Consistency reasoning with possibilites in multi-agent problem solving

Abstract

In cooperative problem solving, communication among agents may be restricted, either due to privacy concerns or to limitations of the system. This problem can be overcome if agents can make deductions from communications they receive, based on reasonable assumptions about sets of possible values for critical items of information. The major mechanism of infererence that we propose involves an extension of CSP technology that uses information about possible values in an unknown CSP. This strategy is applied to "common assignment problems", where agents must find solutions to individual problems, part of which they share in common. We show that *modal* information in this form in combination with arc consistency processing can speed up search for common assignments. This entails an important tradeoff, in that this form of possibilistic reasoning requires a modicum of actual (private) information to be maximally effective. Such information may either be given explicitly, but the machinery we describe can also deduce it. In addition, this form of reasoning can also be used to detect some forms of false communication.

1 Introduction

Constraint satisfaction is a powerful technology that has been successfully extended to distributed artificial intelligence problems [Yokoo, 1998]. Within the multi-agent setting, which is often the framework for distributed AI, new problems arise when agents have a degree of independence. Heretofore, most systems of this sort have been built on the assumption that agents will be completely open about communicating information that they have and that might be relevant to solving a problem [Luo *et al.*, 2000] [Sen and Durfee, 1995] [Sen *et al.*, 1997]. However, this may not always be the case in such settings; agents may want to maintain their privacy as much as possible while still engaging in collaborative problem solving [Garrido and Sycara, 1996]. Or, it may simply not be reasonable in a situation to communicate all information relevant to the task at hand.

As a result of these limits to communication, agents may need to operate under conditions of partial ignorance. In such cases, even though critical information may not be known, agents may be able to reason in terms of sets of possibilities, such as the set of possible values for a known variable. In this paper we show how this can be accomplished, by applying constraint-based reasoning to such possibility sets. This entails the use of a new formal structure that supports consistency reasoning under conditions of partial ignorance. The soundness of this approach is proven in detail elsewhere [Present Author, 2002], although we will summarize that work here. This paper demonstrates that this approach can improve efficiency where agents would otherwise need to proceed more or less blindly. It also explores other potential benefits, such as deducing 'hard' information that would otherwise be inaccessible and detecting false communications.

We demonstrate the efficacy of our methods in some simplified situations, which we think can be extended to more realistic scenarios. We use an independent agent paradigm, where agents communicate with each other to solve a problem of mutual interest. In addition, we are interested in an important variant of distributed problem solving where each agent has its own problem to solve but a portion of the solutions to all problems must be mutually consistent. Here, this will mean that portions of individual solutions must be the same for all agents. We call this the *common assignment problem*. We apply this approach to a version of the meeting-scheduling problem, where agents have pre-existing schedules but need to add a new meeting that all of them can attend. We also consider graph coloring problems where each agent has its own graph to color but part(s) of the graph must be colored in common with other agents.

Using a simplified version of the meeting scheduling problem, we first consider the kinds of information that agents can derive about other agents' schedules in the course of a meeting scheduling session. We show that such information can be encoded using ideas from standard modal logic, in which terms with modal properties are treated as as kind of CSP value. We show that such information gathering can be enhanced by consistency processing based on general temporal relations assumed to hold for any schedule. Finally, we apply these methods to the study of privacy/efficiency tradeoffs and show that the form of the latter is critically dependent on the capacity to make deductions to limit possibilities. These ideas can be naturally extended to other forms of the common assignment problem.

In Section 2 we describe the basic problem for our agents. Section 3 introduces the idea of "shadow CSPs" based on modal information, that are used to represent an agent's current knowledge about another agent's schedule. Section 4 discusses the concept of privacy loss and how to measure it. Section 5 gives results of experiments that test effects of different levels of communication and forms of knowledge (actual and possibilistic) on efficiency and privacy loss. Section 6 describes heuristics based on possibilistic knowledge that can enhance efficiency without requiring extra information in communications. Section 7 discusses how false communications can be discovered with our deductive techniques. Section 8 discusses related work. Section 9 gives conclusions.

