Negotiation: Opportunities and Pitfalls
Experience and Observations

by

Susan E. Conry

The term "negotiation" has been used in the distributed problem solving / multiagent systems community for many years. The first use of the term probably arose in the context of the Contract Net protocol in the early 80's, but more recently, the term negotiation has been applied to a variety of techniques. This paper takes as a point of departure the definition that was given in the call for participation in IWMAS'98 – "negotiation is interactive communication for arranging activity".

This abstract addresses opportunities and pitfalls of negotiation as viewed from the perspective of our experience with experimental work in multiagent systems. Four different problem solving strategies, each of which uses negotiation (viewed as indicated above) in multiagent problem solving has been experimentally investigated. Three of the strategies discussed here involve coarse grained agents, each of which has the capacity to reason in a sophisticated manner about its activities. The third reflects organization of agents whose granularity is at the other end of the spectrum. The sharp contrast among the issues that are delineated in each of these case studies indicates the diversity of opportunity and the presence of pitfalls in devising negotiation strategies for multiagent systems.

Multistage Negotiation

Our earliest work in distributed problem solving involved the development (with Victor Lesser and Kazuhiro Kuwabara [1]) of a protocol for negotiating allocation of resources. Though the protocol itself is domain independent, the example domain of application was concerned with allocation of link capacity and channel bandwidth in management of a large communication system. This domain of application is a very rich one that has many interesting features. In particular, the system may have multiple goals concurrently active and it is not necessary for each agent to be aware of all goals.

Satisfaction of system goals requires that available bandwidth and link capacity be allocated to affected users so that service is restored. Once allocated, these resources cannot be shared, so the scarce resource, and the source of conflict, is available bandwidth and channel capacity. Resources fall into two broad categories: those that are controlled by a single agent and those whose control is shared. Satisfaction of each system goal therefore requires a coordinated set of actions distributed over a subset of the agents in the system. In general, satisfaction of one goal may preclude satisfaction of other goals in the system. What is involved in problem solving is a complex distributed search in which incompatibilities among solutions to interacting subproblems may not be directly observable at a given agent, but can only be recognized as a more global view of the entire problem is constructed.

The protocol which we devised involves an incremental "tentative commitment" of local resource followed by an exchange of information that allows agents to acquire knowledge about the nonlocal impact of their local decisions. In those cases where severe nonlocal impact is projected, agents may retract these tentative commitments and attempt to assess the impact of another local alternative. It is important to observe two things about our protocol: agents never build a complete and accurate global view, and in the worst case, where the overall problem is overconstrained, the protocol must resort to what amounts to an exhaustive search. (Though an overconstrained problem may be detected earlier in some cases.) The knowledge about nonlocal impact is aggregated and exchanged among agents at a higher level -- not at a low enough level to allow agents to acquire adequate detail to solve the problem directly.

The protocol is a three phase protocol The first phase is an asynchronous search phase that is directed towards management of the distributed search for satisfaction of distinct goals that are instantiated independently by different agents in the system. If there is a great deal of conflict among subgoals (as there will be when the overall problem becomes more and more constrained by the availability of resources), it may be necessary for an agent to resort to activity directed towards acquiring all the information that it can about the nonlocal impact of satisfying a particular goal. This requires coordination with other agents and is accomplished as the agent participates in a coordinated search phase. Finally, if the coordinated search phase reveals that the problem is indeed overconstrained, as will be the case when some subset of the goals is mutually infeasible, the protocol enters an overconstrained resolution phase in which some goals are abandoned in order to satisfy others.

Multistage negotiation offers an opportunity for agents to reason about the nature of inter-goal competition for scarce resource. In this system, goal conflict is made explicit. Agents engage in a dynamic detection of the nonlocal impact of their local decisions. The process results in a situation in which agents come to recognize that subset of other agents involved in satisfying a particular goal. This protocol thus allows a system of agents to reason explicitly to determine which constraints should be relaxed in overconstrained situations.

We can get a measure of the degree to which solutions compete by observing the fraction of the overall problem solving time that is spent in asynchronous search versus the amount of time spent in coordinated search. In our experiments, scenarios involving a high degree of goal interaction resulted in a large fraction of the overall time (and a large fraction of the inter-agent communication load) in the coordinated search phase. Those scenarios in which there was less interaction, or when the goal interaction was localized so that conflicts occurred largely within the purview of a single agent, the effort was much more balanced between the asynchronous search phase and the coordinated search phase. Thus we have opportunity: the capacity to reason about inter-agent goal interactions in an iterative fashion. We also have a pitfall: when the interactions are complex, resolving them may dominate activity.

