Dynamic negotiation

Charles L. Ortiz, Jr.
Artificial Intelligence Center
SRI International
Menlo Park, CA 94025
ortiz@ai.sri.com

Timothy Rauenbusch
Division of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02139
tim@eecs.harvard.edu

Abstract

We present an anytime algorithm for adapting a negotiation to a dynamically changing environment in which either new tasks can appear or the availability of resources can change during the negotiation. We use a particular negotiation algorithm, which we call Mediation, in which problem solutions are suggested by a mediator to a team of bidders. In Mediation, agents can bid in the context of a particular set of other tasks; both positive and negative task interactions can be taken into consideration. In addition, an agent's bid need not be restricted to a single value but rather can span a range of values. Bids are also augmented with information that compactly captures important elements of an agent's local state in the form of a description of potential positive and negative interactions with other commitments. We claim that agents involved in a negotiation can make better use of information from prior interactions when bids are more informative in the way described. We provide support for our claim through a set of experiments in a real-time sensor allocation problem.

The development of negotiation protocols to support the distributed allocation of tasks or resources within a multi-agent system has assumed that the set of issues over which agents negotiate are defined in advance: there is no provision within existing algorithms for modifying the set of issues during the negotiation and for adapting the ongoing negotiation accordingly. Most realistic problem domains, however, are dynamic: while agents are negotiating over the distribution of a set of tasks, for example, a new task might appear. Existing protocols would either require that the allocation of the new task be postponed until the current negotiation was complete or require that the negotiation be interrupted and re-started in the context of the augmented set of tasks. Adopting the first option would neglect the potential for exploiting possible positive interactions between the old set of tasks and the new task, since they are allocated separately. In the second option, all of the work performed in the first negotiation is lost and, in fact, there can be no guarantee that the process will ever converge since new tasks could continue to appear. Ideally, it should be possible to adapt the ongoing negotiated allocation to the new task.

Another dynamic constraint that is often imposed on negotiation involves time: execution might need to commence before the best allocation is decided upon. Many negotiation protocols address this issue by casting the protocol in an anytime form. Other factors that can impact a negotiation include the appearance or disappearance of resources during the negotiation. For example, agents not currently involved in a negotiation might unexpectedly complete the tasks to which they had committed and thereby become free to offer their services on the task being negotiated. Alternatively, if an agent becomes disabled, it should be possible to repair the current allocation as reflected by the current state of the negotiation rather than restart the negotiation from scratch.

Communication delays can also introduce difficulties. Consider auction-based negotiation protocols which assume that before a new task is announced for bid, existing tasks under negotiation have already been awarded. Now, consider a situation involving two auctioneers, A_1 and A_2 and two tasks, t_1 and t_2. A_2 might request a bid on t_2 from a bidder who is awaiting announcement of an award on an outstanding bid for t_1 from A_1. The appropriate bidding strategy for the bidder on task t_2 in this situation is then unclear. If the bidder assumes that the first task will be won, it will lower its bid on the second task. Assuming that it will not win the first task will result in a higher bid. If the agent is wrong in the first case, the bid will be overly pessimistic whereas if the agent is wrong in the second case, the bid will be overly optimistic. We avoid this problem by allowing agents to bid on *ranges* of values.

The problem of dynamic negotiation is reminiscent of that faced by conventional single-agent planning systems in the 1980's: then it was assumed that complete executable plans could be generated before execution commenced; the introduction of new goals during the planning process required a complete replanning. It was quickly discovered that such a view was unrealistic for any but simple toy domains.

This paper makes three main contributions: (1) bids in a negotiation need not be restricted to a single value but rather can span a range of values, (2) reasons can be appended to bids that explicate potential positive and negative interactions with other commitments, and (3) ongoing negotiations can be adapted to a dynamically changing environment. We show how negotiators can make better use of information from prior interactions when bids are more informative in the way described.

The next section describes a motivating sensor allocation

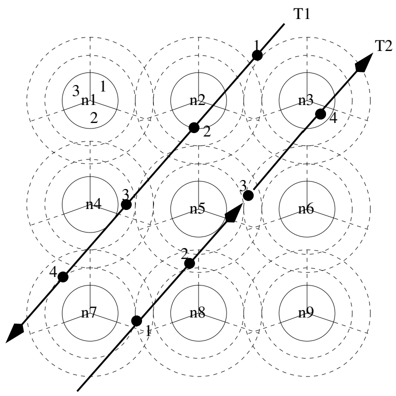

Figure 1: Multisensor tracking.

problem. We then describe a negotiation protocol called Mediation, in which a allocations are proposed to bidders, who can then submit complex, informative bids. We then report on experimental results and related work.