2 Meeting Scheduling as a Testbed

The meeting scheduling problem that has often been studied in the context of multi-agent systems has a number of features that make it both of general interest and a useful domain in which to examine the common assignment problem. It is open-ended in that meeting assignments can be relatively unrestricted, but at the same time there are constraints due to time and space that can be taken hold across any pair of assignments.

In the scheduling problem we consider, 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 all agents can attend given their existing schedules and constraints on travel 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.

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.

Note that if we wish to make this a more realistic problem, we can consider geographical sectors instead of specific cities (supposing the set of cities is indefinite), and hard as well as soft constraints (imprecise but bounded travel times). But here we will restrict ourselves to the simpler case, which we believe is still interesting. In regard to schedule changes, it is not unrealistic to expect that during a single problem solving run the existing schedules will remain fixed.

The basic constraints are times (in hours) required for travel between meetings in different cities, as indicated in Figure 1. Times between cities within one region (Western Europe or former Eastern Bloc) 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.

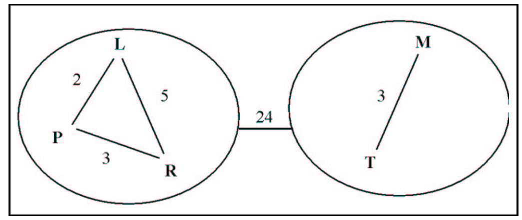

Figure 1: Time constraint graph. Cities are London, Paris, Rome, Moscow and Tbilisi. Further details in text.

3 Communication and Inference (About Other Agents' Knowledge)

In the situation we consider there are three basic messages: a proposal, consisting of a city, day and hour ("Paris on Monday at 3 PM"), an acceptance, and a rejection. In addition, under some experimental conditions agents give reasons for their rejection by communicating one or all conflicts ("I have a meeting in Rome on Monday at 5 PM."). This allows agents to reveal small amounts of private information that has the potential to speed up search. A similar strategy, that of exchanging partial calendars, was used by [Garrido and Sycara, 1996], although the present strategy is more focused in that the information communicated is related to a specific rejection.

In the present situation, agents are operating under partial ignorance, since a solution to each agent's problem depends on $k-1$ other problems that it does not know directly. However, agents can represent common features of these problems (here, temporal and geographical features) and then make deductions from communications received from other agents that are based on their problems to improve overall efficiency.

To this end, each agent maintains "views" of other agents' schedules. A view can be updated after each communication from another agent. Since an agent does not actually know the other schedules, but it does know the basic constraint graph and meeting rules, it can deduce facts about other schedules in terms of *possible* values at any point in the session. This information can be used to guide selection of proposals, both deductively and heuristically.

We model information about possible values in the same way as ordinary CSP values, namely, as a triple, (X, D, C) consisting of variables X, domains D and constraints C. (In the present problem, C involves only unary constraints.) Taking time-slots as variables, we have possible values both for meetings that another agent may have in its schedule and for meetings it may be able to attend. We call these "possible-has-meeting" and "possible-can-meet" values, respectively. Considered more generally, the former represent possible existing assignments in an actual, unknown CSP, while the latter represent possible future assignments in the same CSP. Each type of value is associated with a separate CSP, called a "shadow CSP" to indicate the close semantic relation between it and the actual (unknown) CSP of another agent. In addition, we employ another shadow CSP whose values are possible causes for meeting rejections. The values in this shadow CSP also represent possible existing assignments, so it is closely related to the set of possible-has-meeting's.

The possibilistic character of the domain values of shadow CSPs can be represented by the possibility operator, $\Diamond$, from standard modal logic [Hughes and Cresswell, 1968]. In this paper, these *modal values* are always referred to in conjunction with the modal operator, e.g. $\Diamond x$. Inferences involving such values are subject to the rules of modal logic as well as ordinary truth-functional logic. (The connection to modal logic is spelled out in more detail in [Present Author, 2002].)

