Chapter 10

Negotiating Task Decomposition and Allocation Using Partial Global Planning

Edmund H. Durfee and Victor R. Lesser

Abstract

To coordinate as an effective team, cooperating problem solvers must **negotiate** over their use of local resources, information, and expertise. Sometimes they negotiate to decide which *local* problem-solving tasks to pursue, while at other times they negotiate over the decomposition and distribution of tasks. They might negotiate by sharing all of their information, or by exchanging proposals and counterproposals, or by working through an "arbitrator." In general, negotiation is a complex process of *improving agreement on common viewpoints or plans through the structured exchange of relevant information.* In this paper, we describe how **partial global planning** provides a versatile framework for negotiating in different ways for different reasons, and we examine in detail its utility for negotiating over whether and how problem solvers should decompose and transfer tasks to improve group performance. Finally, we propose how our approach can be extended to capture even more fully the complexity, flexibility, and power of negotiation as a tool for coordinating distributed problem solvers.

10.1 Introduction

A central focus of distributed problem-solving research is coordination—the problem-solving **nodes** in a network should coordinate their use of distributed resources. These resources might be physical (such as computing capacity or communication capabilities) or informational (such as information about the problem(s) being solved or problem-solving expertise). Finding an appropriate technique for coordinating a network depends on the distribution of these resources in the current situation faced by that network, and on the local autonomy of the nodes. Autonomous nodes have their own possibly disparate goals, knowledge, and decisionmaking criteria. Nonetheless, they should still find ways to agree on how to coordinate when coordination could help them achieve their goals better.

Negotiation is the term used in distributed problem-solving research to denote the process by which autonomous nodes coordinate their views of the world and act and interact to achieve their goals. A number of very different techniques with varying behavior, but all embodying aspects of negotiation, have been developed by drawing on the rich diversity in how humans negotiate in different contexts [Conry *et al.* 1986], [Davis and Smith 1983], [Lander and Lesser 1988], [Rosenschein and Genesereth 1987], [Sycara 1988]. This has led to confusion and misunderstanding among researchers who are studying different aspects of the same phenomenon.

For example, the groundbreaking work by Smith and Davis identified **contracting** as a form of negotiation where a node decomposes a large problem into subproblems, announces the subproblems to the network, collects bids from nodes, and awards the subproblems to the most suitable bidders [Davis and Smith 1983, Smith 1980]. In turn, the bidders might subcontract their subproblems. Contracting solves the **connection problem**: nodes match problems to solve with nodes having the resources (expertise, data, computing power) to solve them. However, nodes must often negotiate for other reasons. For example, to solve the **decomposition problem**, nodes should negotiate over decomposing their problems in the first place. Or when subproblems are inherently distributed, nodes must solve the **association problem** by communicating to discover which nodes are working on associated subproblems, and negotiating over how, when, and where to form complete solutions by sharing results. Moreover, negotiation is often an iterative exchange of **counterproposals** leading to **compromise**. Contracting, therefore, is just a rudimentary form of negotiation because it solves only the connection problem using a single round of information exchange.

One general definition of negotiation is: *the process of improving agreement (reducing inconsistency and uncertainty) on common viewpoints or plans through the structured exchange of relevant information.* That is, negotiation leads nodes toward shared plans (where they know of each other's planned actions) or consistent viewpoints (so they are likely to make compatible decisions about local actions). Although they might exchange information of different kinds and in various forms, they begin with some common knowledge about what they might attempt to achieve and how to express themselves. Their negotiation has both protocol and purpose. To negotiate in a wide variety of contexts, nodes need a rich vocabulary, reasoning methods to exploit all the uses of this vocabulary, planning mechanisms to predict and work toward likely future events and interactions, and decisionmaking criteria to choose how to negotiate given the current and possible future situations.