Motivating example

We have been investigating the issues surrounding the problem of dynamic negotiation in the context of a real time distributed resource allocation problem involving multisensor tracking. An example is shown in Figure 1. The figure depicts an array of nine doppler sensors. Each sensor has three sectors associated with it, labeled $\{1, 2, 3\}$. A sensor can turn on a sector and take both frequency and amplitude measurements to determine velocity and distance. A sensor can only have one sector on at a time, however. The farther away the target is from the sensor, the lower the quality of the measurement. At least two sensors are necessary for estimating the location of an object; three sensors are desirable for obtaining a good-quality estimate. Tasks can interact: for example, sectors require a 2 second warm-up time; therefore, an agent can benefit from tracking two targets in sequence because of the saved warm up time. Finally, two objects appearing in the same sector and at the same time cannot be discriminated.

Tasks can appear dynamically; the figure shows projected paths — based on initial localization, direction and velocity measurements — for two targets, $t1$ and $t2$. The problem is to allocate, in a distributed manner, a set of sensors along the paths of both targets. Each path is discretized into a set of space-time points along the path (indicated in the figure by small dark circles). We assume that agents are cooperative and work together to get the best possible measurements.

For presentation purposes, we consider a simplified version of this problem in which only one sensor is necessary for obtaining an accurate measurement. This simplification can be mapped into the original problem straightforwardly: we map each task point to three tasks. For example, if t is a point on the path of a target, then any proposal involving t will instead refer to a larger set that includes (t_1, t_2, t_3). Proposals must be augmented with rules so that the agent proposed for any t_i does not match the agent of some t_j, where $i \neq j$.

Dynamic Mediation

We can view agents as negotiating over some set of *issues*, such as tasks or resources. In the domain we have described, an issue might correspond to the task of tracking a target over a particular time interval or to a particular resource, such as the choice of communications medium. The goal of a negotiation is then to arrive at some resolution regarding those issues. We assume that each issue can be assigned an *option*; for example, a particular sensor agent and sector or a particular communications channel. In the examples and experiments that we will discuss, we will focus primarily on issues that correspond to tasks and options that correspond to particular agents. We will refer to the set of issues and options collectively as the *negotiation space*.

In the Mediation algorithm we will describe, a mediator announces proposals to a set of bidders. A *proposal* p corresponds to one possible way a group, G, of agents could adopt options for issues. For example, the proposals $(1, 0, 6, 5)$ might correspond to a proposed assignment of agent 1 to issue (or task) 1, agent 0 to issue 2, agent 6 to issue 3, etc. Each proposal provides a *context* for each bidder. In the example, agent 1 can see that it will be bidding on task 1 in the context of agent 6 performing task 3. In many domains, agent 1's bid might differ depending on whether or not agent 6 performs task 3 (e.g., task 3 might involve letting everyone know the location of a particular target at some instant).

Each agent g has a *utility function*, u_g, that maps each element in $\mathbf{P}$ to a real value, $u_g : \mathbf{P} \mapsto \mathcal{R}$. The level of detail encoded by each proposal is domain dependent. The algorithms described in this paper assume that each proposal encodes sufficient detail to enable each agent to evaluate its utility function. There is a global objective function that enumerates the desirability of elements of $\mathbf{P}$, given the utility function of each agent. The objective function f takes a proposal and a group of agents as input and returns a real value. In this paper we assume the group objective is social welfare maximization.

