Meeting scheduling with preferences with limited comparability: Effects of agent knowledge*

Richard J. Wallace and Eugene C. Freuder
Cork Constraint Computation Center, University College Cork
Cork, Ireland,email: r.wallace-,e.freuder@4c.ucc.ie

Abstract

In scheduling meetings, agents generally have preferences regarding possible times and sites. In addition, they may have privacy concerns along with the desire to solve collective problems efficiently, which necessarily involves a degree of communication. The present work is a study of meeting scheduling by independent agents, where preference scales are not assumed to be identical and where actual preferences are not communicated directly. The purpose was to study the means of reducing the effort (number of communications) required to find a solution that is in some sense optimal with respect to all agent preferences, as well as the relations between efficiency, solution quality (based on various measures) and privacy loss. Agents propose meeting times and places consistent with their schedules while responding to other proposals by accepting or rejecting them. Agents also store information about other agents that is gleaned from the communications, together with general assumptions about travel constraints and possible meeting sites. This provides a way of improving efficiency but also exacerbates the problem of privacy loss, since agents can deduce personal information regarding meetings as well as relative preferences from limited forms of communication. We develop strategies for finding optimal solutions, given minimal assumptions about the comparability of agent preferences, and we show that efficiency can be improved either by making deductions based on communications received or by revealing a limited amount of information about one's own schedule in communications. We find that in some respects the problem of privacy loss can be finessed by this additional information.

Introduction

As a result of the growth of the Internet and World Wide Web, it has become possible to automate a number of cooperative functions, even among widely dispersed participants. One area of application receiving considerable attention is meeting scheduling (Garrido & Sycara 1996) (Luo, Leung, & Lee 2000) (Sen, Haynes, & Arora 1997). This is a task which might be profitably delegated to software agents communicating over networks. In many cases it is expected that the agents will exhibit a degree of independence, since each is working for an individual with distinct affiliations. Thus, rather than solving parts of a single problem, as in the distributed CSP paradigm (Silaghi, Sam-Haroud, & Faltings 2001), each agent has its own problem, but the solutions must all be mutually consistent.

In cooperative communication involving independent agents there are at least three issues that arise: problem-solving efficiency, privacy of agent information, and agent preferences. With respect to privacy, there will be cases where individuals are interested in restricting the information communicated to other individuals to prevent sensitive information from being received by others (Garrido & Sycara 1996). At the same time, the necessary cooperation involves some minimum of information exchange. However, maintaining privacy in such contexts may make cooperative decision making less efficient. When there are independent preferences for solutions, problems usually become harder, since preferences add a new level of constraints on acceptable solutions. This may also involve finding an optimal rather than just a feasible solution.

In previous work, we have considered relations between privacy and efficiency in this context (Freuder, Minca, & Wallace 2001) (Wallace, Freuder, & Minca 2002). In that work, a major emphasis was on the degree to which agents could deduce private information from restricted communications together with general knowledge of the situation. In this work we consider the addition of preferences for meetings. As before, we work with a simplified situation (preferences are assigned at random) with the expectation that the methods developed here will also work in more realistic settings.

This work builds on a recent proposal for representing preferences in a way that can be handled by techniques derived from the study of soft constraints (Franzin *et al.* 2002). In this proposal, agents have a common scale of preference, so that meeting preferences can be meaningfully communicated; in this way it is possible to find a provably optimal solution based on a reasonable criterion for optimality (the maximum minimum preference across all agents). Earlier work has also assumed a common preference scale, that allowed joint utilities to be calculated (Garrido & Sycara 1996) (Luo, Leung, & Lee 2000) (Sen, Haynes, & Arora 1997).

However, the assumption of a common scale of prefer-

*The second author is supported by a Principal Investigator Award from Science Foundation Ireland.

ence may not be realistic in many situations. The question arises, therefore, whether agents can find solutions of high quality using strategies that do not require a common scale. In the present work we examine communication and solution strategies that do not depend on the this assumption. In addition, there is either no direct communication of preferences, or preferences are communicated in a simple, qualitative manner ("I prefer not" versus "I cannot"). In this situation, we also carry forward the goal of previous work: to examine tradeoffs and other relations between efficiency and privacy when preferences also enter into the picture. In this context, we would like to find methods to enhance efficiency while keeping privacy loss to a minimum.

In Section 2 we describe the basic problem for our agents. Section 3 discusses procedures for finding solutions that are optimal under reasonable criteria. Section 4 discusses knowledge that each agent can acquire about other agents in this situation. Section 5 describes a testbed and design of experiments that test effects of different levels of communication and forms of knowledge (modal and actual) on efficiency and privacy loss. Section 6 describes the basic experimental results. Section 7 gives conclusions.