In this paper, we describe how partial global planning embodies a more complete approach to negotiation that allows nodes to solve the connection, decomposition, and association problems using diverse methods such as compromise, appealing to an arbitrator, and trading counterproposals. We use experiments to show how the greater variety of information that nodes exchange, and their ability to plan and predict, lead to more effective cooperation decisions. Finally, we outline future research directions that we hope will lead to even more general techniques for negotiation.

10.2 Partial Global Planning

Partial global planning is a flexible approach to coordination that does not assume any particular distribution of subproblems, expertise, or other resources, but instead lets nodes coordinate in response to the current situation [Durfee 1988, Durfee and Lesser 1987]. Each node can represent and reason about the actions and interactions for groups of nodes and how they affect local activities. These representations are called **partial global plans** (PGPs) because they specify how different *parts* of the network *plan* to achieve more *global* goals. Each node maintains its own set of PGPs that it may use independently and asynchronously to coordinate its activities.

A PGP is a frame-like structure that nodes use as a common representation for exchanging information about their objectives and plans. The PGP's **objective** contains information about *why* the PGP exists, including its eventual goal (the larger solution being formed) and its importance (a priority rating or reasons for pursuing it). Its **plan-activity-map** represents *what* the nodes are doing, including the major plan steps being taken concurrently, their costs and expected results, and why they are being taken in a particular order. Its **solution-construction-graph** contains information about *how* the nodes should interact, including specifications about what partial results to exchange and when to exchange them. Finally, a PGP's **status** contains bookkeeping information, including pointers to relevant information received from other nodes and when it was received. A PGP is thus a general structure for representing coordinated activity in terms of goals, actions, interactions and relationships.

Besides their common PGP representation, nodes also need at least some common knowledge about how and when they should use PGPs to negotiate. This common knowledge is called the **organization**, and is broken into two parts. The **domain-level organization** specifies the general, long-term problem-solving roles and capabilities of the nodes. Given its local goals and plans and the domain-level organization, a node can locally hypothesize potential interactions with other nodes and identify relevant PGP information. It then uses the **metalevel organization**, which indicates the coordination roles of the nodes, to decide where and when to exchange PGPs during negotiation. For example, if organized one way, the nodes might negotiate through a single coordinator that forms and distributes coordinated PGPs for the network, while if organized differently, the nodes might negotiate by broadcasting relevant information and individually forming more complete PGPs.

Given their common representation for PGPs and their shared organizational knowledge, nodes form, exchange, manipulate, and react to PGPs. In some task domains, the set of possible PGPs might be enumerable, so that once the nodes have classified their current situation they invoke the proper PGP. In other task domains, nodes might need to construct PGPs from their local goals, plans, and information. Our partial global planning approach allows a node to encode a local plan in a special PGP called a **node-plan** (because it corresponds to a single node). Guided by the metalevel organization, nodes can then exchange their node-plans and PGPs to build models of each other. A node uses its models of itself and others to identify when nodes have PGPs whose objectives could be part of some larger network objective, called a **partial global goal**, and combines the related PGPs into a single, larger PGP to achieve it. Given the more complete view of group activity represented in the larger PGP, the node can revise the PGP to represent a more coordinated set of group actions and interactions and a more efficient use of network resources. Finally, the node updates its local plans based on this improved view of group problem solving.

10.3 Implementation

We have implemented the partial global planning framework in the Distributed Vehicle Monitoring Testbed (DVMT), which simulates a network of vehicle monitoring nodes that track vehicles moving through an acoustically sensed area [Lesser and Corkill 1983]. The acoustic sensors and problem-solving nodes are geographically distributed, so that each node receives signals from a local subset of sensors. Nodes track vehicles through their own sensed areas and then exchange partial tracks to converge on a complete map of vehicle movements. A node applies signal processing knowledge to correlate its sensor data, eliminating errorful sensor data as it integrates correct data into an answer map. Each node has a blackboard-based problem-solving architecture, with knowledge sources and levels of abstraction appropriate for vehicle monitoring.

