of a person into that person's associated scheduling process (Dent *et al.*, 1992; Maes and Kozierok 1993). In addition, in this paper, we do not address the problem of developing a human interface to the system that will make people want to use it while at the same time limiting their ability to abuse it (we are, however, currently working on a preliminary version of such an interface). Our hope is that our approach of distributing information and control among the different processes will both make it easier for a person to tailor his or her own scheduling agent and make it harder for a person to access or influence processes that belong to others. Moreover, our research does not involve the modeling of the cognitive processes used by humans to schedule meetings. We are interested though in building scheduling agents that use negotiation strategies understandable and acceptable to the human user. Our probabilistic analysis of alternate heuristic strategies help build automated agents that can make smarter decisions based on the calendar state, as well as provide guidelines to users to tailor the scheduling agents to schedule meetings according to their own preferences. The other point that we need to clarify is that we are not trying to derive any closed form solution to the dynamically arriving meeting scheduling problem. Actually, we believe any unique "neat and optimal" solution to this problem does not exist. Our effort in this work is in predicting the expected efficiency of different reasonable scheduling heuristics under a variety of resource constraints.

When information and control are decentralized, collective achievement of goals (such as scheduling meetings) is complicated because the agents must exchange enough information to converge on consistent agreements of how they (or in this case, their associated users) will behave, and must also decide on how to respond to additional scheduling requests that arrive while they are scheduling other meetings. To begin with, we have identified several heuristic strategies that intuitively address these questions (Section 2). To analyze more fully how the alternative strategies affect performance and cost of meeting scheduling, however, we have had to model the distributed meeting scheduling task more formally (Section 3). Our distributed scheduling procedure is based on the multistage negotiation protocol (Conry, Meyer, and Lesser 1988) (Section 4). To aid the choice of appropriate scheduling strategies, we have analyzed the proposed alternative strategies in detail to justify our intuitive predictions about how they will affect performance (Section 5). We used a testbed that simulates concurrent meeting scheduling activities among agents to experimentally verify our predictions (Section 6), and based on our results we have identified important directions for our ongoing research (Section 7).

2 Heuristic Strategies

Meeting scheduling is an example of a resource allocation problem, where the principal resource under consideration is peoples' time. Resource allocation in a distributed, dynamically changing environment is difficult due to the distribution of information needed for decision making, the dynamic nature of the system (which may lead to changing/conflicting goals), the limited bandwidth of the communication channel, and communication delays between parts of the system. Centralizing control in a single resource allocator suffers from serious drawbacks including a lack of robustness (if the allocator fails, the entire system collapses) and its communication and computation demands on a single bottleneck process. To overcome the limitations in centralized resource allocation, we instead distribute decision making among the processes controlling the separate resources. Distributed resource allocation, therefore, involves the cooperative solution of resource allocation problems among a network of decision makers, and falls into the subclass of DAI known as Cooperative Distributed Problem Solving (CDPS) (Durfee, Lesser, and Corkill 1989).

Individual agents in a Distributed Meeting Scheduling (DMS) system have only partial knowledge of system-wide goals because they do not know all the meetings that are currently being scheduled or have already been scheduled by other agents. Hence they must exchange relevant information to build local schedules that fit into a globally consistent schedule. To enable information exchange, the agents need a common communication protocol for negotiating over meeting times. Moreover, because users must be able to understand and accept how the agents interact, the protocol must be well-defined and straightforward while still providing sufficient flexibility. In cases where the scheduling agents cannot converge using their protocol, they alert their associated human users who can step outside of the system and use whatever protocol they desire to schedule their meetings. For our agents, we have chosen to adapt the multistage negotiation protocol (Conry, Meyer, and Lesser 1988), which is a generalization of the contract net protocol (Smith 1980). In our protocol, each meeting has a particular agent who is responsible for it, called the host. The host contacts other attendees of the meeting (who are called invitees) to announce the meeting, and collects bids (availability information). This process could be repeated several times as they search through different parts of their respective calendars. In between, other meetings could be undergoing scheduling; in general, an agent can simultaneously be involved in scheduling any number of meetings, acting as a host for some and an invitee for others.

How well this protocol performs in efficiently converging on good schedules is strongly impacted by heuristic strategies about what information to exchange and how to model tentatively scheduled meetings. Strategies for communication must balance demands for privacy (which lead to exchanging less information) with demands for quickly converging on meeting times (which can be sped up by exchanging more information). Strategies for modeling tentatively scheduled meetings can range from blocking off tentative time(s) for a meeting unless and until the arrangements fall through, to ignoring tentative commitments about a meeting when scheduling other meetings. Our initial exploration of DMS has involved analyzing the positive and negative aspects of these types of strategies, under a variety of conditions. Specifically, we have embarked on research to develop, analyze, and verify a formal model of DMS to formulate rigorous, quantitative predictions of the performance of the following types of heuristic strategies:

Announcement Strategies determine how a meeting is announced, and usually involve proposing some number of possible times. We specifically consider the options called **best** (where only the best meeting time from the host's perspective, ranked by some heuristic like being the earliest, is communicated) and **good** (where several times preferred by the host, 3 by default, are communicated).

Bidding Strategies determine what information an invitee sends back based on an announcement. We consider the options called **yes_no** (where an invitee says yes or no to each proposal sent by the host) and **alternatives** (where an invitee proposes the nearest later time(s) when it can meet). In this paper, we assume the heuristic of scheduling meetings as early as possible. Sending an earlier time is useless in this case, because had that time been available to the host it would have been proposed already.

Commitment Strategies are **committed** (whenever a time is proposed by an agent (host or invitee), that agent tentatively blocks it off on its calendar so no other meetings can be scheduled there) and **non-committed** (times are not blocked off until full agreement on a meeting time is reached).

3 A Formal Definition of the DMS Problem

Before presenting a definition of the DMS problem we need to clarify that it is not our intent to capture all the different constraints, preferences, biases that are possibly used by humans to schedule meetings. In the following we will try to identify what we believe is the core of the meeting scheduling problem. For discussions in this paper, we will use this core model. In other work, we have extended this definition to include cancellation and rescheduling of meetings, scheduling rooms for meetings, scheduling meeting with subset of specified attendees, etc. (Sen 1993, Sen and Durfee 1994b).

A meeting schedule consists of a group of meetings for a group of persons. Given a set of n meetings and k attendees, a scheduling problem is represented as $\mathcal{S} = (\mathcal{A}, \mathcal{M})$, where $\mathcal{A} = \{1, 2, \ldots, k\}$ is the set of attendees and $\mathcal{M} = \{m_1, m_2, \ldots, m_n\}$ is the set of meetings to be scheduled. A time *slot* is represented as a date, hour pair $\langle D, H \rangle$. A set of contiguous time *slots* is called a time *interval*. A meeting named i is represented by a tuple:

$$m_i = (A_i, h_i, l_i, w_i, S_i, a_i, d_i, f_i, \mathcal{T}_i),$$

where

$A_i \subseteq \mathcal{A}$, is a set of attendees of the meeting;

$h_i \in A_i$, is an attendee who will host the meeting;

l_i is the required length of the meeting in hours;

$0 < w_i \leq 1$ is the weight or priority assigned to the meeting;

S_i contains a set of possible starting times on the calendar for the meeting. If $|S_i| = 1$ the meeting is said to be *constrained* (the exact interval to be used for the meeting is pre-specified); if S_i includes all time slots on the host calendar (assuming that it was empty) from which a meeting of length l_i can be started, then the meeting is said to be *unconstrained*; otherwise, the meeting is *semi-constrained*;

a_i is the time at which h_i becomes aware of the need to schedule m_i;

d_i is the deadline by which the host h_i needs to schedule the meeting m_i;

f_i is the time at which the final decision is made about whether m_i can be scheduled and, if so, when it will take place;

$\mathcal{T}_i$ is the time interval for which the meeting m_i is finally scheduled and is represented by an ordered set $\{\langle D_i, H_i\rangle, \langle D_i, H_i + 1\rangle, \ldots, \langle D_i, H_i + l_i - 1\rangle\}$, (here D_i gives the date and H_i gives the starting hour for which meeting m_i is scheduled) if the meeting could be scheduled, and by $\emptyset$ otherwise.