Bid format

Rather than reducing a bid for some task, T, to a single value, Dynamic Mediation makes use of a richer bid format which allows a bidder to compactly exchange relevant information about its local state to a mediator. The mediator can then choose to use that information during its search process (described below). A bid is formatted in the following way:

$$\langle [V_1, V_2], C \pm A \rangle \tag{1}$$

where V_1 represents the normative value for T, and V_2 represents a higher or lower value due to task A's positive (+) or negative (-) interaction with the bidder's existing commitment, C. A can be thought of as representing the bidder's *reason* for the higher or lower value, V_2. As an example, suppose that agent 3 receives the proposal, $(4, 3, 0, 2)$. It might submit the following bid: $\langle [-\infty, 6], t_5 - t_2 \rangle$ which states that agent 3 is bidding $-\infty$ based on the fact that its commitment to an outside task t_5 conflicts (interacts in a negative way) with task t_2.[1] If it was not committed to t_5, that proposal would have a utility of 6. The mediator can then use this information in the next round of proposals: for example, it might expand the number of issues and submit the proposal $(4, 3, 0, 2, 6)$ which asks agent 3 to estimate a value for task t_2 given that agent 6 will be doing task t_5. In general, the negotiation space might change because of either a *negotiation event* or a *domain event*: the above is an example of the former in which the mediator attempts to resolve a conflict by adding an issue; the latter might come about because a new task appears in the environment.

Mediation

The Mediation algorithm (Rauenbusch 2002) implements an iterative hill climbing search through a negotiation space. It was designed for negotiation environments where relevant preference information is distributed among agents and where centralized collection of preference information was infeasible due to computation or communication constraints. It is an anytime algorithm in which an agent, called the mediator, successively announces complete proposals from the negotiation space to the agents. After each announcement, the agents respond with a message indicating their bid for the announced proposal. The mediator computes the quality of the proposal based on the objective function and the agents' bids, and keeps a record of the best proposal found so far, to be returned as the negotiated outcome when the negotiation is complete.

The Mediation algorithm has been extended so that it is tolerant to faults to the mediator agent. Instead of communicating only with the mediator, agents broadcast their bids to the entire group. If the mediator is disabled, a new one can be chosen through some voting mechanism; it will have a current record of the negotiation and can proceed where the disabled agent left off.

Adapting the negotiation space

A desirable property of Mediation in its application to dynamic negotiation is that the negotiation space can be adjusted on the fly, even after the negotiation has begun. The mediator can make any of the following four adjustments to the negotiation space. The search space is either expanded (adding an issue or option) or narrowed (removing an issue or option). The mediator can combine steps (for example, removing an issue while adding an option).

Adding an issue The set of options remains the same but the negotiation space is expanded to include an additional issue. For example, if the last proposal was $(1, 0, 5, 6)$ involving 4 tasks, then the next proposal might be a 5-tuple. The mediator might either be responding to a new task, such as a new target appearing in the environment, or simply adding an issue to increase the chances of finding a good solution (recall the example given earlier in the discussion on bid format).

Removing an issue The search space, with respect to the set of issues currently under consideration, is reduced; the new proposal will involve fewer tasks ranging over the same set of options. The mediator might either be responding to information that a task that was being negotiated has been completed or no longer of relevance, or the mediator might have identified a conflict between the resolution of two issues.[2] An example from the multisensor tracking domain would be one in which a tracking gap was left along the projected track (resulting in lower social welfare, but still greater than zero).

Adding an option If a new resource becomes available, the mediator might augment the set of options available for an issue. It might decide to expand the set of agents that could contribute to the set of tasks currently under negotiation. In our domain, this latter decision might come about in an incremental way: the mediator might first involve only the sensor agents closest to a target's projected path; if unavoidable conflicts remain (i.e., negative task interactions), the mediator might expand the team to include other sensors even though they might be further from the projected path. Although an allocation from such a set might result in lower track quality (lower utility), such an allocation might represent the only one that could cover all the tasks.

Removing an option In a dynamic environment, faults can occur and agents can become disabled. In response to such events, a mediator should look for substitute agents. While negotiating, the mediator might also discover that a task conflicts with another agent's commitments; the mediator may then decide to remove that agent as an option in the next round of proposals. In doing so, the mediator is essentially adding more constraints to the problem, thereby narrowing the negotiation space.