Nodes must coordinate to use their resources effectively as they solve the inherently distributed problem of tracking vehicles through the overall area. They must communicate to identify possible overall solutions to work toward, and then decide which subproblems to pursue individually and where to send local results. They should coordinate to avoid duplicating effort in tracking vehicles through overlapping sensed areas and to share partial tracks that might help other nodes resolve uncertainty about their own information. Nodes should consider their local expertise (knowledge sources) when deciding which local subproblems to solve and which subproblems to transfer to more suitable nodes if possible. In short, because subproblems, expertise, and other resources may be inherently but possibly unevenly distributed, nodes must be able to solve the association, decomposition, and connection problems.

The details of how local and partial global planning have been implemented in the DVMT have been given in [Durfee 1988], [Durfee and Lesser 1988b], [Durfee and Lesser 1988a], [Durfee and Lesser 1987], so we will only outline the relevant aspects here. The local planner develops a plan at multiple levels of detail, including a representation of major plan steps. In the DVMT, a major plan step corresponds extending a partial track into a new time frame (such as extending the track derived from data d_i–d_j into d_{j+1}, where d_k is data sensed at time k). This step might take several processing actions to analyze the new data, filter out noise, and integrate the correct data into the track. For each major plan step, the local planner roughly estimates what partial results will be formed and when. By representing and coordinating their major plan steps, nodes cooperate effectively without reasoning about details that are frequently revised and quickly outdated.

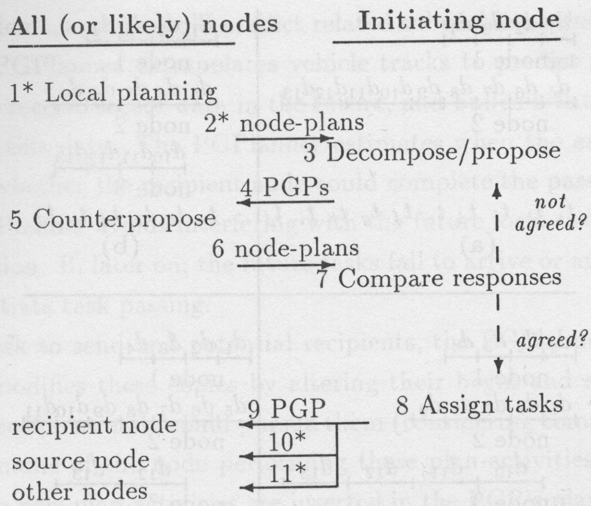
Each node has a partial global planner (PGPlanner) as an integral part of its control activities. The PGPlanner builds a node-plan from each local plan, where a node-plan's objective indicates the possible track(s) being developed and its plan-activity-map is a sequence of plan-activities. Each plan-activity represents a major plan step, and has an expected begin time, end time, and partial result, derived from the local planner's estimates. Guided by the metalevel organization, nodes exchange PGPs and node-plans so that one or more of them develops more encompassing PGPs. When combining PGPs into a single, larger PGP, a node merges the smaller PGP's plan-activity-maps to represent the concurrent activities of all participating nodes, and can reorder the plan-activities to improve coordination. It also builds a solution-construction-graph to indicate which partial tracks formed by the plan-activities should be exchanged to share useful information and construct the complete solution. The PGPlanner then revises local plans based on the PGP. Details of how domain-dependent information and decision-making criteria are encoded and incorporated in the partial global planning framework are given elsewhere [Durfee 1988, Durfee and Lesser 1987].

10.4 Negotiation and Task Passing

PGPs represent *expectations* about how nodes could coordinate their actions and interactions, along with the context (individual objectives, plans, and relationships) that led to those expectations. Negotiation involves exchanging PGP information so that different nodes generate increasingly similar expectations (PGPs). In fact, in stable environments where nodes' plans do not change because of new data, failed actions, or unexpected effects of their actions, nodes can converge on identical PGPs. More generally, however, nodes work in dynamic domains where data, network, and problem-solving characteristics change and communication channels have delay and limited capacity. In these cases, nodes negotiate to *improve* agreement on PGPs; partial global planning allows effective cooperation despite such incomplete agreement.