A Meeting Scheduling Problem

In the scheduling problem we are considering, each of k agents has its own calendar, which consists of appointments in different cities at different times of the week. The problem is to find a meeting time that meets the following requirements:

1. all agents can attend given their existing schedules and constraints on travel time
2. the meeting is sufficiently preferred by each agent

To find such a meeting time, agents communicate on a 1:1 basis; the basic protocol is for one agent to suggest a meeting time in a certain city to each of the other agents, who then tell the first agent whether the choice is acceptable or not, given their existing schedules. This continues until a time and place are found that is acceptable to all agents. We assume that during this time the agents' own schedules do not change.

Basic problem

For purposes of analysis and experimentation, we devised a simplified form of this problem. First, we assume a fixed set of cities where meetings can be held: London, Paris, Rome, Moscow and Tbilisi. We also restrict meeting times to be an hour in length and to start on the hour between 9 AM to 6 PM, inclusive, on any day of the week. These restrictions apply both to the pre-existing schedules as well as the new meeting assignment.

The basic constraints are the times (in hours) required for travel between meetings in different cities. These are indicated in Figure 1. Times between cities within one region (Western Europe or the former Soviet Union) are shown beside arcs connecting cities; the arc between the two ellipses represents constraints between any city in one region and the cities in the other.

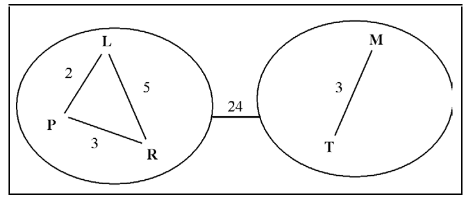

Figure 1: Time constraint graph. Cities are London, Paris, Rome, Moscow and Tbilisi. Further details in text.

Addition of preferences

We assume that for each open meeting an agent has a particular level of preference. In the present situation, we do not assume that agents have identical preference scales, but only that each agent's scale is ordinal. This means that agents can compare preferences for their own meetings and can also establish "preference thresholds" such that a candidate meeting can be compared with the threshold to determine if it meets the minimal level of preference that the agent will accept. We also assume a finite number of distinguishable preference values for each agent.

In the present form of the problem, each meeting is assigned a preference at random. On the face of it, this is highly unrealistic. For example, in this case a meeting in Rome on Tuesday might have a preference value of 7 at 1 PM, 2 at 2 PM and 8 at 3 PM. Or a meeting in Paris at 1 PM might have a preference of 8 for an agent that has a meeting in Moscow at 12 PM on the same day. Nonetheless, this approach was chosen because, (i) it is not yet clear how to generate preference profiles in a sufficiently systematic manner. This is because there are many features of the situation that could serve as the basis for preferences, and one would expect individual differences in the relative importance of these features, (ii) the methods devised to find meetings with good preference values in the random case will also work when preferences are more systematically related to the features of the problem, (iii) under the present conditions, or assumptions, the existence of profiles and the like are less likely to alter the results appreciably, since the extra degree of organization will not be directly reflected in the communications.

In this work, we also consider different levels of communication. The first is the simplest one, described at the beginning of this section. The second one elaborates on the first by using the communication "not preferred" rather than "reject", when a proposal is consistent with the agent's constraints but does not reach its current preference threshold. A third level adds to the second by including reasons for rejection based on constraint violations, i.e. an existing meeting that conflicts with the current proposal.

Procedures for Finding Optimal Meeting-Preferences

Within the context of our assumptions about preferences, we consider two criteria for optimization:

- maximize the minimum preference value of any agent

- find a member of the Pareto optimal set of preference-vectors, i.e. one such that there is no other meeting for which one agent has a higher preference value while the others have preferences that are no worse than their present values

The first criterion requires some explication, since it would seem to require a common preference scale. Here, we create a pseudo-common-scale with the desired properties in the following way. If all agents have a finite number of ordered preference values in their individual scales, we can consider these values in terms of the number of successive steps required to reach them, where each step is from some value to the next higher or lower. A maxi-min strategy then involves maximizing the minimum number of steps, i.e. the minimum number of distinct values between the value selected and the lowest value.

For the first criterion, we have tested two simple strategies, which we call "step-up" and "step-down". These will be described in terms of "preference thresholds", in which for any agent, preferences must be at or above the current preference threshold value for a meeting to be accepted (along with it also meeting the constraints of the problem). For step-up, the agents begin by setting their thresholds to their lowest preference values. Then, proposals are made until a meeting is found consistent with the threshold value for all agents. Then the threshold is increased to the next lowest value for each agent, and more proposals are made, etc. The procedure continues until some step k, where no proposals can be found that are acceptable for all agents. Then the last acceptable proposal can be selected as an optimal solution. Step-down is similar, but in this case agents begin with thresholds set to their highest preference values, decreasing the threshold at each step until a proposal is made that is acceptable to all, which, again, must be an optimal solution.