Distributed Automated Reasoning

In our multiagent automated reasoning work [2, 3] we investigated the behavior of two systems in which multiple theorem proving agents act to cooperatively prove a theorem. In each of these systems, it is assumed that axioms are distributed to agents in the system. The assumption in each case was that no one agent has enough knowledge to prove the theorem by itself, and agents may exchange information in an attempt to solve the problem of establishing the theorem. In one of these systems, we modeled overlap in agent knowledge by permitting axioms to be distributed to more than one agent. Thus redundancy was allowed. In the other system, no redundancy was allowed.

In both of these systems, we utilized a saturation level theorem proving algorithm that used tautology and subsumption based deletion strategies to improve performance. The theorem proving loop was modified in two ways: performance was monitored to ascertain whether any new resolvents were being generated at each saturation level. At this point, a "forward progress" heuristic was applied to judge whether or not effective progress towards a solution was being made. If it appeared that no significant progress was being made, a request for information was formulated and transmitted to other agents in the system. This request had a strong unit preference bias, and resulted in return of knowledge from other agents that was judged to be relevant to the requesting agent. The primary differences in the systems lay in the underlying inference strategy that was employed. In one case (DARES), we used binary resolution, and in the other (SHYRLI), hyper-resolution was employed.

Each of these systems was the object of extensive experimentation, the goal of which was to observe the pattern of problem solving activity and the total amount of work expended in problem solving. In each case, the form of negotiation involved requires communication of partial results among agents so that forward progress towards a solution can be facilitated.

In DARES an axiom could be distributed to more than one agent, so redundancy was allowed in system knowledge. One would expect that this increased redundancy would result in less need for negotiation. Agents would have a greater percentage of the knowledge required for task performance and would therefore spend a greater fraction of their time doing problem solving relative to "their own" tasks. Thus one might expect that increased redundancy would result in performance enhancement. In fact this did not happen in DARES. Increased redundancy leads to greater local activity, but it is less focused. The agents are each working on a larger part of the overall problem and are therefore less effective in their effort.

In SHYRLI, on the other hand, no redundancy was allowed. Indeed, one feature of distributed hyper-resolution is that agents may preserve the privacy of some part of their knowledge. Thus it is possible for the system as a whole to solve a problem without having all knowledge becoming common knowledge (in the worst case). Agents wishing to maintain the privacy of some part of their knowledge must participate in any problem solving that impinges on that knowledge. This makes the amount of coordination required more extensive and may limit the performance enhancement that can be achieved. Experimental data indicate that agents spend a significant amount of time waiting for other agents to provide solutions to subproblems that may be of use to them.

Thus experimental data leads us to the conclusion that narrowing the focus of an agent's attention leads to greater need for coordination, and this can serve to improve problem solving behavior. When agents are "specialists", each focuses attention on its own subproblem, sharing results only when it is needed. The overall system arrives at a solution much more quickly. The reason for this is that each agent has a local search space that is initially determined by its set of local axioms. If there is significant redundancy in the system, there is substantial overlap in the search spaces of agents. Thus each agent is itself solving a larger problem, and the combinatorics of the problem explain the reduced performance. The lesson to be learned from these experiments is that it may pay to increase the need for inter- agent coordination. Herein lies an opportunity. It comes with a pitfall – there is a high degree of sensitivity to the distribution of knowledge. As has been mentioned, SHYRLI agents may keep some knowledge as "private" or "secure" knowledge. When an agent has a particular piece of knowledge that is required for solving multiple problems, that agent is forced to participate in solving each of these problems. Problem solving performance suffers because there is an increased need to explicitly coordinate problem solving activity, not simply share results, and this imposes more sequential ordering on the problem solving process. The patterns of knowledge distribution had little impact on aggregate system performance in DARES, but had significant impact on SHYRLI's performance for this reason.