Partial global planning provides a framework for negotiation that can solve many different coordination problems. For example, when coordinating their pursuit of inherently distributed subproblems, nodes negotiate to solve the association problem by selectively exchanging PGPs to recognize larger network goals and coordinate how they form and share partial results. When nodes have different PGPs (expectations) for working together on the same larger goal, they negotiate by exchanging PGPs to form a compromised PGP that uses the "best" (most up-to-date) information from each of their PGPs. Alternatively, they can negotiate through a third "arbitrator" node (which is assigned that role in the metalevel organization) that forms and distributes a common PGP. The specific mechanisms in the DVMT for this type of negotiation are described elsewhere [Durfee 1988].

In our framework, nodes also can negotiate to solve the decomposition and connection problems by representing proposed problem decompositions and subproblem assignments in PGPs. A node sends such a proposal to possible contracting nodes, which can agree to or reject it, or generate a counterproposal indicating an alternative decomposition, subproblem assignment, or both. By exchanging PGPs, nodes negotiate over both decompositions and contracts. Because a PGP includes information about the larger problem being solved and the participating nodes' activities, a node has more context for accepting, rejecting, or countering a proposal than in a typical contracting protocol, as we now describe in detail.

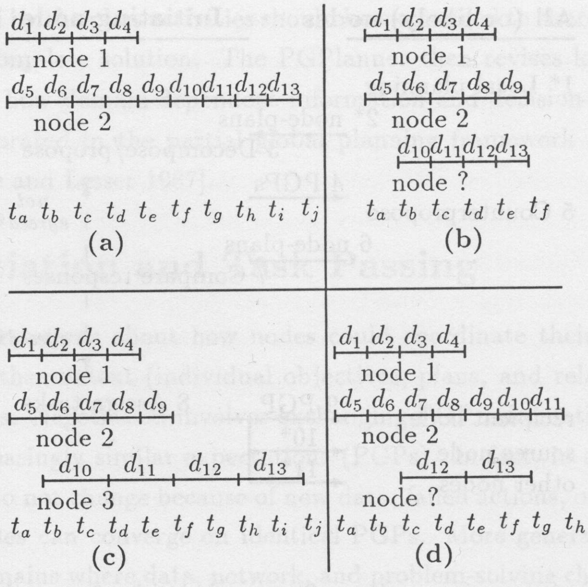


The major steps in task passing are shown. Steps marked with a * are optional.

Figure 10.1: Task passing steps

Before it can transfer tasks, a node must solve the decomposition problem. In some applications, a node might use only local knowledge such as static procedures for how to *always* decompose certain types of tasks. More generally, however, a node cannot intelligently decompose a task without knowledge about other nodes, so that it can decide whether to pursue the task (it might be unimportant relative to other nodes' tasks) and, if so, which nodes might be able to assist it. Before task passing begins, therefore, nodes could share local views. In our framework, nodes form and exchange node-plans so that some nodes develop more global views. These activities are shown graphically in Figure 10.1, steps 1 and 2. Note that these steps are optional, but they help a node make better decisions about decomposing and announcing tasks.

In step 3 (Figure 10.1), a node examines its PGPs, which can represent local activities or activities of several nodes if steps 1 and 2 were taken. The node's PGPlanner checks the solution-construction-graph to detect a *bottleneck* node that expects to complete its partial result much later than other nodes working on the PGP (a node working alone is always a bottleneck). For example, in Figure 10.2a, node 2 is a bottleneck because it expects to finish its activities (to process data d_5–d_{13}) much later than node 1. If it finds a bottleneck node, the node initiates task passing if it is responsible for coordinating the bottleneck node, based on the metalevel organization. Thus, in a network with a central coordinating



Graphic depictions of how a plan-activity-map represents possible task decompositions and assignments. In (a) node 2 is a bottleneck node, and (b) indicates a possible transfer of tasks to an unknown node. Node 3 provides the counterproposal in (c), and this triggers the new proposal in (d).