For the Pareto optimal criterion, agents begin by setting their thresholds to the lowest possible value, say 1. They then search for an acceptable solution, essentially without regard to preference values. When this is found, each agent proposes a solution that is at least minimally better than the present one, and each other agent accepts it only if it is at least as good as the present choice. This continues until there are no further proposals to consider.

Each of these criteria has its limitations. It is possible that better solutions could be found - in terms of average number of steps - for the same maxi-min criterion. (Here, it may be possible to elaborate the step procedure to obtain a maxi-min solution that is also better according to an averaging criterion.) For the Pareto strategy, a solution can be found that meets the criterion but which is not even close to maxi-min, for example a solution that is associated with the top value for one agent and the lowest value for the other agents.

Another issue is the possibility of different numbers of scale values. Fortunately, this does not prevent the step-up procedure from finding maxi-min solutions. For example, suppose there are two agents with five and 10 distinct values, respectively, and there is a solution associated with the fifth and eighth values for the respective agents. Then both agents will step up to their fifth values, after which (since no failure has occurred) the second agent will continue stepping up until the ninth step, when search fails. In contrast, if the step-down strategy is used, the first success will occur on step 3, when the second agent has reached its eighth value and the first agent its third. Despite this potentially serious limitation of step-down, we decided to test both strategies when there were equal numbers of scale values, in order to compare their performance.

Agent Knowledge

Agent Views

In the situation we are considering there are three basic messages: a proposal, consisting of a city, day and hour ("Paris on Monday at 3 PM"), an acceptance ("OK."), and a rejection ("Sorry, I cannot."). Agents may also indicate their preference for a particular proposal, in a limited, qualitative fashion ("Sorry, I prefer not to."). In addition, under some conditions agents give reasons for a rejection by communicating one of their conflicts ("I have a meeting in Rome on Monday at 5 PM."). (In this work, under any specified condition all agents communicate the same kinds of information.)

In this situation, agents do not know each other's schedules or their preferences. However, since a solution to each agent's problem depends on $k - 1$ other problems, agents can improve the overall efficiency of problem-solving by collecting information about other agents in the course of search. This, together with general information common to all agents (here, temporal and geographical features) can allow agents to avoid proposals that will not be accepted and to select those that are more likely to be accepted. Such effects are possibly more relevant when preferences are involved, due to the greater difficulty in satisfying criteria for meeting preferences as well as constraints.

To this end, agents maintain "views" of other agents' schedules. A view can be updated after each communication from another agent. Part of a view is *factual* information that has been collected or deduced, in this case actual meetings or open slots. It could also, of course, include actual preferences if these were communicated, although this was not done in the present tests.

In addition, agents can store information concerning *possibilities* (e.g. agent k might be able to meet in Moscow on Friday at 3 PM, or, agent k might have a meeting in Rome on Tuesday at 9 AM). This can be derived from acceptances and rejections, as well as more explicit communications. For example, if agent k rejects a proposal for a meeting in Moscow on Friday at 3 PM, this may (given certain protocol conventions or levels of communication) eliminate this possibility from the proposing agent's view of agent k.

As shown in earlier work (Wallace, Freuder, & Minca 2002), the latter type of information can be represented in the form of constraint satisfaction problems whose domain values are "modal values" (Figure 2). Taking time-slots as variables, there are possible values both for meetings that another agent may have in its schedule and for meetings that it may be able to attend, which we term "possible-has-meeting" and "possible-can-meet" values, respectively. Considered more generally, the former represent possible

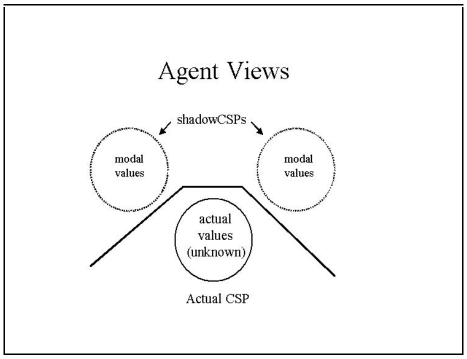

Figure 2: Part of one agent's view of another agent, consisting of two modal-valued CSPs, one representing possible meetings that the latter can attend, the other possible meetings that the latter has already.

existing assignments in an actual, unknown CSP, while the latter represent possible future assignments in the same CSP. This means that, at the beginning of a session when an agent knows nothing about other agents' schedules, there are two such *modal-valued* CSPs one whose domains each consist of five possible-has-meeting values and one in which each domain has five possible-can-meet values (in each case, one for each city). We have called these modal-valued CSPs "shadow CSPs", to indicate the close semantic relation between them and the actual CSP of an agent. Note that we can handle facts such as value x is both a possible existing and a possible future assignment in a natural way with this formalism. In practice, it is more useful to store knowledge of nogoods than positive modal values, in order to rule out possibilities.