Inferences consistent with the rules of modal logic are subject to some important restrictions in comparison with inferences based on ordinary CSP values. For example, given a constraint between two variables that prevents values x and y from appearing together in a solution, from the presence of x we can infer $\neg y$. But, while from $\Diamond x$ we can infer $\Diamond \neg y$, we cannot then infer $\neg \Diamond y$ (and certainly not $\neg y$). More importantly, while from $\neg \Diamond x$ we can infer $\neg x$, we cannot in general make the opposite inference. This would seem to severely limit our capacity to make deductions regarding possibilities.

However, we can make a deduction like the latter if we adopt a closed world assumption. This, in fact, is what one ordinarily does when representing problems as CSPs, where the domains in question are considered closed worlds. In this case, $\neg x$, entails $\Box \neg x$, where $\Box$ is the necessity operator. But, by a standard modal equivalence,

$$\Box \neg x \;\equiv\; \neg \Diamond \neg \neg x \;\equiv\; \neg \Diamond x.$$

Under this assumption, therefore, an agent can make deductions from whatever 'hard' information it can glean during the scheduling session in order to refine domains in the shadow CSPs.

For instance, when another agent makes a proposal or accepts a proposal, since we are assuming honest communications, the recipient of this information knows that the other agent has no meeting in that slot (we call these "open slots"). Therefore, up to five possible-has-meeting values can be deleted, since under closed world assumptions $\neg x$ entails $\neg \Diamond x$. Moreover, if x implies $\neg y$, then we can also deduce $\neg \Diamond y$; hence, arc consistency based on the original constraint graph can be used to delete possible-has-meeting values for other variables (i.e. other cities in nearby time slots).

In addition, for possible has-meetings, it is possible to make inferences back to the actual values. This follows because $\neg \Diamond x$ implies $\neg x$ under standard assumptions. In this case, if, for all cities x associated with a single time-slot, we have inferred $\neg \Diamond x$, we can then infer that the agent has no meeting at that time, i.e. it has an open slot.

From a simple rejection, an agent can only infer that one possible-can-meet value is invalid. However, if an existing meeting is given as a reason, it is possible to remove up to five possible-can-meet and four possible-has-meeting values from that time slot, and, using arc consistency reasoning, to delete other possible-can-meet and possible-has-meeting values based on the known constraints between hard values.

From a rejection, an agent can also deduce possible-cause values, i.e. it can deduce a set of possible causes for that rejection. Since possible-cause values must be included in the set of possible-has-meeting values, this subset relation allows agents to prune sets of the former kind of value, so that knowledge about another agent's schedule can be refined.

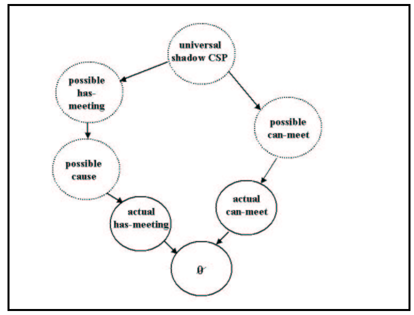

Figure 2: Structure of shadow CSP system for the meeting-scheduling problem. Arrows represent the set inclusion (or "realization") relation holding between the domains of super- and subordinate CSPs.

To show that this deductive machinery is valid, we set up a system composed of the actual and shadow CSPs, establish certain requirements for the system to be well-structured, and show that these requirements are satisfied at every stage of the search process. We add to the present set of CSPs a supremum whose domains contain the set of all possible domain values. Then, the corresponding domains of these CSPs can in each case be arranged in a partial order under the relation of set inclusion (see Figure 2). (Corresponding domains are associated with the same variable.) The proof that the system remains well-structured shows that, with the present communications and rules of inference, this relation remains achievable at every step in search [Present Author, 2002]. Informally, this means we cannot deduce a value that is not (potentially) contained in the corresponding domain of all superordinate CSPs. In other respects, the system is sound because it relies on standard logic plus the closed world assumption.