Figure 10.2: Task passing example

node, that node initiates task passing, while in a broadcast organization where each node has equal responsibility, the bottleneck node itself initiates task passing. Unlike protocols where only nodes with tasks can initiate task passing, our framework permits a node to have an "agent" negotiate for it.

The initiating node's PGPlanner forms a task to pass, where the task is to generate some piece of the bottleneck node's partial result. The task is represented as a sequence of plan-activities. Before deciding on a decomposition, the PGPlanner scans its models of nodes to identify underutilized nodes that might perform this task. A node is underutilized if it participates only in lowly-rated PGPs or is idle, where a node without any local plans can transmit an **idle node-plan** to explicitly indicate its availability. When the PGPlanner finds underutilized nodes, or when the metalevel organization specifies that nodes should attempt to pass tasks despite incomplete network views, then the PGPlanner forms a task to pass from the bottleneck node's activities. For the situation in Figure 10.2a, the PGPlanner decides to reduce the bottleneck by assigning plan-activities for data d_{10}–d_{13} elsewhere (Figure 10.2b).

From its subproblems, a node could predict related subproblems that might arise later. In the DVMT, the PGPlanner extrapolates vehicle tracks to predict whether a possible recipient node might receive sensor data in the future, and builds a **future node-plan** to represent processing this data. The PGPlanner estimates when the earliest future node-plan will begin and whether the recipient node could complete the passed task before this time. If not, the PGPlanner avoids interfering with the future local tasks by removing the node from consideration. If, later on, the future tasks fail to arrive or are unimportant, the PGPlanner can reinitiate task passing.

When it has a task to send and potential recipients, the PGPlanner copies the task's plan-activities and modifies these copies by altering their begin and end times based on estimates of when a recipient node could pursue them (considering communication delays). It also changes the name of the node performing these plan-activities to a special *unassigned* marker. These new plan-activities are inserted in the PGP's plan-activity-map. For example, in Figure 10.2b, an unknown node is expected to start the transferred task at time t_b (because of communication delays). The modified PGP is sent to the potential task recipients or broadcast if specified in the metalevel organization (Figure 10.1 step 4).

When it gets the PGP (Figure 10.1 step 5), a recipient node's PGPlanner extracts the unassigned plan-activities from the plan-activity-map and builds a node-plan from them with this node's name replacing the special marker. The PGPlanner then examines its current set of PGPs and node-plans to determine the earliest that it could begin working on the plan-activities. During this computation, it also uses its own information to form future node-plans since it might have information the initiating node lacks. The PGPlanner modifies the plan-activities to avoid interfering with any actual or expected commitments. It also modifies them if it has or lacks expertise that may affect the time it needs to complete them. In Figure 10.2c, for example, node 3 expects to take twice as much time for each plan-activity because it lacks expertise.

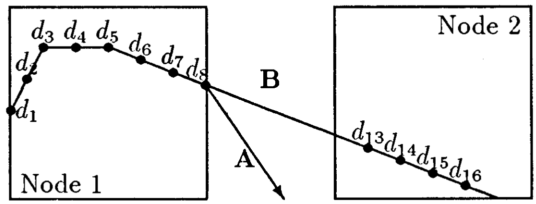
Having modified the new node-plan (if necessary), the recipient node sends it to the initiating node as a counterproposal (Figure 10.1 step 6). As it receives node-plan messages, the initiating node stores them with the PGP. When it has waited long enough (depending on communication delays), the PGPlanner scans the responses (Figure 10.1 step 7) to find which nodes could complete the task earliest, and if any could complete the task sooner than the node currently with the task can, then the PGPlanner decides to transfer the task (possibly to several nodes to increase reliability). Otherwise, the initiating node might give up on passing the task, or it might further negotiate over problem decomposition. To negotiate, the initiating node modifies and sends a PGP proposing that nodes do fewer or different plan-activities (in Figure 10.1, it returns to step 3). In Figure 10.2d, for example, the PGPlanner might propose to transfer a smaller task covering data d_{12}–d_{13} (note that the extra round of negotiation further delays the task's expected starting time). Because PGPs contain information about why and how nodes could cooperate, nodes need not simply accept or reject tasks, but instead can engage in multistage negotiation [Conry *et al.* 1986]. Alternatively, the larger task could be passed, and node 3 could then subcontract out parts of it to other nodes.