In this shadow CSP system, we reason from communications to restrictions in the domains of the actual (originally unknown) CSP, to restrictions of corresponding values in the relevant shadow CSPs. For example, if an agent proposes a meeting in London on Thursday at noon, the other agents can deduce that it has no meeting at that time, thus deleting all five possible-has-meeting values for that variable in the view of that agent. (The reasoning here uses closed-world assumptions with respect to actual CSP domains, so that $\neg x$ entails $\Box\neg x$ which is equivalent to $\neg\Diamond x$.) Or, if an agent rejects a proposal, the proposer can deduce that this is not a valid future assignment, so that one possible-can-meet value can be deleted from the shadow CSP in that view. (Note that, here, nothing can be deduced regarding the corresponding possible-has-meeting value.)

These ideas can be extended to what can be called "shadow preferences", given the assumption of ordinal scales. These can be represented in terms of ranges, starting with a complete range, based either on an agent's own set of possible preference values or the number of steps through the pseudo-scale if a step strategy is used. Then, under the given communication and protocol conventions, these ranges can be restricted. For example, in the step-up procedure, acceptance indicates that the preference for this meeting is at least as good as the present step, while a rejection indicates that the preference value is below the present threshold. Note that if, for step-up, a pattern of acceptance and then rejection is detected for a given proposal, then that meeting can be assigned a single shadow value, and a similar deduction can be made under the complementary situation for step-down. In contrast, the Pareto strategy does not allow such deductions.

Some of the present protocols place fairly tight restrictions on the capacity of agents to deduce possibilistic information from rejections. Consider first the case of minimal communication: proposal, acceptance or rejection. For the step-up and Pareto procedures, rejections due to constraint violations can only be distinguished from those due to preference violations at the first step of the procedure, before the first acceptable solution has been found. For step-down, even this is not possible. In contrast, with additional prefer-not communications, agents can continue to gather and deduce information based on rejections. In all cases, proposals and acceptances imply open slots and in each instance of such communications five nogood possible-has-meeting values can be added to an agent's view of another agent. In addition, with the step procedures it is possible to refine knowledge of preferences throughout a session.

Using agent knowledge

Agents can use the stored knowledge in their views to avoid making proposals that will not be accepted or to temporarily avoid those that are deemed less likely to be approved, as shown in (Wallace, Freuder, & Minca 2002). This is obvious in the case of known meetings (agent k knows that agent r has a meeting in Rome on Tuesday at 2 PM), but it can also be deduced from possibilistic information, either directly (e.g. if all possible can-meet's for agent k have been ruled out for a certain time slot) or indirectly (e.g., if all has-meeting's have been ruled out for a time slot due to conflicts, then the agent cannot meet at that time slot even though it is open). A particularly powerful form of inference is based on possible causes for a rejection; since these must be a subset of the set of possible-has-meeting values, reducing the latter also reduces the former and this may allow agents to make inferences about meetings in another agent's schedule with enough precision to avoid unacceptable proposals.

Preference information can also be used to improve efficiency. If an agent knows that a possible meeting in its schedule is associated with a range of shadow preference for another agent whose upper bound is below the present threshold, then it can avoid proposing that meeting.

Experimental Methods

An experimental testbed

To get some idea of how these different procedures and communication protocols would work together in practice, we used a demonstration system written in Java (Wallace, Freuder, & Minca 2002), to which we added features related to preferences that were described in previous sections. This system allows the selection of the number of agents and initial meetings. Test runs are carried out either interactively or in batch mode.

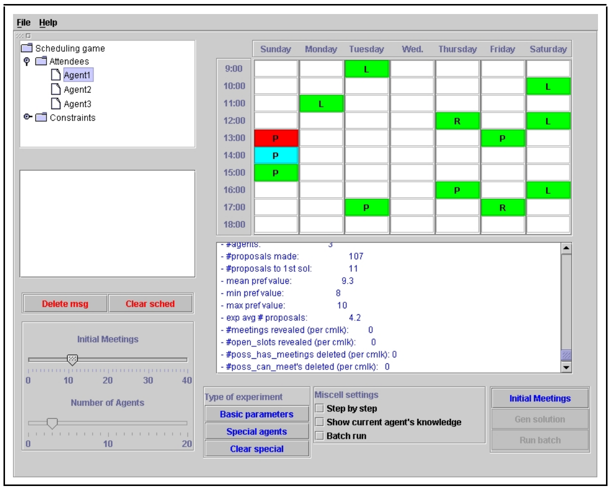

Figure 3: Multi-agent meeting sheduling system.