For our meeting scheduling problem, at the beginning of a session when an agent knows nothing about other agents' schedules, all domains of the possible-has-meeting and possible-can-meet CSPs, as well as the "universal" shadow CSP, have five modal values, corresponding to the five possible cities. The possible-cause shadow and the actual CSPs have empty domains. As search proceeds, values are deleted from the domains of the first two CSPs, while they are added to the last three. Essentially the same procedures can be used for multi-agent graph coloring, although under some conditions possible constraints can also be brought into this framework.

4 Characterizing and Measuring Privacy Loss

In our meeting scheduling problem, the most obvious measures of privacy loss involve meetings and open slots communicated or deduced. In this case, a viable measure of privacy loss is in terms of information gain. This is supported by cases such as the following. Suppose 100 possibilities can be discarded in two situations, in the first from an original set of 400, in the second from an original set of 101. It certainly appears that the latter constitutes a greater loss of privacy.

This suggests the following conceptualization. We consider cases where agent a has an agenda with respect to another agent k. Part of this agenda involves knowing something about that agent. For example, in the present meeting scheduling situation a may want to determine whether k will be in Rome on Wednesday afternoon. Now, for each agenda, there is some relevant set of possibilities. If that set can be reduced, privacy loss is related to this reduction in number. In this case, we refer to an "original-relevant-set" and a "resultant-set" of possibilities, and privacy loss is defined as:

$$\mathrm{privacyloss} = \log_2(|\mathrm{original\_relevant\_set}|) - \log_2(|\mathrm{resultant\_set}|)$$

which can, therefore, be measured in bits of information.

In the present agent scheduling situation, we assume that the agenda is simply one of information gathering. Since agents gather two kinds of information, when other agents have meetings and when they can meet, they are considered to have two kinds of agenda. In both cases, the original-relevant-set consists of all possible meetings plus the possibility that there is no meeting (or that the agent cannot meet in any city). This gives six possibilities for the original-relevant-set *for each slot in each view*. Note that, with respect to privacy loss we do not treat modal and actual values differently; the same framework handles a reduction in possibilities whether or not actual values are discovered.

It is still of additional interest to know how many actual items of information are discovered in the course of a problem solving session, e.g. the number of actual meetings and open slots. Therefore, this information was also collected in the experiments to be described.

5 Efficiency and Privacy in Meeting Scheduling: Experimental Results

Although the previous sections and the work referred to demonstrate the soundness of the system, we still need to know how well it will perform in practice. Will reasoning based on possibilities improve efficiency? Under what conditions and to what degree? Under what conditions is there a privacy/efficiency tradeoff? In addition, this framework raises new issues regarding privacy: to what extent can agents 'invade' each other's privacy by making deductions from possible to actual information?

5.1 Experimental methods

To study these issues, a system was built in Java to conform with the description of the meeting scheduling problem described in Section 2. It allowed the user to vary the kinds of knowledge used: actual knowledge and the three types of possibilistic knowledge, to optionally enhance knowledge gathering by employing arc consistency techniques, and to independently vary the way that knowledge is used in selecting proposals (proposal strategies). The proposal strategies of interest are: (i) no use of information (other than avoiding previous proposals), (ii) use of knowledge accumulated (of whatever kind) to eliminate candidate proposals, (iii) additional use of heuristics based on possibilistic knowledge (described in the next section). In addition, the level of communication could either be restricted to the three basic messages (propose, accept, reject) or extended to include a reason for each rejection (meeting in conflict), or to include all such reasons. The number of solutions to find could also be specified, and whether that number could actually be found (no-solutions cases).