Once an assignment has been negotiated, the initiating node updates the PGP to represent the assignment (Figure 10.1 step 8). It sends this PGP to the chosen node(s) (Figure 10.1 step 9), and either sends the task (subproblem to solve) if it has it, or sends the PGP to the **source** node (that has the task) so it will send it (Figure 10.1 step 10). The node that sends the task also keeps a copy in case communication errors, node failures, or poor coordination cause a need to reassign the task.

The initiating node might also send the PGP to unchosen nodes, depending on the metalevel organization. Whether they explicitly receive the PGP (Figure 10.1 step 11), or they learn that they were not chosen because the task does not arrive when expected, the unchosen nodes remove the future node-plan they formed and adjust other future node-plans. For example, if it had responded to several PGPs and had modified plan-activities based on possibly receiving tasks, then once a task is awarded elsewhere the PGPlanner may modify and transmit other future node-plans to indicate that it could pursue tasks earlier. This way, nodes can respond to multiple requests and update their responses when tasks are assigned, although because of communication delays the updated information may not reach an initiating node before the task is assigned and a less than optimal assignment might be made. In a network with communication delays, potentially errorful channels, and asynchronous activities at the different nodes, such incoherence is unavoidable.

10.5 Results

Our experiments focus on simple task passing negotiations; more detailed discussions and experiments with larger networks are provided elsewhere, along with results showing partial global planning's ability to coordinate nodes with inherently distributed subproblems and its overhead costs [Durfee 1988, Durfee and Lesser 1987]. We use two simple two-node environments (Figure 10.3). In the first (A), a vehicle moves through node 1's sensed area and then turns, missing the area sensed by node 2, while in the second (B) the vehicle is detected by node 2. In each case, we simulate data d_i arriving at time i. A knowledge source takes 1 time unit to execute, and the communication delay between nodes is also 1 time unit. These environments explore negotiation for task passing and the role of prediction



Node 1 senses data in both environments, while node 2 only senses data in B.

Figure 10.3: Task passing environments

in deciding whether to accept a task. Environment A is a case where task passing will improve network performance by using node 2's resources in parallel with node 1's, while environment B is a case where prediction allows nodes to avoid passing tasks that the recipient will be unable to complete.

The experimental results are summarized in Table 10.1. Beginning with environment A, experiment E1 uses a noncooperative metalevel organization: node 1 working alone needs 44 time units to generate the solution. In a broadcast organization without prediction (E2), the nodes exchange node-plans (node 2 has an idle node-plan) and each forms its own PGPs. Node 1 passes d_5–d_8 to node 2 early on and they cooperatively form the overall solution sooner than in E1. A centralized organization with node 2 as coordinator (E3) is even faster since node 2's proposed task passing is to itself and it accepts the proposal without delay. When nodes can build predictions, they expect node 2 to get its own tasks. With the broadcast organization (E4), nodes delay task passing until after the expected tasks fail to arrive, at which point they only pass d_6–d_8. The delay causes them to form the overall solution later, and the same thing occurs in the centralized organization (E5). Thus, nodes can recover from incorrect predictions, but they may degrade performance.