The main window of the system is shown in Figure 3, including the results of a single run using baseline parameter settings (described below). In this instance there are three agents, each with 11 initial meetings. The schedule in the upper right-hand panel is for agent 1. This agent's meetings are highlighted on a calendar. The darkened slot in the schedule is a meeting chosen as the "guaranteed solution" for this experiment, here Paris on Sunday at 1 PM. The actual solution found in the run just completed is Paris on Sunday at 2 PM. Some of the data for this run can be seen in the panel on the lower right.

This system was built to be used for experiments. Hence, almost all features are optional, so that various combinations of features can be selected for testing. This is especially important for evaluating tradeoffs among the three main factors of efficiency, solution quality (in terms of preferences) and privacy loss. In addition, it allows us to examine these three factors in relation to variables such as information exchanged, knowledge use, proposal strategies, and communication protocols.

Features are chosen via a dialog box evoked by the button labeled "Basic parameters" in the main window. The user can choose the level of communication, kinds of knowledge (actual and possibilistic) to be gathered, whether to use arc consistency, the proposal strategy, and the preference strategy. These settings are independent, so, for example, knowledge gathering can go on even under the baseline proposal strategy, where knowledge is not actually used.

Design of experiments

The experiments reported here involve three agents, where the number of initial meetings varies from 5 to 40 in steps of 5. For these experiments every agent was given a preference scale of 10 values. This allowed the assessment of solution quality that was more comprehensive and comprehensible than it would have been if agents had different numbers of values. It also allowed us to evaluate both step strategies, as discussed earlier.

At the start of each run, schedules are generated at random for each agent, with number of meetings determined by the slider setting (see Figure 3). These schedules are built so as to be consistent with a guaranteed solution that is also selected at random, as well as the basic constraints of the problem. (That is, each agent's schedule is internally consistent.) At this time, all meetings are also given a random preference value (an integer from 1 to 10).

An individual test run consists of a series of proposals made until one is found that is acceptable to all agents. At each (mini)step of the experiment, an agent selects a candidate proposal (a day, hour and city) at random that is a solution for its own schedule; it then checks that this solution has an acceptable preference value, given the current stage of processing. (If it does not, further solutions are selected at random from the remaining set until an acceptable one is found or there are no further choices.) Then, depending on the parameter settings, it may check this proposal against the knowledge it has gathered. The first proposal that satisfies the agent's own criteria is then communicated to each of the other agents, and the latter replies with an acceptance or rejection to the proposer (alone). At this time agents may also update their views, depending on the parameter settings. If the proposal is not acceptable to all of the other agents, another agent is selected and the cycle is repeated. The present experiments used a round robin protocol in which all agents take turns proposing a meeting according to a fixed order.

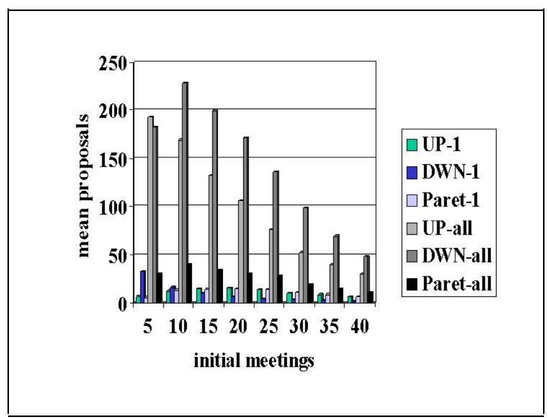

Figure 4: Efficiency measure (proposals per run) in 'baseline' experiments, where no explicit information is communicated about an agent's meetings and no knowledge is used. "-1" is number of proposals before the first solution was found, for the three strategies. "-all" is number of proposals before an optimal solution was found.

In these experiments, the efficiency measure was number of proposals, averaged over all (500) runs in an experiment. The measures of solution quality were the minimum and the average preference values. (Averaging could be used here because in this case the scales did, in fact, have the same values.) The measures of privacy lost were number of meetings identified, number of open slots identified, number of possible-values removed for can-meet and has-meeting, and the number of possible preference values discarded. These privacy tallies were averaged per "communication link" per run, to give an average per agent view. (There are two links for each pair of agents, or $n \times (n-1)$) links for n agents.) In addition, average number of solutions per agent was determined as well as the number of common solutions.

Efficiency and Privacy: Empirical Results

Baseline performance

Figure 4 shows the efficiency measures for the baseline condition, in which agents used one of the three procedures, step-up, step-down or the Pareto optimal procedure to find solutions of optimal quality under the present assumptions. Under this condition, no information on meetings or meeting preferences was gathered during a test run. Both the number of proposals to find the first acceptable solution and the total number of proposals required to establish that an optimal solution has been found are shown. As might be expected, the step procedures are not very efficient in comparison to the Pareto optimal procedure. For comparison, the number of proposals required to find a maxi-min solution when explicit preference information is exchanged under the assumption of a common scale of preference, and using a procedure similar to step-up, is about 75, 65, 52, 45, 30, 20, 15 and 15 for 5, 10, 15, .., 40 initial meetings, respectively (data from (Franzin *et al.* 2002)).