In experiments reported here, a "round-robin" protocol was used in which agents take turns as the coordinator who proposes a meeting and evaluates the responses of other agents. This protocol has the merits of being democratic and allowing agents to update their knowledge efficiently. A similar protocol was used in [Garrido and Sycara, 1996]. A "one-coordinator" protocol, where one agent makes all proposals, gave similar results, although differences were smaller.

Most experiments involved three agents, and the number of initial meetings varied from 5 to 40. This was supplemented by experiments with more agents. In each experiment, an individual test run began with random generation of schedules followed by a series of proposals which continued until one was found that was acceptable to all agents. There were 500 runs per experiment.

In these experiments, the efficiency measure was number of proposals per run, averaged over an experiment. Privacy loss was measured for both possible existing and possible future assignments, using the entroptic measure described in the last section; number of actual meetings and open slots identified were also tallied. All privacy measures were averaged over agent views as well as runs. (There are $n \times (n-1)$ views for n agents.) Differences in means were evaluated using either t-tests or analyses of variance.

5.2 Basic results

In the 3-agent experiments under baseline conditions, mean proposals per run varied between 5 and 16, depending on the number of meetings in the agents' own schedules. Problems were hardest in the middle of the range, because with more meetings solutions are rarer (making the problems harder), but there are also fewer candidates per agent (eventually making problems easier). Since the problems were hardest around 15 initial meetings, this value is used for comparisons in this paper, although the same pattern of differences was found for all values.

Figure 3 shows results for efficiency and actual data discovered under various conditions of communication, information gathering and inference. Figure 4 shows the basic measure of privacy loss both for possible existing and possible future meetings.

The main conclusions from these experiments are as follows (based on statistical comparisons with the corresponding baseline condition):

- explicit communication about meetings *does not* improve efficiency, despite giving up 'hard' information (1- and all-conflicts conditions).
- efficiency is not improved by deriving modal information even when enhanced by arc consistency reasoning, which does allow deduction of many possible-has-meeting nogoods (know and know-ac conditions).
- a combination of explicit communication, use of modal information, and arc consistency processing results in a

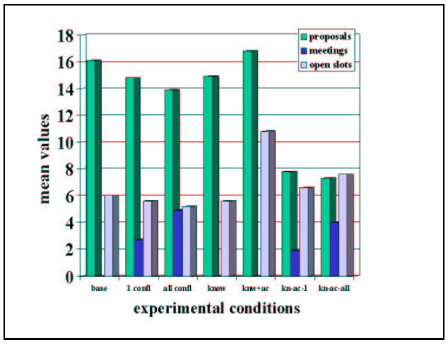

Figure 3: Efficiency and loss of ‘hard’ information concerning meetings and open slots under various conditions of communication, knowledge, and consistency reasoning. Three agents, each with 15 initial meetings.

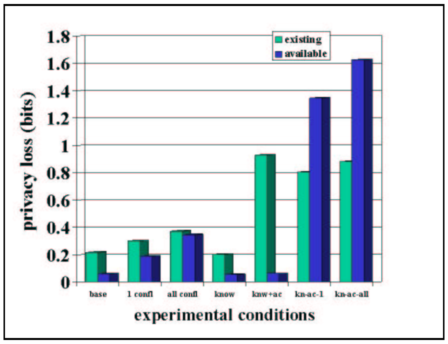

Figure 4: Information measures of privacy loss under various conditions of communication, knowledge, and consistency reasoning. Three agents, 15 initial meetings.

marked improvement in efficiency (know-1 and know-all conditions). In this case, large numbers of possible-can-meet nogoods are deduced, reflected in the results in Figure 4 for future meetings. Further experiments (not shown) indicate that most of this improvement can be obtained when a fraction (1/3-1/2) of rejections are accompanied by a single reason, with less privacy loss.

6 Heuristics Based on Possibilistic Reasoning

Information on possibilities gathered during search can also be used to choose which values to propose. In the meeting scheduling situation this involved possible-cause values, and more specifically, storing these values in relation to the rejection that gave rise to them. In this case, as possible-cause values are pruned, the lists often approach one value. This has two implications.