3.1 Criteria for Algorithm Evaluation

The DMS problem involves scheduling the meetings in $\mathcal{M}$, subject to all constraints. Any algorithm used to solve the above problem will incur overhead in terms of communication cost and time required for scheduling the meeting (since time can be viewed as the resource that is being scheduled, time spent scheduling meetings is 'wasted'). Hence, the problem statement can be modified to say that the desired goal is to schedule as many important meetings as possible in $\mathcal{M}$, expending as little overhead (in terms of communication cost and processing time) as possible.

We thus evaluate alternative strategies in terms of two first-order metrics, namely, performance and efficiency. Performance reflects the degree to which the strategy succeeds in scheduling important meetings. Efficiency reflects the cost incurred for attaining that level of performance.

In DMS, global performance can be measured by the weighted **success ratio** in scheduling n meetings:

$$\eta = \frac{\sum_{i=1}^{n} w_i * \rho_i}{\sum_{i=1}^{n} w_i}$$

where

$$\rho_i = \begin{cases} 1 & \text{if } m_i \text{ has been scheduled} \\ 0 & \text{otherwise.} \end{cases}$$

Global efficiency can be measured in terms of two parameters, communication cost and the total time taken to schedule meetings. Communication cost is proportional to the number of packets exchanged by the agents while scheduling the assigned meetings. The time taken to schedule a meeting includes both processing and communication time, and can be calculated from f_i and a_i which can be obtained from system clocks. For simulation experiments, however, we used other rough measures for time taken to schedule meetings. For most experiments, we use the *number of negotiation iterations* taken by agents to schedule a meeting as a measure of time. In order to roughly estimate the processing time involved in finding intervals to propose, we used the *slots searched* metric, that counts the number of calendar slots checked (to see if an interval of the desired length is free starting at that slot) by all the attendees of the meeting. The above measures are practical in the sense that, in real life, some users may not have access to automatic schedulers, or else may decide to schedule some meetings themselves. Strategies that reduce the number of iterations and/or the number of proposals required to schedule meetings will be essential for such systems. In actual practice then, the efficiency of an automated meeting scheduler will be calculated as a function of both cost of communication and total time for scheduling.

4 Meeting Scheduling Protocol

In building scheduling agents, our goal is to balance the need to have a flexible routine to handle a range of situations, while still keeping the routine well-defined and understandable enough to be embraced by a user. We have therefore chosen to initially examine the multistage negotiation protocol. While it does not capture some of the sophistication that people might employ in exceptional circumstances, it fulfills our initial needs by providing a flexible algorithm for scheduling routine meetings. In particular, it assigns ultimate authority for scheduling a meeting to a single process (the host) while still permitting invitees to have input in the scheduling decisions through the use of counterproposals.

The protocol given below allows users to block off times for themselves; these requests are simply processed as single user meetings. We also do not assume that agents are completely cooperative. They may refuse to accept a proposal from a host even though the corresponding time slot was available. We are currently studying mechanisms that can be used to make such decisions using user preferences and biases. The actual protocol followed by agents is as follows:

1. *When a meeting needs to be scheduled, the host tries to find time intervals in its schedule that suit the constraints of date and time (when these are unspecified, the host tries to find the earliest intervals in its calendar where the meeting will fit). If it cannot find any intervals, it fails and the meeting is abandoned. Otherwise, if it is the only participant in the meeting, it schedules the meeting for the best (earliest) interval. If there are other participants, the host announces the meeting to them by proposing one or more of the earliest intervals it found.*

2. *Each invitee receives the announcement proposal(s), and tries to find local solutions and send them back as bids to the host. Bids consist of time interval(s) for which the bidder (the invitee) can schedule the announced meeting. The time interval(s) sent as bids can simply be the subset of those announced by the host that is (are) free on the invitee's calendar, or they can be completely different from those the host announced. A bid can thus be a counter-proposal for when to meet.*

3. *The host collects and evaluates these bids. If the bids suggest a common time interval which is open for the host also, the meeting can be scheduled and the host sends awards to the bidders. If the meeting cannot be mutually scheduled yet, the host comes up with new proposals depending on the bids it received and its own calendar and sends these off to the bidders. It also sends rejections for the bids received.*

4. *When the bidders receive new proposals they reply as above. On receiving an award, they check to see if the time intervals are still free. If so, they mark their calendar, recording the scheduling of the meeting. Otherwise they send back rejections.*

The above algorithmic steps are repeated until a satisfactory schedule is arrived at or it is recognized that the meeting cannot be scheduled (due to an over-constrained schedule or due to the fact that the meeting could not be scheduled before the deadline).

A given meeting m_i is scheduled through negotiation by $|A_i|$ processes, one for each of the attendees of the meeting. The process created by the jth attendee to meeting m_i is referred to as M_{ij}. A process M_{ij} will interact with other processes in two possible ways. First, because agents attending a meeting must exchange information to converge on a schedule, M_{ij} will interact via communication with other M_{ik} processes — the processes forked by other attendees of task m_i. Second, because the same agent may simultaneously fork several processes to schedule different meetings it plan to attend, M_{ij} will interact with M_{lj} for other meetings m_l that are being concurrently negotiated by agent j—this interaction takes place in the processes' contention for the shared calendar for the corresponding resource. The latter form of interaction can be particularly detrimental to the quality of schedules generated, and we examine the nature of these interactions in more detail in the next section.

4.1 Interaction Via Shared Resources

In this section, we identify different modes of interaction between two processes that are sharing the same resources (an agent's calendar) to schedule meetings. Scheduling inefficiency arises due to interaction via shared resources when two such processes try to use overlapping time intervals to schedule their respective meetings. The particular scheduling strategy choice that plays a major role in affecting these types of interaction is the choice of the **commitment strategy**.

The choice of committing or not committing to a proposed time interval amounts to either blocking or not blocking valuable calendar resources until complete agreement is reached. Commitment can cause non-optimal schedules as some meetings block time intervals that cause other meetings to be abandoned due to lack of uncommitted times within the meeting's constraints. Those blocked intervals might be released later. On the other hand, blocked time intervals prevent attempts to propose overlapping time intervals for two different meetings, which can save scheduling time and the amount of information exchanged to schedule meetings. So, although the primary effect of commitment is on the success ratio performance criteria, this strategy choice also affects the total time and proposals exchanged in the scheduling process.

Viewing a commitment strategy as affecting the interaction between processes that require common resources, we can formally represent the resource requirements of process M_{ij} as a 4-tuple,

$$\Re(i,j) \;=\; (v_{ij}, p_{ij}, b_{ij}, r_{ij})$$

where

- v_{ij} represents the set of all *viable* time intervals that could have been proposed for meeting m_i by individual j. It is given by the set of all time intervals of length l_i whose starting slot belongs to S_i, the set of possible starting times on the calendar for meeting i.

- p_{ij} represents the set of time intervals that have been *proposed* by individual j and are still being considered for meeting m_i. $p_{ij} \subseteq v_{ij}$, since only viable time intervals are proposed.

- b_{ij} represents the set of time intervals that have been *blocked* for probable use by individual j for meeting m_i. These time intervals are under active consideration, but at most one of these will be used for the meeting. $b_{ij} \subseteq p_{ij}$, since only a subset (possibly empty) of the proposed time intervals can be blocked.

- $r_{ij} \;=\; \mathcal{T}_i$ if m_i has been scheduled (represents the time *reserved* for the meeting), and is $\emptyset$ otherwise. $r_{ij} \subseteq p_{ij}$, since the finally reserved time interval for a meeting is one on which all attendees have agreed and hence must have been proposed.

For process M_{ij}, v_{ij} represents the static part of resource requirement, p_{ij} and b_{ij} are the dynamic parts, and, assuming no cancellation, r_{ij} changes at most once (from $\mathcal{T}_i = \emptyset$ to $\mathcal{T}_i \neq \emptyset$ if $\rho_i = 1$) during the lifetime of the process.

5 Predictions

To aid the process of selection of appropriate heuristic strategies for use, it will be useful to analytically develop predictions about the performance and efficiency of the protocol using the different heuristic strategies outlined in Section 2. In this section, we present of the predictions that we have developed to date. To validate that our analytical models correctly capture the essence of the problem, in Section 6 we describe experimental work in verifying the predictions.

In the following sections, we examine in turn:

Announcement Strategies: If the host proposes more possible meeting times to invitees, how does this affect the expected number of announcement-bid iterations required to schedule the meeting? In turn, how does this affect the time and communication needs of scheduling?

Bidding Strategies: If an invitee sends back its best available time(s) rather than just vetoing the time(s) proposed by the host, what is the expected savings in iterations and what is the expected communication cost?