Example: task contention

Consider Figure 1. We will use the notation Sn/s to refer to sector s of sensor n and the notation t_p^r to refer to the point p on projected path r and t^r to refer collectively to the tasks on path r. Suppose a negotiation involving task t^1 has been ongoing for n rounds with the current best proposal, $P1 = (S2/1, S2/2, S5/3, S7/3)$. Target t^2 then appears and is projected to follow the path shown. Suppose that the best allocation for t^2, assuming that there were no conflicts, would be $P2 = (S7/1, S8/3, S5/1, S3/2)$. However, if tasks t_3^1 and t_3^2 occur at the same time, a conflict involving sensor $S5$ will result.[3] We refer to this as an instance of *task contention*. There are two cases to deal with. In the first case, if we assume that the same mediator is coordinating the negotiation for both targets, then the mediator will be aware of the commitment of $S5$ to task t_3^1. If it is aware of

[1] Note that the bidder can bid on the entire set of tasks, not just the one assigned to it. We do not pursue that possibility here; its potential use lies in the fact that an agent might have better knowledge of the cost of performing some task than another agent, even though the latter agent was assigned the task in the proposal. For example, in the multisensor tracking domain, an agent might not be aware that it is bidding on a task whose sector overlaps with that of another task. The agent might then bid too optimistically (recall that only one target in a sector can be discriminated by a sensor).

[2] In such a case, the mediator could decide to search for an approximate solution to the original problem.

[3] Recall that a node can only have one sector on at a time.

the contention between tasks t_3^1 and t_3^2, then it can adjust its proposal to the group accordingly to avoid a conflict. However, if some other agent is acting as mediator for t^2, or is not aware of the task contention, then that mediator might very well propose $P2$, the allocation resulting in task contention. In very large domains, with hundreds or even thousands of nodes, it is infeasible to have a single agent that will centrally coordinate all negotiation. Therefore, agent $S5$ might respond to $P2$ with the bid, $\langle[-\infty, 10], t_3^1 - t_3^2\rangle$ indicating the conflict and its source. To resolve this conflict, the mediator may add an option (i.e., enlarge the group of agents) and propose $(S7/1, S8/3, S6/3, S3/2)$.[4] The key point is that the contention cannot be resolved on the basis of local information alone (such as a bid involving a single value); hence the the richer bid format we have proposed.

Experimental results

The experiments were designed to test the hypothesis that agents would achieve better outcomes with dynamic negotiation that allows the negotiation space to be adjusted due to negotiation events (and rich agent bids) than they would using traditional static negotiation methods. The goal of the experiments was to find evidence that an ability to change the issues and options set under negotiation while the negotiation was ongoing allowed agents to more quickly find solutions with high social welfare.

The domain for the experiments was the simplified multisensor tracking domain. Each experiment consisted of four sensor agents and two targets, each with five points in time at which a radar measurement must be taken. The first target has associated tasks $t_1^1, \ldots, t_5^1$; the second target appears some time after the first and has associated tasks $t_1^2, \ldots, t_5^2$.

The appearance of the second target after the negotiation over the tasks of the first target has begun is a domain event that requires adding issues to the negotiation. Specifically, issues corresponding to tasks $t_1^2 \ldots t_5^2$ are added. Dynamic Mediation supports the addition of these issues during negotiation. However, the focus of these experiments is on evaluating dynamic negotiation as it relates to negotiation events — making changes to the negotiation space based on agents' richer bid responses. Therefore, the analysis and data provided in this section is confined to the period beginning with the arrival of the second target and its associated issues.

The goal of the negotiation is to find the best possible task allocation (as measured by social welfare) as soon as possible. The data provided shows the average social welfare attained in 100 different instances of the problem (each differs in the location of agents and targets).

Removing an option

Figure 2 illustrates the negotiated social welfare attained after each iteration of Mediation subsequent to the appearance

[4]In the actual multisensor domain we have described, there are also instances in which the contention can only be resolved if a node switches from one sector to another, halfway between two points on the projected target path, thereby lowering the quality of the measurement for both targets. This complication can be dealt with by partitioning the projected track in a more fine grained way.

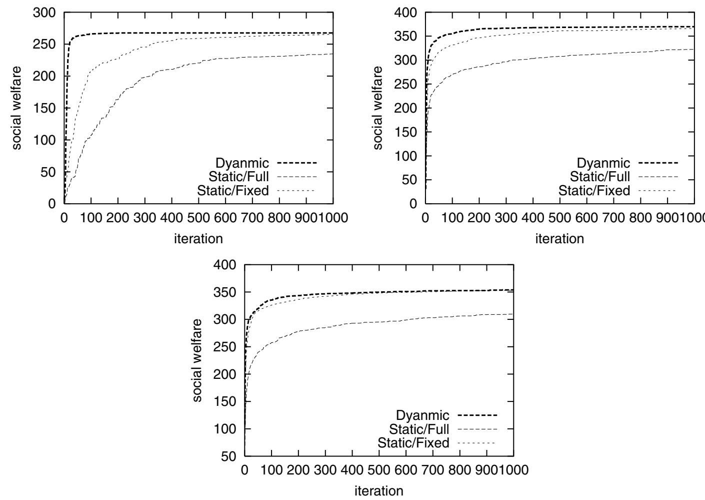

Figure 2: Social welfare after each Mediation iteration for (a) 3 second, (b) 5 second, and (c) 10 second target times.