First, if such a set is reduced to one value, it can be concluded that this meeting must be in the schedule of the agent that made the rejection. Since many rejections are based on more than one conflict, empirical tests are needed to determine whether this strategy will work in practice. In such tests, we found that agents using modal values in this manner dicovered an appreciable number of actual meetings, up to 2-3 per view per run for the longer runs, and averaging over 0.1 per view per run for 15-25 initial meetings. This shows that it is possible for an agent to deduce another agent’s meeting even when no explict meeting information is exchanged.

Second, possible-cause values stored in this manner can indicate which proposals are likely to fail, based on potential conflicts. This idea was used to generate the following heuristic. Given a candidate proposal, if possible-cause values associated with the nearest rejections before and after would also conflict with the new proposal, then the latter is (temporarily) avoided. It was felt that, other things being equal, the nearest rejection is more likely to be the most relevant, but refinements may be possible using more rejections.

Both ideas can be incorporated into the shadow CSP system, in a way that maintains its soundness [Present Author, 2002]. For the meeting scheduling problem, there are up to 70 shadow CSPs for possible-cause values, each subordinate to a subgraph of the possible-has-meeting shadow CSP. In this case, to maintain the set inclusion requirements within the shadow system, each possible-cause shadow CSP only includes domains of variables adjacent to the one associated with a rejection.

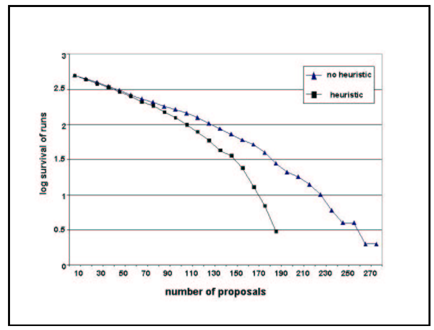

Figure 5: Performance on efficiency measure with and without heuristic based on possible causes, for problems with 20 agents each having 7 initial meetings. Log survivorship curves showing number of unfinished runs after k proposals have been made. Each curve is based on a total of 500 runs.

In empirical tests, this heuristic reliably improved efficiency, and this often occurred before actual meetings were deduced. Differences in means were modest but statistically significant, e.g. for 15 initial meetings, there was an average improvement of 20%. However, this occurred because the range of values for number of proposals was drastically reduced. This is shown in Figure 5 for a much harder problem involving 20 agents and seven initial meetings. The figure shows the number of unfinished runs after k proposals, where k ranges from 0 to the maximum number in the sample. The maximum number of proposals was 275 without and 188 with the heuristic. And despite extra processing, total realtime to complete the experiment was reduced by a factor of 2-3.

When backtracking is required, either to solve problems requiring more than one meeting or to show there is no solution, these methods have an even greater effect; in these cases the effect on average number of proposals is comparable to the case with inference and explicit communication described earlier. (For example, for problems with 8 agents having 8 meetings and with *no* common solution, means of 252 and 126 proposals were found for baseline and heuristic conditions, respectively.) At the same time, there is more privacy loss since agents have more opportunity to refine their possible-cause lists. In preliminary work with graph coloring which also requires backtracking, it has been possible to obtain efficiency improvements with the same tradeoffs with respect to privacy loss.

7 Detecting False Communications

Inferences that agents make on the basis of communications from other agents depend in part on the honesty of the other agents. This is a general problem and not limited to a system that reasons in terms of possibilities.

If we assume that agents behave reasonably, although not necessarily honestly, then in the meeting scheduling situation, we can assume that proposals and acceptances are valid; otherwise an agent making a false communication would be in a compromising position if that meeting was the one agreed upon. False rejections are not as risky; moreover, it seems reasonable that if an agent has some reason to reject a time and place when it could actually attend, it will do so, especially under conditions of limited communication. In this case if there are other possible meeting times a (valid) solution may still be found, but it is still possible to falsely conclude that there is no common meeting time.