Commitment Strategies: How will the success ratio be affected by committing to tentative meeting times, and how will the relative effects depend on characteristics of the scheduling task?

We now present some qualitative predictions based on our model:

- The influence of announcement strategies **best** and **good** is reflected in the amount of information that is exchanged between the host and the invitees of a meeting. Clearly, the more information the host process has about the states of the other attendees, the better the quality of its decisions. Similarly, the invitees also get more information from the host and are able to provide more informative replies which help in identifying a satisfactory schedule for the meeting more quickly.

- The choice of bidding strategies **yes_no** or **alternatives** is reflected only in the amount of information the invitees of a meeting send back to the host. If the host gets back less information, the quality of its decisions can be affected.

- The choice of commitment strategies **committed** or **non-committed** is reflected by the calendar resources used ($\Re(i, j)$ for meeting i by attendee j). Commitment to a proposed time interval involves blocking it off (so that it cannot be proposed for any other meetings) until one hears back from the other individuals to whom these time intervals were proposed. Only the **committed** strategy incorporates blocked times; for a **non-committed** strategy, the set of blocked times is always empty. Commitment is a conservative strategy that avoids overcommitting times, at the risk of underutilizing resources and, thus, form suboptimal schedules. A non-committed strategy is more aggressive, often leading to better utilization of time but sometimes leading to situations where an agent backs out of a proposed meeting, which can create a chain reaction of meeting cancellations.

For the **announcement strategies** we develop predictions for the number of iterations of information exchange required to schedule meetings and the associated communication cost incurred. The actual time taken to schedule meetings, a criterion we mentioned for algorithm evaluation in Section 3.1, is directly proportional to the number of iterations required to schedule the meeting. The actual communication cost, as mentioned in Section 3.1, is obtained by multiplying the communication cost difference (from the following section) with the number of attendees of the meeting. Similar considerations hold for the predictions developed for **bidding strategies**. For both of these strategies, we have considered only *unconstrained meetings* (meetings for which neither data nor hour is fixed ahead of time), and sufficiently long calendars so that they can always be scheduled, and hence success ratio of scheduling these meetings is 1. For the predictions involving **commitment strategies** (Section 5.3), we consider only *constrained meetings* (date and time fixed), and develop predictions for the success ratio criterion. The number of iterations required is always 1 (since there is one interval to check for) and hence the communication cost is also fixed.

In order to formally characterize the behavior of the heuristic strategies, it was necessary to resort to certain assumptions and simplifications. We now present some of these assumptions and simplifications underlying our probabilistic analysis. The fundamental assumption is that the scheduling strategies used capture a regularity in information processing that can be subjected to probabilistic analysis. This assumption may not hold for the scheduling methods used by a human, but becomes realistic when we use the multistage negotiation protocol for meeting scheduling. We assume that different iterations used in scheduling a particular meeting are independent events, and as such, the probability that a meeting could be scheduled in one iteration is independent of its position in a sequence of iterations. This simplifying assumption does not strictly hold, and gives rise to the discrepancies observed between expected and experimental values of certain performance metrics (Section 6). The densities of the agents' calendars are assumed to be stable over the negotiation time for a meeting. Also, given a particular density of an agents' calendar, the meetings on the calendar are assumed to be uniformly distributed. We have often used expected values of random variables instead of probability distributions for comparison of different strategy choices.

5.1 Announcement Strategies

5.1.1 Total Time.

We begin by determining the number of iterations taken on the average to schedule a meeting using the **best** strategy. Because the **best** strategy transmits exactly 1 proposed time per iteration, we represent the number of iterations for this strategy as I_1. Let p_x be the probability of success that xth meeting could be scheduled in one iteration using one proposal, and let $q_x = 1 - p_x$ be the corresponding probability of failure. Let $\mathcal{I}_x = A_x - \{h_x\}$ be the set of invitees to the meeting x, and $p_{l,y}$ be the probability that a time interval of length l is free on individual y's calendar. Then,

$$p_x = \prod_{k \in \mathcal{I}_x} p_{l_x,k}.$$

In the above equation we make a simplifying assumption that the free slots on the calendar of one invitee is independent of the free slots on the calendar of other invitees. This assumption is violated in practice, and hence we need to run simulations (as presented in Section 6) to demonstrate the viability of the following analysis.

Let $I_{x,1}$ and $I_{x,N}$ represent the random variables corresponding to the number of iterations taken to schedule the meeting x by the **best** and the **good** announcement strategies respectively. The probability density function for the number of iterations required to schedule meeting x this way is given by

$$P_{I_{x,1}}(i) = p_x q_x^{i-1}$$

and the expected number of iterations is given by

$$E[I_{x,1}] = \frac{1}{p_x} = \frac{1}{1-q_x}.$$

This equation assumes that the probabilities of scheduling the meeting in different iterations are mutually independent. This is not true in general. but was used to simplify calculations. Simulations were run to see the effects of this simplifying assumption on the accuracy of the derivations, and results from these simulations are presented in Section 6.

Now, let us consider the **good** announcement strategy where the host of a meeting sends N proposals per iteration. The probability density function for the number of iterations required to schedule a meeting this way is given by

$$P_{I_{x,N}}(i) = (1 - q_x^N)(q_x^N)^{i-1}$$

and the expected number of iterations is given by

$$E[I_{x,N}] = \frac{1}{1 - q_x^N}. \tag{1}$$

This equation indicates that, as N gets larger, the expected number of iterations decreases towards 1 (since $q_x < 1$). That is, as the amount of information about available times that an attendee sends approaches its entire calendar, the host and invitee will only need to engage in a single round of message passing.

The ratio of the expected number of iterations required to schedule a meeting using the **best** announcement strategy to that of the **good** announcement strategy is thus given by

$$\frac{E[I_{x,1}]}{E[I_{x,N}]} = \frac{1 - q_x^N}{1 - q_x} = \sum_{i=0}^{N-1} q_x^i.$$

A closer inspection of this result suggests that the greater the probability of failure (q_x) of scheduling a meeting in any iteration using one information packet, the more the savings in time that we obtain by using **good** announcement strategy over the **best** announcement strategy; this savings can be increased by sending more and more information packets per iteration.

5.1.2 Communication Cost.

Let $p_{x,1}(i) = P_{I_{x,1}}(i)$ be the probability that the meeting x could be scheduled in exactly i iterations using one proposal per iteration. Also assume that if the number of iterations taken to schedule a meeting using the **best** strategy is i, then the number of iterations taken to schedule the same meeting using the **good** strategy will be $\lceil \frac{i}{N} \rceil$. Then the absolute difference between communication costs incurred by the **best** and the **good** (which uses N proposals per iteration) strategies is given by

$$\begin{aligned} C_d &= C_{\mathbf{good}} - C_{\mathbf{best}} \qquad (2) \\ &= (N-1) \cdot p_{x,1}(1) + (N-2) \cdot p_{x,1}(2) + \cdots + 0 \cdot p_{x,1}(N) + \\ &\quad (N-1) \cdot p_{x,1}(N+1) + \cdots + 0 \cdot p_{x,1}(2N) + \cdots \\ &= \sum_{i=1}^{\infty} \{N - 1 - [(i-1) \bmod N]\} \cdot p_{x,1}(i). \end{aligned}$$

This equation implies a periodicity in the communication savings obtained by using the **best** instead of the **good** strategy when scheduling unconstrained meetings. This is because sometimes, when using the **good** strategy, all the proposals sent in the last batch are not needed. If the actual number of iterations while using the **best** strategy is a multiple of N, the number of suggestions sent out by the **good** strategy per iteration, then no communication savings are achieved. However, as N increases, the likelihood increases that the **best** strategy will incur less communication overhead than the **good** strategy.

5.2 Bidding Strategies

Consider the **alternatives** bidding strategy. Let t_{a_i} be an interval proposed by the host for meeting i to the other invitees, and $t_{r_{ij}}$ be the interval returned by invitee j in response to this proposed time. Let $t_{r_i}^{max} = \max(t_{r_{ij}}), \forall j, j \in (A_i - \{h_i\})$, denote the farthest time from the proposed time returned by an invitee. It is assumed that the invitees respond with the closest interval to the proposed interval in which they can meet. So, the earliest interval that the host should propose in the next iteration is $t_{r_i}^{max}$. Now, consider the case of **yes_no** strategy in which the invitees respond with an affirmative/non-affirmative answer to a proposed meeting time. In the same scenario, the host has no idea about the nearest available meeting times of the different invitees, and thus has to step through his own calendar sequentially, announcing in turn every free interval that can be used for the meeting. This would incur a number of extra iterations which could be eliminated from the information gathered if the **alternatives** strategy was being followed. Effectively, the number of iterations saved (I_s) is equal to the number of free intervals of length l_i in the calendar of h_i in the interval $I_i = \left(t_{a_i}(1), t_{r_i}^{max}(1)\right]$, where $t_x(i)$ denotes the ith slot of interval t_x.