of the second target. It is assumed here that the there was time for 200 iterations of the hill-climbing Mediation algorithm to assign the first set of tasks $t_1^1, \ldots, t_5^1$ before the second target and the second set of tasks appeared. The following three algorithms for subsequently negotiating the assignment on all ten tasks are compared in the experiments.

Dynamic Leaving the assignments for $t_1^1, \ldots, t_5^1$ unchanged, the mediator proposes successive assignments for tasks $t_1^2, \ldots, t_5^2$. The agents respond to the proposals with rich bids. The mediator uses the rich bids to dynamically adjust the negotiation space during the negotiation by eliminating options for issues in t^2 when they conflict with the assignment for issues in t^2.

For example, consider that t_2^1 was assigned to agent $S1$, and that the mediator made a proposal which assigned $S1$ to t_2^2. If t_2^1 and t_2^2 must be executed at the same time, this proposed assignment is not executable. $S1$'s response for the proposal value would therefore be $-\infty$. With its richer bid, $S1$'s response might be $\langle[-\infty, 6], t_2^1 - t_2^2\rangle$, which indicates that the two tasks conflict and should thus not be assigned to the same agent. The mediator removes $S1$ from the option set of t_2^2.

Static/Fixed Leaving the assignments for $t_1^1, \ldots, t_5^1$ unchanged, the mediator proposes successive assignments for tasks $t_1^2, \ldots, t_5^2$. The agents respond to the proposals with regular bids. The mediator does not change the negotiation space, but continues hill-climbing search for the best assignment for t^2 given the assignment for t^1.

Static/Full When t^2 is introduced, the mediator ignores the values currently assigned to t^1 and proceeds to iteratively find an assignment for all tasks at once.

Each graph illustrates the results of experiments using a different value for the timing (and thus the speed) of the targets, used to manipulate the number of conflicts between the two target paths. For instance, the first graph shows results for a target time of 3 seconds; therefore, each target passes through the sensor area in 3 seconds. As a result, the five tasks associated with tracking that target are more likely to conflict than those in the experiment shown in the third graph, where the target time was 10 seconds.

The results verify that dynamic negotiation outperforms static negotiation. In all tested cases, the social welfare attained by dynamic negotiation is higher than that obtained by static negotiation. The superior performance of dynamic negotiation is more pronounced when the number of conflicts is higher (due to lower target time and higher target speed) and in the early iterations of the Mediation algorithm.

It was expected that dynamic negotiation would be more useful in finding good negotiated outcomes when the target speed was higher: when the target speed is higher, there are more conflicts between tasks because the tasks are more likely to occur simultaneously. Dynamic negotiation makes use of that conflict information revealed by the agents and dynamically adjusts the negotiation space.

It was also expected that Dynamic Mediation's outperformance of static Mediation would be more prominent in early iterations. The value of dynamic negotiation is due to the elimination of unfruitful (conflicting) proposals during the negotiation. In early stages, this elimination allows the search to focus on better outcomes. That, in turn, enables the search to find the most desirable outcomes more quickly. Given more time, Static/Fixed negotiation will catch up: it will eventually find those desirable outcomes that would have already been found by Dynamic Mediation.

Both Dynamic and Static/Fixed methods outperformed Static/Full in our domain. This was not surprising in this domain because the introduction of the second set of tasks does not have a large affect on the best assignment for the first set. The Static/Full method ignored the fairly good assignment that was found by 200 iterations on tasks $t_1^1, \ldots, t_5^1$ before tasks $t_1^2, \ldots, t_5^2$ appeared. The next set of experiments described below provide some insight into the extent to which this superior performance of Dynamic and Static/Fixed results from having a good assignment for t^1 when t^2 appears.