In fact, false rejections can sometimes be detected by the present system, using the reasoning described in the last section. If all of the possible causes associated with a rejection can be eliminated (because the corresponding possible-has-meetings have been eliminated), then it can be concluded that the rejection was invalid.

Tests of this have been carried out successfully within the present experimental testbed. One agent in the set is designated as 'bad' and one of the (actual) common meetings is chosen for rejection. Under these circumstances, in about 2/3 of the cases where the possible cause set associated with this rejection is restricted to one element (leading to a false conclusion), it is further restricted so that there are no elements. This serves to 'unmask' agents making false communications and to discover invalid inferences.

8 Comparisons with other work

Recently, other investigators have considered situations where agents reason about other agents' knowledge [Larson and Sandholm, 2002]. In this work, knowledge of possibilities is represented by probability distributions, to which Bayesian reasoning can (potentially) be applied. Although this is a viable (and interesting) alternative to the present approach and is likely to have greater generality, it will often be more difficult to maintain and update, and its capacity for rapid, effective deductions is probably weaker than the present system.

9 Conclusions

In this work we have shown how consistency reasoning can allow agents to find solutions more efficiently under conditions of ignorance. In doing so, we have developed a new CSP formalism that makes the possibilistic aspects of such reasoning more explicit and more coherent. The present system is also efficient because processing is mostly a matter of storing and looking up information in arrays.

Significant improvements in efficiency were obtained either by giving up a limited amount of private information in communications or by linking modal information to specific communications, which could in turn be associated with specific elements in the actual CSP. In both cases, possibilistic information was essential: in the former case, the critical feature was the deduction of possible-can-meet nogoods; in the latter case, it was a combination of possible-cause and possible-has-meeting values.

In the present situation we have established some conditions for improving efficiency, and in so doing we have also established some parameters for privacy/efficiency tradeoffs. Most importantly, we found that such tradeoffs do not emerge unless communication is accompanied by effective reasoning. Given these conditions, the tradeoff can be modulated by variations in communication and even to a degree finessed by sophisticated heuristic techniques.

References

[Garrido and Sycara, 1996] L. Garrido and K. Sycara. Multi-agent meeting scheduling: Preliminary experimental results. In *Proc., Second Int. Conf. on Multi-Agent Systems, ICMAS-96*, pp. 95–102, Menlo Park, CA, 1996. AAAI Press.

[Hughes and Cresswell, 1968] G. E. Hughes and M. J. Cresswell. *An Introduction to Modal Logic*. Routledge, London and New York, 1968.

[Larson and Sandholm, 2002] K. Larson and T. Sandholm. An alternating offers bargaining model for computationally limited agents. In *Auton. Agents Multi-Agent Syst., AAMAS'02*, pp. 135–142, New York, 2002. ACM.

[Luo *et al.*, 2000] X. Luo, H. Leung, and J. H. Lee. Theory and properties of a selfish agent protocol for multi-agent meeting scheduling using fuzzy constraints. In *Proc., 14th European Conf. on AI*, pp. 373–377, Amsterdam, 2000. IOS.

[Present Author, 2002] Present Author. Representing possibilities in relation to constraints and agents. In *Logics in Artificial Intelligence, JELIA 2002, LNCS 2424*, pp. 407–418, Berlin, 2002. Springer.

[Sen and Durfee, 1995] S. Sen and E. H. Durfee. Unsupervised surrogate agents and search bias change in flexible distributed scheduling. In *Proc., First International Conference on Multi- Agent Systems, ICMAS-95*, pp. 336–343, 1995.

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

[Yokoo, 1998] M. Yokoo. *Distributed Constraint Satisfaction. Foundations of Cooperation in Multi-Agent Systems*. Springer, Berlin, 1998.