Let n_{I_i,l_i} be the number of distinct intervals of length l_i that can be placed on the calendar in the interval I_i. Let I_s be the random variable representing iterations saved. Then, the probability that **alternatives** strategy will save j iterations over **yes_no** strategy is given by

$$P_{I_s}(j) = \left(\begin{array}{c} n_{I_i,l_i} \\ j \end{array} \right) p_{l_i,h_i}^{j} (1 - p_{l_i,h_i})^{n_{I_i,l_i} - j}.$$

This probability follows a binomial distribution, and hence, the expected iteration savings is given by

$$E[I_s] = n_{I_i,l_i} p_{l_i,h_i}. \tag{3}$$

Hence the expected iterations saved increases as the difference increases between the original proposed meeting time and the farthest time returned by an invitee in response to the proposal. Also, the less crowded the host calendar is, the more the expected savings.

Now, we calculate $t_{r_{ij}}$ from t_{a_i}. Let $s_{a_{ij}}$ be the random variable representing the number of intervals considered by the invitee j in response to the proposed time t_{a_i}, before a free interval is found. The probability that x intervals were looked at before a free interval was found is given by:

$$P_{s_{a_{ij}}}(x) = p_{l_i,j}(1 - p_{l_i,j})^{x-1}.$$

This probability distribution closely resembles the geometric distribution and the expected number of intervals looked at is:

$$E[s_{a_{ij}}] = \frac{1}{p_{l_i,j}}.$$

Therefore, we have,

$$t_{r_{ij}} = t_{a_i} + \frac{1}{p_{l_i,j}} - 1.$$

This implies that as the calendar of the invitee gets more full ($p_{l_i,j}$ becomes smaller), the counter-proposal becomes increasingly distant in time from the proposed interval. Given these expressions we can now calculate n_{I_i,l_i} as following:

$$n_{I_i,l_i} \;=\; \max\left(\frac{1}{p_{l_i,j}}\right),\ j \in (A_i - \{h_i\}).$$

The communication cost in this case is directly proportional to the time taken, which is a function of the number of iterations.

5.3 Commitment Strategies

Given the definitions of Section 4.1 for viable, proposed, blocked, and reserved time intervals, we predict that the following kinds of interactions can take place between two processes $M_{x,j}$, $M_{y,j}, x \neq y$ in which an individual j is participating:

possible: $\exists X, Y,\ X \in p_{xj}, (\exists y, Y \in p_{yj}), X \bigcap Y \neq \emptyset$. If overlapping time intervals were proposed for different meetings by the respective processes, there is a possibility that both these meetings could be scheduled for these time intervals in which case one of the processes will fail to schedule its meeting.

actual: $\forall X, \exists Y, X \in v_{xj}(\exists y, Y \in r_{yj}), X \bigcap Y \neq \emptyset$. This scenario corresponds to the case where a request for a meeting m_x comes in such that all viable time intervals corresponding to that meeting overlap with reserved time intervals for some other meetings (m_y). In such a case the processes M_{xj} and M_{yj} are actually competing for overlapping intervals of time and this results in a failure to schedule meeting m_x.

preemptive: $\exists X, Y, Z, X \in v_{xj}, (\exists y, Y \in b_{yj}, X \bigcap Y \neq \emptyset) \bigwedge \neg(\exists z, Z \in r_{zj}, X \bigcap Z \neq \emptyset)$. This scenario corresponds to the case where a request for a meeting m_x comes in such that at least one viable time interval corresponding to that meeting overlaps with a blocked time interval for some other meeting (m_y), but does not overlap with any reserved time intervals. If $\forall X, \exists Y, Z, X \in v_{xj}(\exists y, Y \in b_{yj}, X \bigcap Y \neq \emptyset) \bigwedge \neg(\exists z, Z \in r_{zj}, X \bigcap Z \neq \emptyset)$, it is not possible to schedule m_x, which affects the success ratio of the scheduling strategy. Note that the scheduling process does not wait to see if the blocked time interval is actually used or not, but simply signals a failure to schedule the new meeting. This design decision was incorporated to prevent deadlocks. Of course, the process could time out after some prespecified time period, but then the scheduling process will slow down considerably.

We now develop some predictions of the ratio of the available to the total number of possible intervals of length l that are free on a calendar that contains r reserved hours and (in the case of **committed** commitment strategy) b blocked hours out of $\mathcal{L}$ hours per day. This availability ratio will determine the probability of success of scheduling a constrained meeting (that can be scheduled in exactly one position on the calendar) on that day of the calendar. The expected availability ratio of scheduling under the **committed** commitment strategy is given by

$$AR_{committed} = \frac{\binom{\mathcal{L}-l}{r}\binom{\mathcal{L}-l-r}{b}}{\binom{\mathcal{L}}{r}\binom{\mathcal{L}-r}{b}}. \tag{4}$$

The expected percentage of **actual** interactions (conflicts) is given by

$$\begin{aligned} Conf_{ac} &= \frac{\left(\sum_{i=1}^{min(l,r)}\binom{l}{i}\binom{\mathcal{L}-l}{r-i}\right)\binom{\mathcal{L}-r}{b}}{\binom{\mathcal{L}}{r}\binom{\mathcal{L}-r}{b}} \times 100 \\ &= \frac{\left(\binom{\mathcal{L}}{r}-\binom{\mathcal{L}-l}{r}\right)\binom{\mathcal{L}-r}{b}}{\binom{\mathcal{L}}{r}\binom{\mathcal{L}-r}{b}} \times 100. \end{aligned} \tag{5}$$

The expected percentage of **preemptive** interactions (conflicts) is given by

$$Conf_{pr} = \frac{\sum_{i=1}^{min(l,b)} \binom{l}{i} \binom{\mathcal{L}-l}{b-i} \binom{\mathcal{L}-l-(b-i)}{r}}{\binom{\mathcal{L}}{r} \binom{\mathcal{L}-r}{b}} \times 100. \tag{6}$$

The expected availability ratio under **non-committed** strategy is given by

$$AR_{non-committed} = 1 - \frac{Conf_{ac}}{100}. \tag{7}$$

5.4 Probability of Free Time Intervals

A configuration of meetings on a calendar is given by a set of meetings scheduled in a particular way on the calendar. For a set of n meetings of length $l_i, i \in \{1, \ldots, n\}$, the set $\{(l_1, \langle D, H_1 \rangle), \ldots, (l_n, \langle D, H_n \rangle)\}$ represents a configuration where H_i gives the starting hour of the ith meeting and D is the calendar date. Let I represent a given time interval, and $\tilde{l}$ represent a set of free slots representing one or more possible configurations of meetings on the calendar for the day of the given time interval. Then, the probability that the time interval I is free is:

$$p(I) = \sum_{\tilde{l}} p(I|\tilde{l})\, p(\tilde{l}),$$

where the summation extends over all possible configurations that can be accommodated in a calendar day.

The conditional probability of the interval being free, given that the configuration $\tilde{l}$ is present on the calendar, is given by

$$\begin{aligned} p(I|\tilde{l}) &= 1, \text{if } \forall h \in I, h \in \tilde{l} \\ &= 0, \text{otherwise.} \end{aligned}$$

The probability of a particular configuration, $p(\tilde{l})$, is the number of ways meetings of varying lengths can occupy the slots in the calendar day not in $\tilde{l}$, divided by the number of ways meetings of varying lengths can be placed on the calendar day. We now present two simplifying assumptions in order to reduce the complexity of analysis. The first assumption is that all meetings already on the calendar are of length 1 (longer meetings are simply combinations of one-hour meetings), and the second assumption is that the density of the calendar d (fraction of hours occupied) is known and every day is equally dense. Thus

$$p(\tilde{l}) = \frac{1}{\begin{pmatrix} \mathcal{L} \\ \mathcal{L} - d\mathcal{L} \end{pmatrix}}.$$

With the above simplifying assumptions we now have $p(I) = p_{|I|,j}$, the probability of an interval of certain length being free on individual j's calendar (since the actual position of the interval on the day does not matter any more). We can now directly calculate the required probability $p(I)$ as

$$p(I) = \frac{\begin{pmatrix} \mathcal{L} - l \\ (\mathcal{L} - d\mathcal{L}) - l \end{pmatrix}}{\begin{pmatrix} \mathcal{L} \\ \mathcal{L} - d\mathcal{L} \end{pmatrix}},$$

where $l = |I|$, is the length of the interval I (the length of the meeting to be scheduled).