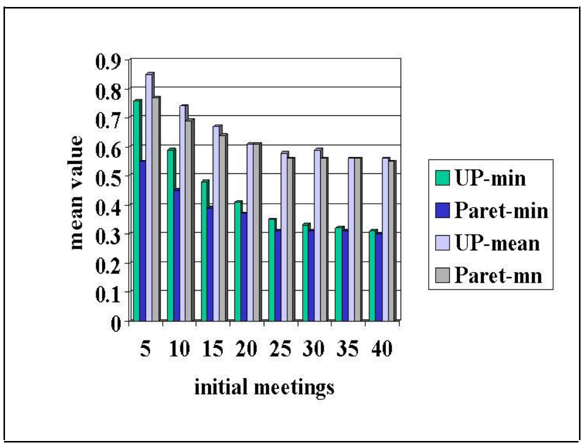

Figure 5: Solution quality measures for baseline experiments. Minimum preference and mean preference values per run for optimal solutions using maxi-min (with step-up) and Pareto optimal strategies. Scales have been transformed to [0..1] for purposes of presentation.

Figure 5 shows two quality measures, the maxi-min score and the mean preference value for the step-up and Pareto optimal procedures. (Step-down, of course, gives the same values as step-up.) In the cases where there are fewer initial meetings per agent, there is an appreciable fall-off in quality for the Pareto strategy. The lack of difference with more meetings is not as consequential, since it is due to the fact that the number of common solutions approaches 1 as the number of meetings per agent increases. Hence, we have a clear efficiency/quality tradeoff.

Effect of knowledge gathering and enhanced communication

In earlier work (Wallace, Freuder, & Minca 2002) we have shown that in this situation efficiency can be significantly enhanced by the use of knowledge gathered during the course of trying to find a common solution. This effect depends critically on possibilistic knowledge, either the deduction of nogood possible-can-meet's or the reduction of sets of possible causes for a former rejection to a small set of (possible) conflicts. In contrast, merely communicating meetings in conflict is not sufficient to guide the agents to a common solution more quickly. This means that efficiency measures and tradeoffs involving efficiency can be positively effected by using these kinds of knowledge. Therefore, it may also be possible in the present situation to improve the efficiency of the step strategies, while retaining the better quality solutions that are obtained with these methods.

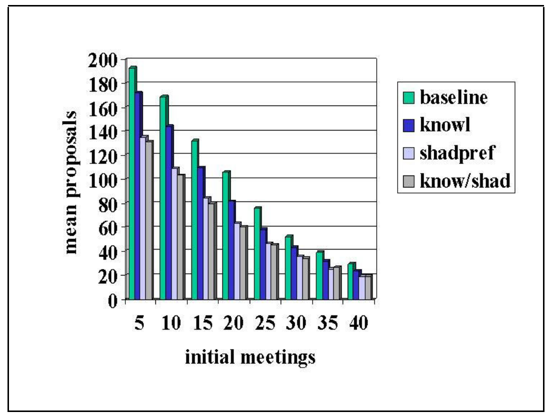

Figure 6: Reduction in mean number of proposals required to find an optimal solution when knowledge is used. Step-Up procedure. Baseline condition is shown for comparison. In the "knowl" condition, knowledge was gathered concerning actual meetings and open slots as well as nogoods for possible-can-meet's and -has-meeting's. In the "shadpref" condition, shadow preference scales were maintained and tighter bounds were deduced from communications. In the "know/shad" condition, all of these sources of information were used together.

Unfortunately, nogoods of the former type can only be derived in quantity when some knowledge of actual meetings is obtained, either by direct communication or by deduction via possible causes (if a set of possible causes is reduced to one, then the agent must have this meeting). This means that there is a tradeoff between efficiency and privacy loss that will be entailed by knowledge gathering. The purpose of the tests described in the rest of this section was to examine this tradeoff, as well as to determine the degree to which efficiency could be improved by these methods.