With environment B, however, the predictions are correct. When noncooperative (E6), the nodes individually form their own solutions. In a broadcast organization without prediction (E7), node 1 sends d_5–d_8 to node 2 because it expects node 2 to be available. Node 2 ignores the task until it forms its own result, and then only works on d_7–d_8 (node 1 kept copies of d_5–d_6 which it completes locally). Thus, node 2 does help node 1, but failure to predict future tasks causes unnecessary communication overhead from exchanging too much information. In the centralized organization (E8), the task d_5–d_8 passed by node 1 is adopted earlier by node 2 (because node 2 proposes and accepts at the same time), and node 2 prefers it to its own task. Consequently, the nodes form d_1–d_8 together first as if node 2 had no tasks of its own (as in E3), and then they pass tasks and work together on d_{13}–d_{16}. This cooperation on both tasks involves substantial communication.

Table 10.1: Experiment Summary

Exp	Env	MLO	Pred	Pass	Done	STime
E1	A	loc	-	-	-	44
E2	A	bc	none	d_5-d_8(12)	d_5-d_8	33
E3	A	cn(2)	none	d_5-d_8(11)	d_5-d_8	32
E4	A	bc	all	d_6-d_8(19)	d_6-d_8	35
E5	A	cn(2)	all	d_6-d_8(18)	d_6-d_8	34
E6	B	loc	-	-	-	44/32
E7	B	bc	none	d_5-d_8(12)	d_7-d_8	39/32
E8	B	cn(2)	none	d_5-d_8(11) d_{15}-d_{16}(27)	d_5-d_8 d_{15}-d_{16}	32/42
E9	B	bc	all	d_8(35)	d_8	42/32
E10	B	cn(2)	all	d_8(34)	d_8	42/32
E11	B	cn(1)	local	d_8(35)	d_8	42/32

Abbreviations

Env: The problem-solving environment
MLO: Metalevel organization: loc=local, bc=broadcast, cn(n)=centralized with node n coordinator.
Pred: Nodes a node can predict future tasks for.
Pass: Tasks passed in network (and time passed).
Done: Passed tasks that are actually by recipient.
STime: The time to find solution(s); if more than one, earliest time for each is given (d_1-d_8/d_{13}-d_{16}).

When they can predict future tasks, nodes avoid premature task passing. In both a broadcast (E9) and centralized (E10) organization, they recognize that they will both be busy, and do not pass tasks until after node 2 completes its result, at which time node 1 sends d_8 to node 2. By waiting until after node 2's local tasks are complete, they incur much less communication overhead because they send only what node 2 can process in time. However, by waiting until node 2 is completely done before negotiating, they waste time transferring tasks while node 2 is idle. That is why E7 found the solution sooner—it had received (too many) tasks ahead of time. We hope to improve our mechanisms to negotiate over passed tasks *as* a node is completing its local tasks. We also plan on studying the tradeoffs in making and reacting to predictions. Specifically, we want to develop mechanisms that will decide how and when to make predictions given knowledge about the current situation, including the probability of making correct predictions and the costs of missed or incorrect predictions.

Finally, in E11 we minimize overhead spent in reasoning about others by allowing nodes to hypothesize only their own future node-plans. We use environment B with a centralized organization, but where node 1 is the coordinator. Unable to predict node 2's future plan, node 1 sends a PGP proposing task passing to node 2 early on. Node 2 uses that PGP to hypothesize its future node-plan, and responds to node 1 that it could not perform the task until much later. Thus, nodes negotiate despite incomplete local views to arrive at appropriate task passing decisions, and use information in the PGP that is not transmitted in typical contracting protocols.

10.6 Discussion

Partial global planning provides a more general framework for negotiation in distributed problem-solving networks than contracting because nodes can communicate more information (encoded in PGPs), can structure their coordination activities (encoded in the metalevel organization), can plan for and predict possible future events that affect negotiated decisions, and can use these mechanisms flexibly to negotiate over whatever coordination problems (association, connection, decomposition) they face in their current situation. Through implementing and evaluating our framework in the DVMT, we have examined its costs and benefits, and shown how it integrates planning, prediction, and negotiation. Moreover, although the low-level representations of objectives and actions and the criteria for ordering them are domain-dependent, the high-level PGP structures and the methods for forming, exchanging, manipulating, and revising them provide a generic foundation that should be applicable in a variety of domains such as air-traffic control, job scheduling, and multiple-robot environments.