Adding an issue

The second set of experiments explored the value of dynamically adding an issue into the negotiation based on rich bids.

Dynamic/Expand Similar to Dynamic, but instead of removing an option when negative interaction occurs, a previously negotiated issue is added into the negotiation.

For example, consider again t_2^1 which conflicts with t_2^2. If $s1$ was assigned to both tasks, its response might be $\langle[-\infty, 6], t_2^1 - t_2^2\rangle$. Instead of $s1$ being removed as an option for t_2^2, the issue t_1^1 is added into the negotiation, thus expanding the negotiation space.

The results are presented in Figure 3. Both graphs assume a medium intruder speed (time = 5 seconds). Graph (a) shows results for experiments that assume that 200 iterations of mediation occur on the tasks associated with the first target before the second target appears, similar to the assumption made in the first set of experiments. Graph (b) shows experiments run with the contrasting assumption that

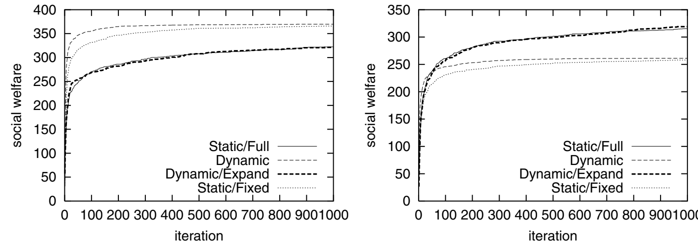

Figure 3: (a) Assumes 200 iterations occur before second target, and (b) assumes one iteration before second target.

there was time for only one iteration before the arrival of t^2. Therefore, the quality of the initial assignment for $t_1^1 \ldots t_5^1$ is considerably higher for the experiment whose results are shown (a) than that whose results are shown (b).

With this higher quality of initial assignment, Dynamic outperforms Dynamic/Expand. This was expected in this domain: reopening negotiation on tasks that had already been assigned to well-suited agents increases the size of the search space, with no significant advantage.

With an initial assignment of low quality (shown in (b)), Dynamic/Expand outperforms Dynamic. Since the quality of the initial assignment is low, significant social welfare improvements result from the expansion of the negotiation to include issues that had previously been "negotiated."

In some domains, the introduction of additional tasks may itself render the quality of the initial assignment for existing tasks poorer. For example, consider a case where there is only one agent capable of performing some newly introduced task that conflicts with a task to which it is already assigned but can be executed by any agent. In that case, even having a large amount of time with which to negotiate an initial assignment does not necessarily mean that little value can be attained by expanding the negotiation.

Related Work

We noted in the introduction the similarity that dynamic negotiation bears to the problem of single agent replanning. In that context, the work of Nebel and Koehler is significant in demonstrating that a conservative modification of a plan can be as computationally expensive as a repetition of the entire planning process (Nebel & Koehler 1993). It would be interesting to investigate dynamic mediation to ascertain whether such results carry over to situations in which task allocations must be formulated in a *time-critical way*.

The MARS transportation scheduler (Fischer *et al.* 1995) addresses the problem of renegotiating multi-agent plans in a dynamic environment (e.g., with traffic jams). Their approach to changes in the environment focuses on replanning *within* an agent and specifies communication with other agents only when a new single-agent plan cannot be found. By contrast, dynamic mediation supports an incremental search for the optimal solution among all interested agents.

In dynamic mediation, the mediator conducts a hill climbing search in which the search space can change in response to external events. The focussed D* algorithm is a real-time replanning algorithm that has been applied to robot path planning in partially known environments(Stentz 1995). In focussed D*, the arc costs in the search graph can change during the search process. As negotiation has been argued to be a form of distributed search (Durfee 1999; Ephrati & Rosenchein 1993), it would be interesting to cast negotiation first as a distributed search problem and then apply an algorithm such as focussed D*. A slightly different perspective is offered by research in self-stabilizing distributed algorithms; those are algorithms that can adapt to transient faults (Dolev 2000). However, work in that area has generally not been concerned with the dynamic reuse of solutions; a self-stabilizing distributed search algorithm designed along lines of previous research would be designed to trigger a *new* search when a fault was encountered.