Figure 6 shows the effect of information gathering on efficiency under the basic communication protocol involving minimum information exchange. This is for the step-up procedure; these techniques were not effective with the step-

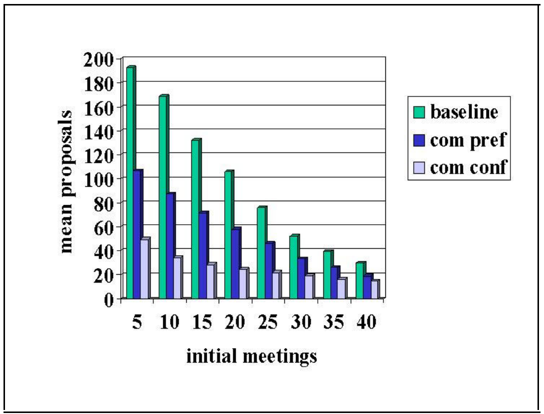

Figure 7: Reduction in mean number of proposals required to find an optimal solution when level of communication is altered. Step-Up procedure; baseline condition is shown for comparison. All other conditions involve full knowledge use (as in the "know/shad" condition in the previous figure). In the "com pref" condition, agents indicated whether their rejection was due to constraint or preference violations (as described in Sect. 4.1). In the "com conf" condition agents *also* gave one reason for a rejection based on constraint violations in the form of one meeting that was in conflict with the proposal.

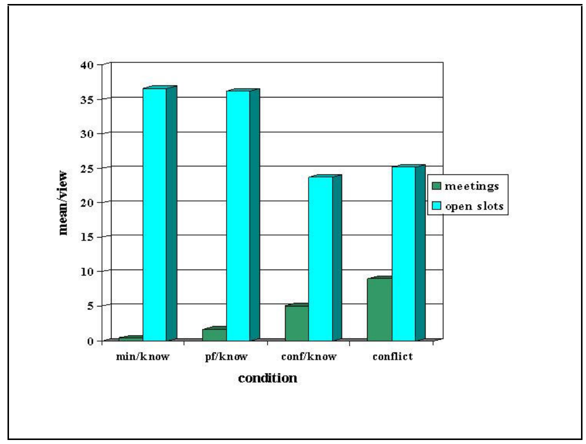

Figure 8: Privacy measures for tests with knowledge and/or amplified communication. Number of actual meetings and open slots identified per run per agent view. Data for conditions with 15 initial meetings per agent. "min/know" = minimal communication with knowledge (actual, modal and preference), "pf/know" = preference violations distinguished, "conf/know" = one conflict revealed with each rejection, "conflict" = conflicts revealed when modal and preference knowledge are not gathered.

down procedure (basically because previous rejections could not be used to bound shadow preferences in a way that allowed agents to avoid making proposals again), while with the Pareto optimal procedure the bounding procedures used with the step procedures are not available.

In the case of minimum communication, bounding preferences via the shadow preference representation is more effective than actual and possibilistic knowledge that is deduced about meetings. This is undoubtedly because information can be deduced from rejections only until the first solution is found, since after this rejections due to constraint violations cannot be distinguished from those due to preference violations. If both are used together, the improvement is slightly greater than with preference information alone.

Figure 7 shows the effect of including more information in the communications. There are two important findings. First, if preference violations are communicated separately from constraint violations, then the use of knowledge involving possibilities is more effective. Under this condition, the effect of preference information alone (not shown in the figure) is no different from the effect under minimal communication. This shows that the improvement is due to the use of rejections based on constraint violations, which can now be detected throughout the run.

Second, when agents communicate meetings in conflict there is a further reduction in the number of proposals required to find an optimal solution. This effect is due to inferences involving modal information, since without such knowledge use, communication of conflicts gives results for efficiency that are identical to the baseline condition. By using modal knowledge together with arc consistency processing, agents are able to deduce a large number of nogoods for possible-can-meet values, and this in turn allows them to rule out candidate proposals. In all the above conditions, runtimes were about the same with and without the extra overhead due to information storage.

Figures 8 and 9 show the effect of communicating more information on privacy loss. These are for runs where agents had 15 meetings in their schedules; very similar results were found for other numbers of initial meetings. There are several points to note:

- The small but consistent increase in actual meetings identified when non-preferences are communicated separately from constraint violations (Figure 8) is due to deductions via possible causes for rejections (cf. (Wallace, Freuder, & Minca 2002)).
- The number of meetings identified when conflicts are cited by agents is greater when possibilistic knowledge is *not* gathered (Figure 8), since the number of proposals is much greater. This reflects the inability of agents to use this information directly to guide proposal selection effectively, as already noted.
- Fewer open slots are deduced when conflicts are communicated (Figure 8). When possibilistic knowledge is used, this is due to improved efficiency. In this case, therefore, the tradeoff between improved efficiency and privacy loss is avoided.
- The same effect was observed for preference information (Figure 9).
- Communicating conflicts results in a marked increase in nogoods deduced for possible can-meet's (Figure 9). As

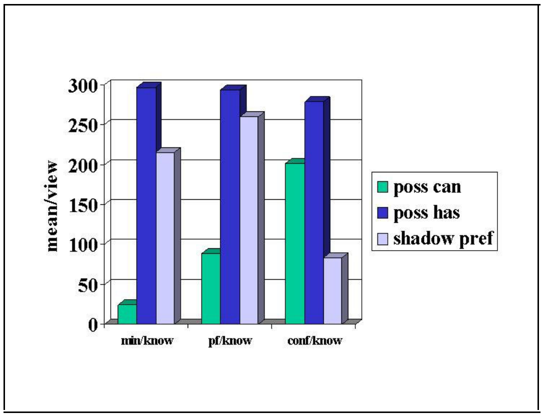

Figure 9: Privacy measures for tests with knowledge and/or amplified communication. Modal nogoods and shadow preference values gathered per run per agent view. Data for conditions with 15 initial meetings per agent. Labels on x-axis as in Figure 8.

discussed above, this is the major mechanism by which efficiency is improved.