5.5 Effect of commitment on unconstrained meetings

The direct effect of commitment is to change the density of the calendar. Let us represent by d_n and d_c the densities of the calendar when we are using non-committed and committed commitment strategies respectively. The effects of the change in densities on the communication cost incurred and number of intervals required to schedule meetings can be obtained from the equations in Sections 5.1 & 5.2. In the following we investigate how the probability of successfully scheduling an unconstrained meeting is affected by commitment, when a limit on the communication cost that can be incurred or the number of iterations of information exchange that can be performed is predetermined for a given meeting. Let $\mathcal{I}$ be the maximum number of iterations allowed to schedule a meeting and $\mathcal{P}$ be the maximum number of information packets that can be send by the host to schedule the meeting. Actually, from $\mathcal{P}$ we can calculate $\mathcal{I}$ and vice versa ($\mathcal{I} = \frac{\mathcal{P}}{N}$, where N is the number of intervals proposed per iteration by the host). Given that i is an unconstrained meeting, the probability that it can be scheduled in $\mathcal{I}$ or less iterations is given by

$$P^{\mathcal{I}}_{i,I_N} = \sum_{j=1}^{\mathcal{I}} P_{i,I_N}(j) = p_i \sum_{j=1}^{\mathcal{I}} q_i^{j-1} = p_i \frac{q_i^{\mathcal{I}} - 1}{q_i - 1}.$$

From Section 5.1 we know that p_i depends on the probability that an interval of length equal to the length of the meeting to be scheduled, is free on all the attendees calendar. These probabilities in turn depend on the densities of the calendars of those attendees, and are affected by the choice of the specific commitment strategy to be used. Since $d_c \geq d_n$, the committed strategy, in general, serves to reduce $P^{\mathcal{I}}_{i,I_N}$ and hence leads to a lower availability ratio given the limits on communication cost and time taken for scheduling. This is our first attempt at analyzing the effects of commitment on the success of scheduling unconstrained meetings. Experimental verifications for these results as well as more comprehensive analytical formulation of the effects of commitment on the evaluation metrics are forthcoming.

6 Experimental Verification

To validate our model, we need to verify its predictions experimentally. This is particularly true because we made a number of simplifying assumptions while analytically developing the predictions. In this section, we summarize our preliminary results in verifying some of the predictions given in the previous section. Our experimental testbed is written in Common Lisp and the Common Lisp Object System (CLOS), and models concurrent meeting scheduling activities through discrete event simulation. Because of the complexity of the concurrent meeting scheduling activities and the interactions between the effects different heuristic strategy choices have on the performance measures, we have decided to verify the predictions of each strategy choice in isolation from the others to begin with.

We have verified the expected values for the number of iterations required for the alternative announcement strategies, as predicted in Equation 1. While varying between the **best** and **good** strategies, we hold the other two strategies constant at **yes_no** (for bidding strategy) and **non_committed** (for commitment strategy). A controlling routine takes the density of the calendar and the length of the meeting to be scheduled as input, and generates all possible calendar configurations corresponding to that density. To keep computation tractable, we have initially simulated only two agents, the host starting with an empty calendar and the invitee starting with each of the possible configurations generated. The host is given the job of scheduling an unconstrained meeting of specified length with the invitee. The average over all of the different runs and the expected values of the performance measures as predicted by the formal analysis are then tallied.

Figure 1 illustrates a representative example from our experiments. To avoid unwieldy combinatorics in enumerating all the possible initial states of the invitee's calendar, these experiments assume a very low meeting density: 8 hours of the 9 hour day are free. Each experimental data point represents the average number of iterations from an exhaustive set of runs, where each run corresponds to a different possible configuration of the invitee's calendar. Between data points, we vary the length of the meeting that the host is attempting to schedule, ranging from 1 hour up to 6 hours.

In Figure 1, we have plotted both the expected number of iterations as predicted from our model and the observed average value as determined experimentally. Over much of the range of meeting lengths, the experimental values are slightly larger than the expected values, which can be attributed to the simplifying assumption of assuming mutually exclusive iterations. Actually, the probability of scheduling a meeting in an iteration varies by a small amount (depending on the number of attendees of a meeting, meeting length and the density of the calendars), and this gives rise to the perceived difference between expected and experimental values.

Given that only one hour per day is filled in the invitee's initial calendar, meetings of length 4 hours or less must be schedulable on the first day of the calendar. As the host attempts to schedule meetings longer than 4 hours and approaching the maximum of 8, it becomes more likely that additional days will have to be investigated. In fact, because the **yes_no** bidding strategy provides so little guidance to the host, scheduling meetings of length 7 and 8, which require days whose filled hour is very near either the beginning or the end of the day, becomes too computationally costly when formulating all possible combinations of hours free over a range of days. For experiments in which meetings of 5 or 6 hours are scheduled, we truncated the calendar to only 2 days; if the meeting cannot be scheduled at all in a specific run, the run is not counted toward the average. The effects of restricting the possible calendars is evident in Figure 1, where the trend of exceeding the analytical expectations established by the experimental results for shorter meetings is broken for meetings longer than 4 hours.

Because the total scheduling time is directly proportional to the number of iterations spent scheduling it, Figure 1 provides quantitative results with which to compare our intuitions (Section 2). We had anticipated that strategies involving more information, such as **good**, would converge faster than strategies such as **best**. Our results bear this out, but the speed-up can vary widely, from an insignificant amount when scheduling a 1 hour meeting, to a factor of more than 2 when scheduling a 6 hour meeting.

In Figure 2 we present experimental verification of the expected difference in communication cost for different *announcement strategies*, as developed in Equation 2. The experimental setup is identical to that used previously. We have plotted the number of packets exchanged per meeting using both **good** and **best** strategies for scheduling unconstrained meetings of varying lengths. Also plotted are the expected and experimental values of the difference in communication cost using the two announcement strategies, which correlate closely. We would expect each iteration of announcements and bids to add to the communication overhead, which is confirmed by noting that the communication cost increases as the ratio of meeting length to available hours per day increases towards one (which increases the number of iterations as in Figure 1).

Our intuition that communication costs should be greater for strategies like **good** is evident from Figure 2. Quantitatively, in our experiments, the **good** strategy sent 3 proposals per iteration; only when the number of iterations for this strategy is less than a third of those needed by the **best** strategy will overall communication be saved. And only when the number of iterations required by **best** is greater than the number of proposals (3) in **good** can we hope to see the periodicity in communication cost for alternative announcement strategies predicted in Equation 2. For our experiments, the maximum expected iterations obtained is just about 3 (Figure 1), and thus the predicted periodicity is not seen. We are currently extending our experiments to verify this prediction.

Figure 3 presents experimental results to verify the predictions of Equation 3. We used only unconstrained meetings to verify the predictions and collected data over all cases with some prespecified hours already reserved in the host and the invitee calendars. A controlling routine generates all possible host and invitee calendars (limited to 2 days maximum to maintain tractability) with given densities. These possible calendars are paired up in all possible ways and then the scheduling algorithm is used to schedule meetings of different lengths using both **yes_no** and **alternatives** bidding strategies. *Announcement strategy* is fixed at **best** and *commitment strategy* is fixed at **non-committed**. In Figure 3 we present the case where 1 hour per day is already reserved in the host calendar and 2 hours per day are reserved in the invitee calendar.

From Figure 3, we can confirm our expectation that the number of iterations required to schedule a meeting with **yes_no** bidding strategy increases at a greater rate than for the **alternatives** bidding strategy as the length of the meeting to be scheduled increases. Hence, we save more iterations and time in scheduling longer meetings using the **alternatives** bidding strategy than the **yes_no** bidding strategy. As in Figure 1 and for the same reasons, we find that the difference in iterations used by the two strategies obtained experimentally is a little more than the expected values. Further experiments (not shown here) confirm our prediction that the denser the schedules, the more closely the theoretical and the experimental values agree, because the probability of scheduling a meeting in one iteration changes less over successive iterations.