Distributed Constraint Satisfaction

In our experiments with DARES and SHYRLI, we distributed knowledge among agents in the system in a random manner. The knowledge resident at a problem solving agent was monotonic in the sense that no local knowledge was ever deleted. Once an axiom or a clause was known to an agent, it was always known by that agent. There was no adaptive shifting of knowledge among agents. Thus knowledge was not viewed as a resource in the same sense as link capacity or computation capacity or any other non-sharable resource. Our work on generalized distributed constraint satisfaction [4] resulted in development of a distributed constraint satisfaction system (DCONSA) in which we can investigate the ways in which knowledge is propagated during problem solving. This enables investigation of ways in which management of allocation of knowledge among agents has impact on problem solving behavior.

DCONSA is based on the GEM formalism. A problem is specified to DCONSA as a set of groups and elements. These groups and elements represent regions of activity that may overlap, may be mutually disjoint, and may be hierarchically nested. Each region of activity has associated with it a set of constraints that must be satisfied by events in that region of activity. System goals and task decomposition knowledge are represented as constraints. Any unsatisfied constraint in a region may be satisfied through application of a "resolver" that may require addition of events and/or temporal relations among existing events in a region.

Initially, group and element definitions are distributed among a group of agents in DCONSA. Only one instance of each region's definition is present in the overall system at any point in time. Problem solving is a process that proceeds until the set of activities in each region satisfies all applicable constraints. Within each agent, an attempt is made to satisfy any unsatisfied constraints about which the agent has knowledge. When an agent does not have knowledge about a region that is involved in satisfying some constraint, an interaction (negotiation) with the agent who does have the requisite knowledge is needed.

In DCONSA, negotiation is required to solve problems related to the locus of knowledge during problem solving. A particular piece of knowledge is initially only resident at one agent in the system. Thus a situation can arise in which an agent does not have the knowledge it needs, and that knowledge is resident at another agent. There are two actions that could be taken when this occurs: the knowledge itself could be transferred or a copy of the knowledge could be transmitted. In DCONSA, we adopted the first strategy: there is only one copy of any particular piece of knowledge (region definition) in the system. The primary reason for this decision lies in the observation that if replication of knowledge is allowed in this kind of system, there is a need to ensure that the beliefs of different agents having the same piece of knowledge are at least compatible. This adds another form of required inter-agent coordination, which we intend to investigate further.

Agents in DCONSA have a degree of autonomy. Issues related to allocation of knowledge to agents are focused on goal satisfaction and are managed by the agents as they exercise this autonomy in determining the form of their participation in problem solving. When an interaction is requested of an agent, that agent may elect to interact in one of three modes:

1) the agent may choose to assume responsibility for satisfying the goal that is the focus of the interaction,

2) the agent may choose to do whatever it can locally to further goal satisfaction, but not assume responsibility for satisfaction of this goal,

3) the agent may decide that it does not wish to get involved, and send all of its local knowledge related to the goal to the agent requesting the interaction.

In our experimental work, agents have used a heuristic that is related to an assessment of how busy they are and how much work they project will be involved in doing the requested task in deciding upon an interaction mode.

Our experimental work with DCONSA has been focused on assessing the effectiveness of this interaction strategy. The data indicate that by allowing agents to autonomously select their interaction mode, effective load balancing is achieved. In planning scenarios, planning effort is reasonably evenly distributed among agents over time, and the overall agent utilization is improved. The data also indicate that when agents are allowed to choose their own mode of interaction, knowledge appears to migrate to those loci where it can be most effectively used. Communication overhead is reduced and overall system performance is improved. It is not surprising that this phenomenon is particularly evident when system goals are relatively decoupled from one another.

Fine-Grained, Net-Based Multiagent Systems

Virtually all of the existing work on multiagent systems makes the assumption that agents are sophisticated, that they are able to interact in complex ways, and that the problem solving that is performed by each agent is coarse grained. We believe it is important to gain an understanding of coordination strategies and their effectiveness across a spectrum of agent granularities. In pursuit of this goal, we have devised an agent architecture and system structures to support fine-grained multiagent problem solving over the internet [5-8]. In our model, each agent resides at a workstation on the internet and is capable of coordinating its activities with a variable number of other agents using the same coordination protocol.

We have done experimental work involving two sets of assumptions concerning system structure. In one [6, 7], the system of agents has a virtual topology that is structured as a loosely coupled network of domains, each of which contains some number of nodes in the overall system. Activity in these domains is coordinated via *gateway* agents. This structure serves to limit the amount of inter-agent message traffic and the amount of global knowledge that an agent must have. In the other [8], the virtual system structure is more hierarchical, with a tree-like structure involving multiple levels of control responsibility. In either case, agents are resident at hosts on the internet. Thus hosts are heterogeneous, participation in the problem solving network is voluntary, and nodes can join and disengage from the network at any time, in an asynchronous fashion. Thus the underlying task management infrastructure must be scalable, fault tolerant, and platform independent.