These results show that a considerable amount of information can be gained about another agent during the course of solving a problem of common interest. (For reference, if an agent has 15 meetings, then it has 55 open slots, and there are 350 distinct modal values of each type, and 350 preferences, which under the assumption of 10 distinct preference values, gives a total of 3500 possible values.) For some kinds of information, this occurs even when explicit communication is kept to a minimum (cf. "min/know" conditions in Figures 8 and 9). In fact, for information about open slots and preferences, privacy can actually be reduced by revealing other kinds of private information, here actual meetings. This means that there is no straightforward tradeoff between efficiency gain and loss of privacy, at least when agents are able to make sophisticated deductions about another agent's schedule.

Conclusions

This work shows that 'optimal' solutions can be found, according to reasonable criteria, under limited assumptions regarding the commensurability of preference scales among independent agents. This can also be done fairly well with only limited forms of communication - provided agents are given powerful deductive machinery based on specific assumptions regarding constraints and possible meeting sites. With more explicit communication - either by distinguishing preference from constraint violations or by communicating a portion of the actual meetings in conflict with a proposal - the effort required to find an optimal solution can be reduced by a factor of 3-5 with respect to baseline conditions. This means that methods (in particular, step-up) that give good-quality solutions reliably can be made almost as efficient as simpler methods that do not carry the same guarantees of quality (Pareto strategy).

As would be expected, tradeoffs appear between efficiency and privacy loss, but these are complicated by the manner in which information is handled. In fact, by making deductions based on modal information, it is possible to finesse the tradeoffs in some respects. This was found for information regarding preferences as well as open slots, when direct information about meetings was communicated. In contrast to previous work where only a feasible solution was desired (Wallace, Freuder, & Minca 2002), it was not possible to avoid a significant amount of privacy loss in the form of actual meetings. This was undoubtedly because of the greater number of communications required to find an optimal solution.

It has already been pointed out that in the present situation, where preferences are not communicated explicitly, the existence of more sophisticated methods for determining preferences will probably have a limited effect on the pattern of results. The present methods are, of course, compatible with such methods and, in fact, they should be quite general. In addition, the quality of solution returned, especially with the step procedures, would seem to make this approach competitive with approaches that involve more elaborate conceptions of solution quality (e.g. (Ephrati, Zlotkin, & Rosenschein 1994)).

References

Ephrati, E.; Zlotkin, G.; and Rosenschein, J. S. 1994. A non-manipulable meeting scheduling system. In *Proceedings, 13th International Distributed Artificial Intelligence Workshop*, 105–125.

Franzin, M. S.; Freuder, E. C.; Rossi, F.; and Wallace, R. 2002. Multi-agent meeting scheduling with preferences: efficiency, privacy loss, and solution quality. *To appear in AAAI02 Workshop on Preferences in AI and CP*.

Freuder, E. C.; Minca, M.; and Wallace, R. J. 2001. Privacy/efficiency tradeoffs in distributed meeting scheduling by constraint-based agents. In *IJCAI-01 Workshop on Distributed Constraint Reasoning*.

Garrido, L., and Sycara, K. 1996. Multi-agent meeting scheduling: Preliminary experimental results. In *Proc., Second International Conference on Multi-Agent Systems, ICMAS-96*. Menlo Park, CA: AAAI Press.

Luo, X.; Leung, H.; and Lee, J. H. 2000. Theory and properties of a selfish agent protocol for multi-agent meeting scheduling using fuzzy constraints. In *Proc., 14th European Conference on Artificial Intelligence, ECAI-00*, 373–377. Amsterdam: IOS.

Sen, S.; Haynes, T.; and Arora, N. 1997. Satisfying user preferences while negotiating meetings. *Internat. J. Human-Computer Stud.* 47:407–427.

Silaghi, M.; Sam-Haroud, D.; and Faltings, B. 2001. Consistency maintenance for ABT. In Walsh, T., ed., *Principles and Practice of Constraint Programming - CP2001, LNCS 2239*, 271–285. Berlin: Springer.

Wallace, R. J.; Freuder, E. C.; and Minca, M. 2002. Consistency reasoning and privacy/efficiency relations in multi-agent systems. *Submitted for publication*.