In Figure 4 we present experimental results to verify theoretical predictions developed in Equations 4-7. We used only constrained meetings to verify predictions and ran experiments with a host calendar in which r hours are reserved for agreed upon meetings and b hours are blocked for only tentatively scheduled meetings. The latter can be thought of as the negotiation density of the scheduling process, which is related to the difference between the rate at which requests for scheduling new meetings arrive and the rate at which the process is able to schedule requested meetings. We are looking at a single individual's calendar, and whether the new meeting is requested by the owner of this calendar or by another individual is immaterial to the analysis. By also assuming that all meetings are constrained for these experiments, we achieve our initial goal of studying strategies in isolation by decoupling the effects of announcement and bidding strategies from the commitment strategy.

A generating routine takes as input the number of reserved and blocked hours and the length of the meeting (l) to be scheduled. It then generates all possible calendar days satisfying these constraints and for each day attempts to place the meeting at each of the $\mathcal{L} - l + 1$ feasible times (recall $\mathcal{L}$ is the number of hours per calendar day). If the new meeting overlaps with any reserved hour, a failure to schedule the meeting is signaled for both commitment strategies. On the other hand, if the new meeting only overlaps with blocked time, then a scheduling failure is signaled only for the **committed** strategy. The percentage of conflicts with reserved hours (**actual** interaction) or only with blocked hours (**preemptive** interaction) is also recorded. These data for calendar days in which 2 hours are reserved and 1 hour is blocked are plotted in Figure 4.

As expected, the availability ratio of the **non-committed** strategy is greater than for the **com-**

mitted strategy. Also, as the ratio of the meeting length to the number of free hours per day increases towards one, the availability ratios decrease considerably because conflicts with reserved hours greatly increase. One interesting observation is that, because **preemptive** interactions (conflicts with blocked hours) take place only when **actual** interactions (conflicts with reserved hours) do not, preemptive interactions actually decrease as meeting lengths increase beyond 3 hours. Because **committed** takes its greatest toll on performance (lower availability ratios decrease the probability of success in scheduling constrained meetings) when the number of conflicts with blocked hours is greatest, **committed** is worst relative to **non-committed** when the percentage of preemptive interactions is greatest (Figure 4). When meetings to be scheduled are much longer or shorter than 3 hours, the availability ratios of the two strategies become closer.

While our preliminary experiments have confirmed some of the predictions from our model, much experimental work remains to be done to confirm the predictions under different meeting densities and with more agents. Due to the computational intensity of the experimental work, our approach has been to verify our model with simplistic studies first; more extensive testing is an ongoing activity.

7 Conclusion

While qualitative predictions about heuristic strategies for controlling meeting scheduling can be based on intuitions, we have argued that a more detailed analysis of the process of DMS can yield well-founded, quantitative, and testable predictions. In this paper, we have outlined our preliminary work toward developing and testing a formal model. Quantitative predictions can be crucial in building a workable system. For example, results like those in Figures 1 and 2 show how cost/performance tradeoffs vary with the types of meetings to be scheduled, as do the results in Figure 4. Rather than holding to static heuristic strategies, a good system should assess the current situation and adopt appropriate strategies. Analyses like those in this paper provide the foundation for making informed decisions about alternative strategies. Our ongoing work involves extending the range of predictions we can make and verify. We have also extended our model and protocol to initiate a second phase of negotiation when all the viable time intervals for a new meeting is found to be occupied by other meetings. This additional negotiation phase allows us to develop a structured mechanism for cancellation and rescheduling, so that higher priority meetings can "bump" lower priority meetings, if it is found useful by the scheduling agents to do as such (taking into account rescheduling costs) (Sen and Durfee 1993).

At the same time, we are also working on expanding the formalism to be applicable in a broader context. From the DAI perspective, bidding protocols like contract-net have emphasized the decomposition and distribution of large tasks in a network, but little work has gone into more formal studies of how tasks that are simultaneously available for distribution lead to conflict and inefficiency. In other words, how many tasks should an agent be allowed to bid on, and what happens if it is awarded too many? Such questions are fundamental to our model of concurrent scheduling processes that interact through messages and shared memory, possibly using alternative commitment strategies.

A second direction that we are pursuing involves enriching the protocol to improve the quality and efficiency of scheduling. Building on our work using hierarchical protocols (Durfee and Montgomery 1991), in which we view coordination as a distributed search process, we are looking at alternative representations for abstracting temporal intervals, to enable agents to negotiate abstractly in early phases to converge with less communication. For example, they could first identify intervals of their calendars where each is relatively free (equivalent to saying, for example, "My schedule is pretty light next week."), and then converging on a meeting by narrowing choices within larger intervals (Sen and Durfee 1995) .

This work represents the first significant step to addressing the complex problem of formally representing and evaluating heuristic strategies for contract-based distributed scheduling. Based on the work presented in this paper, we have recently proposed a design for an adaptive meeting scheduling agent that can choose appropriate scheduling heuristics based on current environmental conditions (Sen and Durfee 1994a). Design considerations presented in the above-mentioned paper and other experimental results obtained in the distributed meeting scheduling domain (Sen and Durfee 1994b) suggests that we need to extend our probabilistic model in two important directions:

- we need to develop reasoning techniques with more abstract probability measures that are easier to obtain than exact probabilities used here,
- we need to incorporate more dynamic information in our protocol, which enables agents to

adapt to changing circumstances.

In summary, the problem of scheduling meetings is a time-consuming, repetitive, and essential part of the daily chores in an organization. The use of DAI techniques can facilitate automating such a process to relieve the members of the organization of this tedious process. In this paper, we have developed a formal model for distributed meeting scheduling, which we used to make predictions about the impact of various heuristic strategies. Our preliminary experiments have verified some of these predictions. Future work will help us to develop a formal theory to address a wider class of problems of interest both to organizational computing systems and distributed AI.

Acknowledgments

This research has been sponsored, in part, by the National Science Foundation under Coordination Theory and Collaboration Technology grant IRI-9015423 and a Research Initiation Award IRI-9410180, and by a grant from Bellcore.

References

Conry, Susan E., Robert A. Meyer, and Victor R. Lesser. (1988). Multistage negotiation in distributed planning. In Alan H. Bond and Les Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 367–384. San Mateo: Morgan Kaufman.

Dent, Lisa, *et al.* (1992). A personal learning apprentice. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 96–103.

Durfee, Edmund H., Victor R. Lesser, and Daniel D. Corkill. (1989). Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering*, 1(1):63–83.

Durfee, Edmund H. and Thomas A. Montgomery. (1991). Coordination as distributed search in a hierarchical behavior space. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1363–1378.

Feldman, M. (1987). Electronic mail and weak ties in organizations. *Office Technology and People*, 3:83–101.

Greif, Irene. (1982). PCAL: A personal calendar. Technical Report TM-213, MIT Laboratory for Computer Science, Cambridge, Mass.

Grudin, Jonathan. (1987). Social evaluation of the user interface: Who does the work and who gets the benefit? In H. Bullinger and B. Shacketl, editors, *Human Computer Interaction – INTERACT87*, pages 805–811. North Holland.

Hewitt, Carl and Jeff Inman. (1991). DAI betwixt and between: From "intelligent agents" to open systems science. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1409–1419.

Kelley, J. F. and A. Chapanis. (1982). How professional persons keep their calendars: Implications for computerization. *Journal of Occupational Psychology*, 55:141–156.

Kincaid, C., P. Dupont, and A. Kaye. (1985). Electronic calendars in the office: An assessment of user needs and current technology. *ACM Transactions on Office Information Systems*, 3(1):89–102.

Maes, Pattie and Robyn Kozierok. (1993). Learning interface agents. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 459–464.

Malone, Thomas W., *et al.* (1987). Intelligent information-sharing systems. *Communications of the ACM*, 30(5):390–402.

Pan, Jeff Y-C and Jay M. Tenenbaum. (1991). An intelligent agent framework for enterprise integration. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1391–1408.

Sen, Sandip. (1993). *Predicting Tradeoffs in Contract-Based Distributed Scheduling.* PhD thesis, University of Michigan.

Sen, Sandip and Edmund H. Durfee. (1993). Using temporal abstractions and cancellations for efficiency in automated meeting scheduling. In *Proc. of the International Conference on Intelligent and Cooperative Information Systems*, pages 163–172.