The constraint satisfaction problem (CSP) is a closely related area of research. On the surface, the structures of the CSP and negotiation problem are analogous: variables and domains in CSP could correspond to issues and options in negotiation. One important difference between the two problems is that interaction among variables in a CSP is limited to constraints whereas issues in negotiation may interact both positively and negatively. Additionally, in a CSP there is no welfare function to be optimized; any variable assignment that satisfies the problem's constraints is a solution. In negotiation, the desirability of different outcomes depends on agents' local utility information.

Research in dynamic CSP has investigated ways in which a solution to a CSP can be adapted to changes in the environment when those changes are expressed as new constraints(Dechter & Dechter 1988; Verfaillie & Schiex 1994). In distributed CSP (DCSP) (Yokoo & Ishida 1999) relevant constraint information is distributed among several agents. DCSP algorithms specify inter-agent communication of constraints (e.g., through *nogood* message). Dynamic Mediation is appropriate for agents in more general environments; it allows communication of positive as well as negative interaction between issues.

Heuristic repair methods have been introduced to CSP problems in the context of dynamic rescheduling (Minton *et al.* 1992); efficient dynamic rescheduling is important for time-critical applications. In these approaches, a combinatorial search problem is solved by first generating a suboptimal solution and then applying local repair techniques. Approaches vary from constructive backtracking in which a partial assignment is incrementally extended to those in which a complete inconsistent assignment to variables is incrementally extended (Minton *et al.* 1992).

Summary and future work

This paper makes a number of contributions: (1) an agent's bid in a negotiation need not be captured by a single value but rather can span a range of values, (2) reasons can be appended to bids that explicate potential positive and negative interactions with other commitments, and (3) ongoing negotiations can be adapted in the context of a dynamically changing environment in which new tasks or faults can appear. The experimental results suggest that dynamic negotiation methods have significant promise. Still, further experiments and a systematic evaluation and quantification of the increased outcome value attained through dynamic negotiation given various domain assumptions is needed. Based on our experiments, we conclude that it is best to use dynamic mediation when time is an important resource and negotiations must end quickly. If negotiation time was unlimited, the quality of outcomes attained by static methods would eventually catch up to those attained by dynamic methods. In addition, dynamic methods are most appropriate when the extra burden placed on agents in making a richer bid is small.

The experiments suggest that dynamically expanding the negotiation (e.g., with adding an issue) is preferred to narrowing the negotiation (e.g., with removing an option) when the values for issues already negotiated are poor. Expansion is also preferred when new issues conflict to a greater degree with the issues already negotiated and may render the values assigned to issues already negotiated as far from ideal.

In future work we plan to explore further the dynamic expansion or narrowing of a negotiation based on bids from agents, domain events, etc. In addition, we plan to explore ways in which mediators can make better use of values provided by agents. Currently, we consider only which issues conflict and do not fully capitalize on the richer information provided by agents' bids in dynamic negotiation.

References

Dechter, R., and Dechter, A. 1988. Belief maintenance in dynamic constraint networks. In *Proc. of AAAI*, 37–42.

Dolev, S. 2000. *Self-Stabilization*. MIT Press.

Durfee, E. H. 1999. *Distributed problem solving and planning*. MIT Press. 121–164.

Ephrati, E., and Rosenchein, J. 1993. Multi-agent planning as a dynamic search for social consensus. In *IJCAI*, 423–429.

Fischer, K.; Muller, J. P.; Pischel, M.; and Schier, D. 1995. A model for cooperative transportation scheduling. In *Proceedings of the First International Conference on Multiagent Systems.*, 109–116.

Minton, S.; Johnston, M. D.; Philips, A. B.; and Laird, P. 1992. Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence* 58:161–205.

Nebel, B., and Koehler, J. 1993. Plan reuse versus plan generation: a theoretical and empirical perspective. In *Proceedings of the IJCAI*.

Rauenbusch, T. 2002. The Mediation algorithm for real time negotiation. In *Proc. of 1st. Intl. Joint Conference on Autonomous Agents and Multiagent Systems*. To appear.

Stentz, A. 1995. The focussed D* algorithm for real-time replanning. In *Proceedings of the IJCAI*.

Verfaillie, G., and Schiex, T. 1994. Solution reuse in dynamic constraint satisfaction problems. In *Proceedings of AAAI*, 307–312.

Yokoo, M., and Ishida, T. 1999. *Multiagent Systems*. MIT Press. chapter 4.