Ours is not a completely general framework for negotiation, however, because of one important limitation. Although it flexibly allows nodes to negotiate over their plans at a specific level of detail, it does not allow negotiation at other levels. For example, to supply the context needed to agree on a plan, nodes might need to exchange more detailed information about their current situation, such as what data they are working with. Or nodes might need to negotiate at a higher level, because although they might have common views of each other's goals and plans, they might have different views on how to compare, rank, or combine this information. That is, nodes might need to negotiate not only over what they should do (such as what actions to take), but also over the criteria to use when deciding what they should do (such as why some actions are preferable to others). And then given the ability to negotiate at several levels, how should nodes decide at what level to negotiate at any given time?

Clearly, many questions remain regarding negotiation in distributed problem-solving networks, and much work needs to be done. We are optimistic that these questions will eventually be answered, however, based on our experience with partial global planning. There, by moving to a new representation for plans and interactions, we discovered that negotiation for sharing partial results and for assigning tasks, which have traditionally been treated separately, are really just two sides of the same coin. We hope that applying partial global planning in other domains will help us to develop similar insights that will lead to an even more complete framework for negotiation.

Acknowledgements

This research was sponsored, in part, by the National Science Foundation under CER Grant DCR-8500332, and by the Office of Naval Research under University Research Initiative Grant Contract N00014-86-K-0764, and under Contract N00014-79-C-0439.

References

[Conry *et al.* 1986] S. E. Conry, R. A. Meyer, and V. R. Lesser, "Multistage Negotiation in Distributed Planning," in A. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann Publishers, 1988.

[Davis and Smith 1983] R. Davis and R. G. Smith, "Negotiation As a Metaphor for Distributed Problem Solving," *Artificial Intelligence*, Vol. 20, 1983, pp. 63–109.

[Durfee 1988] E. H. Durfee, *Coordination of Distributed Problem Solvers*, Kluwer Academic Publishers, 1988.

[Durfee and Lesser 1988b] E. H. Durfee and V. R. Lesser, "Incremental Planning to Control a Time-Constrained, Blackboard-Based Problem Solver," *IEEE Transactions on Aerospace and Electronics Systems*, Vol. 24, No. 5, September 1988, pp. 647–662.

[Durfee and Lesser 1988a] E. H. Durfee and V. R. Lesser, "Predictability Versus Responsiveness: Coordinating Problem Solvers in Dynamic Domains," *Proceedings of the Seventh National Conference on Artificial Intelligence*, August 1988.

[Durfee and Lesser 1987] E. H. Durfee and V. R. Lesser, "Using Partial Global Plans to Coordinate Distributed Problem Solvers," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, August 1987, pp. 875–883.

[Lander and Lesser 1988] S. Lander and V. R. Lesser, "Negotiation among Cooperating Experts," *Proceedings of the 1988 Workshop on Distributed Artificial Intelligence*, May 1988.

[Lesser and Corkill 1983] V. R. Lesser and D. D. Corkill, The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks," *AI Magazine*, Vol. 4, No. 3, Fall 1983, pp. 15–33.

[Rosenschein and Genesereth 1987] J. S. Rosenschein and M. R. Genesereth, "Communication and Cooperation among Logic-Based Agents," *Proceedings of the Sixth Phoenix Conference on Computers and Communications*, February 1987, pp. 594–600.

[Smith 1980] R. G. Smith, "The Contract-Net Protocol: High-level Communication and Control in a Distributed Problem Solver," *IEEE Transactions on Computers*, Vol. C-29, No. 12, December 1980, pp. 1104–1113.

[Sycara 1988] K. Sycara, "Resolving Goal Conficts Via Negotiation," *Proceedings of the Seventh National Conference on Artificial Intelligence*, August 1988, pp. 245–250.

Edmund H. Durfee
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109

Victor R. Lesser
Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003