Sen, Sandip and Edmund H. Durfee. (1994a). On the design of an adaptive meeting scheduler. In *Proc. of the Tenth IEEE Conference on AI Applications*, pages 40–46.

Sen, Sandip and Edmund H. Durfee. (1994b). The role of commitment in cooperative negotiation. *International Journal of Intelligent and Cooperative Information Systems*, 3(1):67–81.

Sen, Sandip and Edmund Durfee. (1995). Unsupervised surrogate agents and search bias change in flexible distributed scheduling. In *First International Conference on Multiagent Systems.*

Smith, Reid G. (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113.

Taub, Eric. (1993). Sharing schedules. *MacUser*, pages 155–162.

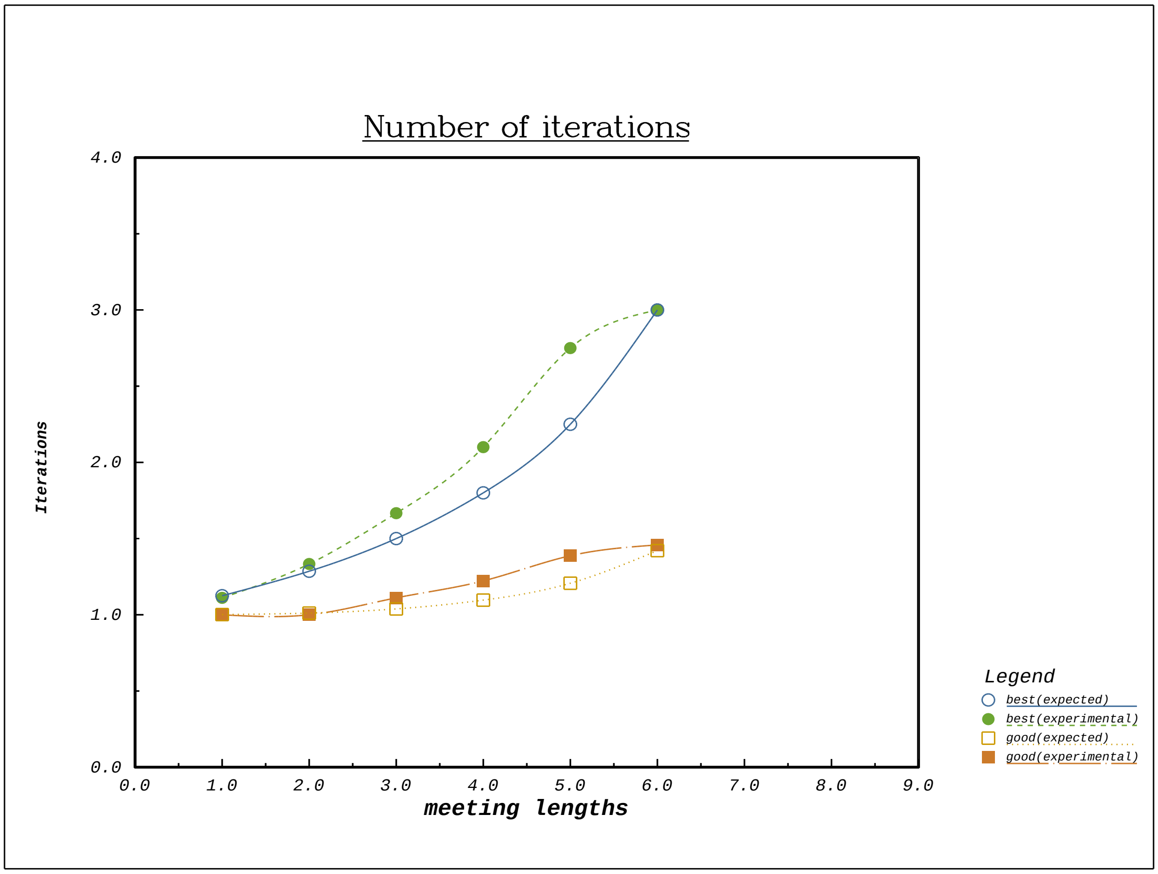


Figure 1: Expected and experimental values of the number of iterations required to schedule an unconstrained meeting using **best** and **good** *announcement strategies* when 8 of the 9 hours in a calendar day is free. *Bidding strategy* is fixed at **yes_no** and *commitment strategy* is fixed at **non-committed**.

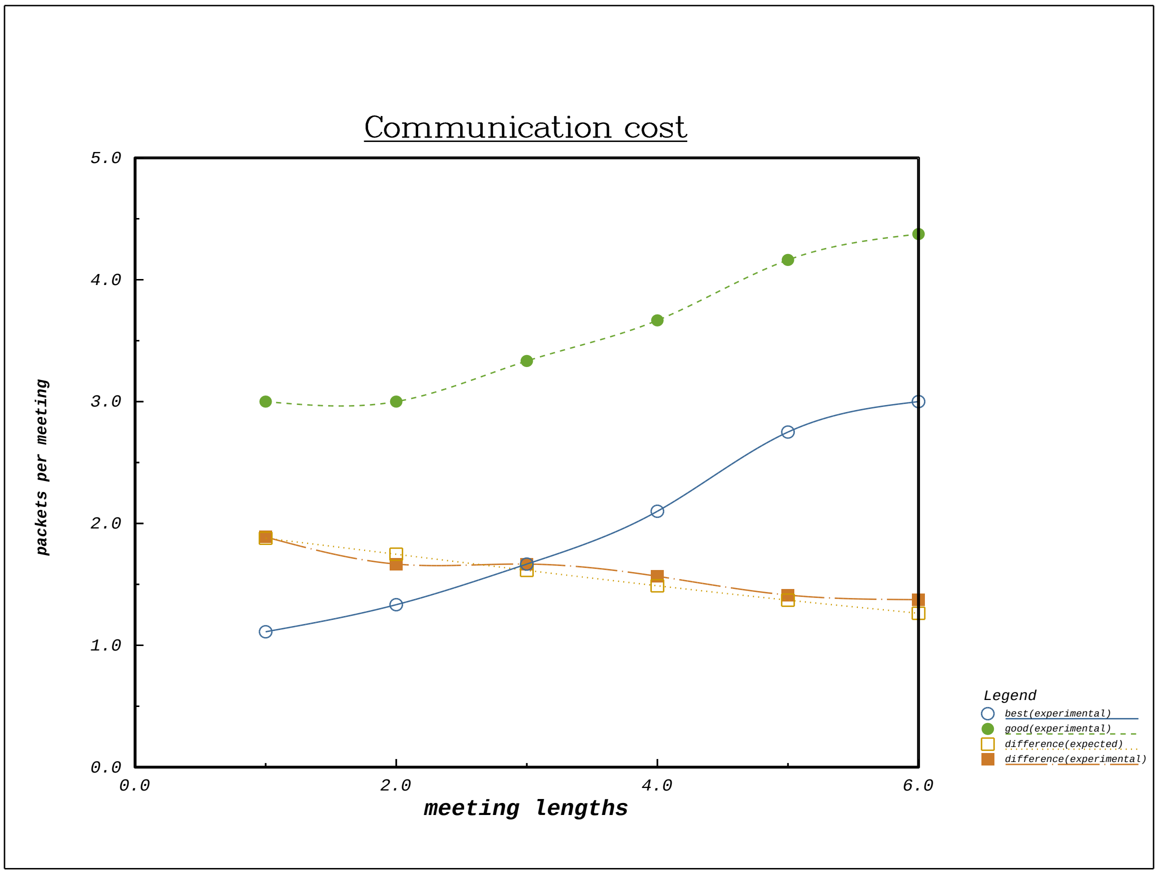


Figure 2: Expected and experimental values of the difference in communication cost incurred in scheduling an unconstrained meeting using **best** and **good** *announcement strategies* when 8 of the 9 hours in a calendar day is free. *Bidding strategy* is fixed at **yes_no** and *commitment strategy* is fixed at **non-committed**. The legends for the different curves are empty circle for best (experimental), filled circle for good (experimental), empty square for difference (expected), and filled square for difference (experimental).

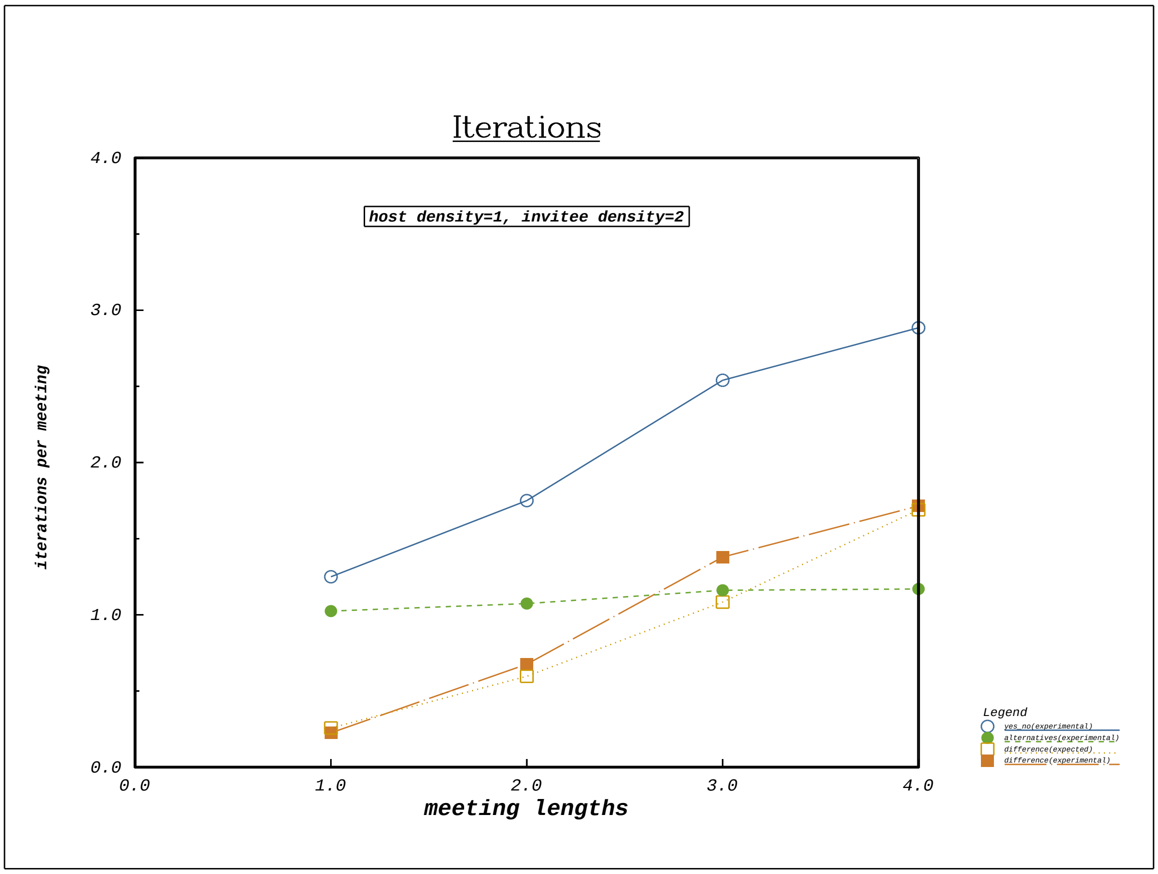
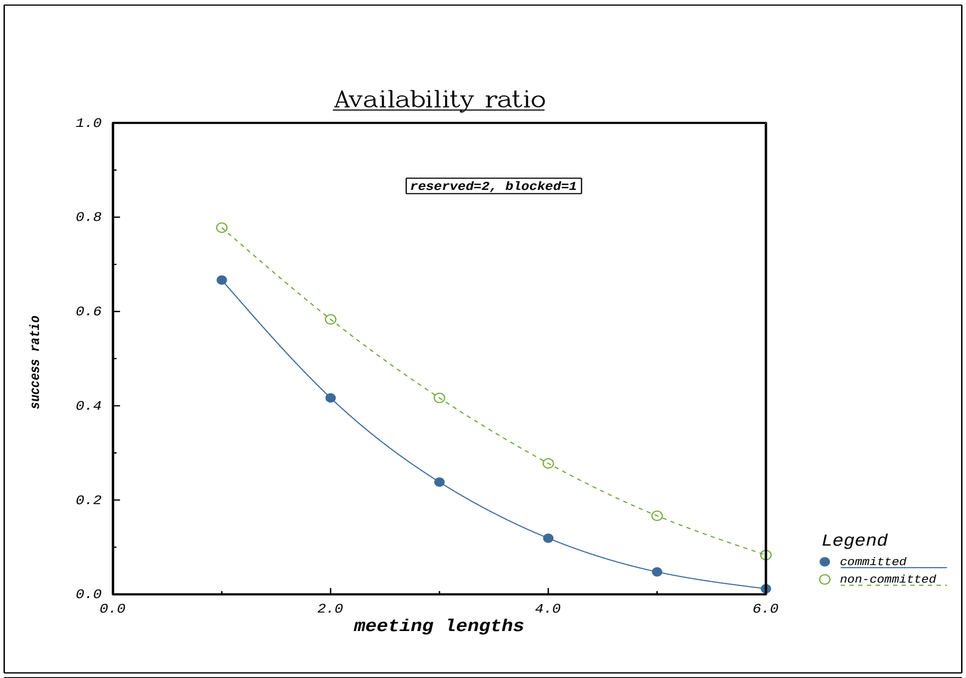


Figure 3: Expected and experimental values of the difference in iterations required to schedule an unconstrained meeting using **yes_no** and **alternatives** *bidding strategies* when 1 out of 9 hours per day is reserved in the host calendar and 2 out of 9 hours per day is reserved in the invitee calendar. *Announcement strategy* is **best** and *commitment strategy* is **non-committed**. The legends for the different curves are empty circle for yes_no (experimental), filled circle for alternatives (experimental), empty square for difference (expected), and filled square for difference (experimental).



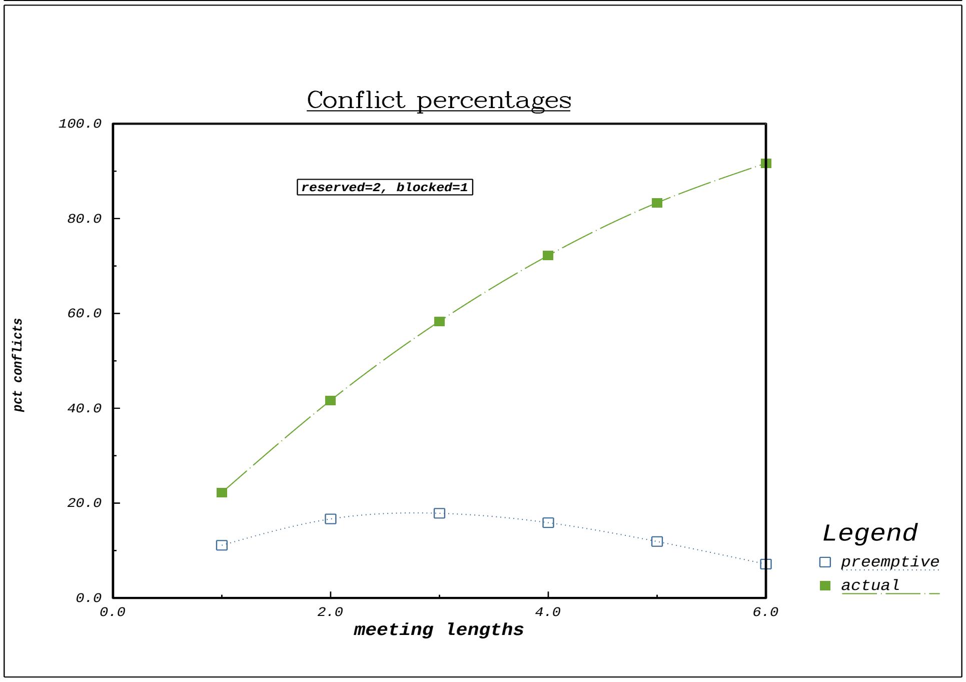


Figure 4: Availability ratios and conflict percentages obtained with different commitment strategies when 2 hours are reserved and 1 hour is blocked out of 9 hours in a day. The expected and the experimental values match exactly for all the measures and hence only one curve per measure is drawn. In the top figure, the availability ratios obtained with the two different commitment strategies is presented. In the bottom figure, the percentage of **actual** (conflict with reserved hours) and **preemptive** (conflict with blocked hours) conflicts out of all scheduling tasks are shown.