Our agents perform problem solving tasks using evolutionary algorithms over the internet. The coordination protocols involved in managing problem solving are adaptations of the contract net protocol, though the messages generated are slightly different in the context of a hierarchical virtual structure than they are in the context of a loosely coupled virtual structure.

Our experimental work has generated data for a wide variety of problem solving situations. Within the context of the loosely coupled network, our experiments involved solution of four standard genetic programming problems. We allowed the system to tune its own set of genetic parameters, so that the network of problem solvers could adaptively tune its own performance. From this set of experiments, we learned that with a network of N agents we could achieve slightly less than a speedup of N. We also learned that the sets of genetic parameters that evolved during problem solving differed substantially from the "standard" set that has been used in the literature. In at least one situation chosen from a colleague's engineering design application, we were able to achieve better results with better convergence rates than had been achieved using a conventional genetic programming approach.

Our observations concerning this set of experiments led us to ask how much of the problem solving overhead is due to the agents' efforts to construct a reasonably accurate view of the network. We then devised a virtual system structure which is hierarchical and generalized the agents so that we could handle a variety of evolutionary algorithms (not just genetic programming).

Experiments with this organization have involved thirty different problems including 11 numerical optimization problems, 13 combinatorial optimization problems, and 6 machine learning problems. Each of these problems has been solved using eleven different system configurations so far, with characteristics ranging from single agent execution to a "flat" structure involving a single domain controller and many hosts to a "vertical" structure involving many domain controllers, each of which has few hosts. These experiments have been performed over a network of processors that is not dedicated to this task, but which may have other users doing other work at workstations concurrently with our tasks.

Our results are encouraging. The data is voluminous and we are still in the process of generating data for other configurations so that we can characterize the shape of the performance curves. In general, we are observing that superlinear speedup is achieved for approximately 95% of the problems which took longer than 20 seconds to run in a single agent environment. Optimization of parameters appears to improve performance in the vast majority of the cases, but there are some in which standard settings seem to be better. From this set of experiments, we can draw two tentative conclusions: 1) adaptation of the contract net coordination protocol is effective in fine grained systems and 2) the cost of maintaining an relatively accurate view of system state may be more than is reasonable (better performance can be obtained by imposing some structure on the virtual system topology).

Concluding Remarks

Our base of experimental work in multiagent systems has spanned a variety of systems and problem solving strategies. From this base of experience, we can begin to identify a number of opportunities for negotiation (broadly construed) in multiagent systems. Among these are:

- The ability to reason about nonlocal impact of local decisions provides a vehicle for coherent problem solving in the absence of complete local knowledge
- Narrowing the focus of an agent's activity can increase the need for coordination but can also improve system performance, so it may pay to increase the need for negotiation

- Allowing agents to autonomously choose their interaction mode based in inter-agent communication can be used to effectively achieve distributed load balancing in a system
- Use of coordination strategies that have been used with coarse grained agents doing sophisticated reasoning may provide effective and efficient problem solving in fine grained systems as well

There are also a number of pitfalls that become evident when we examine this base of experience.

- If the interactions among goals in a multiagent system are complex and extensive, resolving them can dominate problem solving and be very expensive
- There may be substantial sensitivity to the initial distribution of knowledge among agents in the system
- In situations involving migration of knowledge among agents, performance may be degraded due to the cost of knowledge transmission among agents. This is true in cases where the initial distribution is suboptimal and also in situations in which there is a high degree of goal coupling
- Management of fault tolerance, maintenance of local security, and construction of knowledge about the state of a problem solving network can be costly

Our work leads us to believe that the pitfalls can be overcome, but understanding the underlying phenomenon is required to do so, and there is not yet a broad base of experimental work to support this understanding across a spectrum of problem solving situations.

References

1. S.E. Conry, K. Kuwabara, V.R. Lesser, and R.A. Meyer, "Multistage Negotiation for Distributed Constraint Satisfaction", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 6, (1991) pp. 1462-1477.

2. D.J. MacIntosh, S.E. Conry, and R.A. Meyer, "Distributed Automated Reasoning: Issues in Coordination, Cooperation, and Performance", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 6, (1991) pp. 1307-1316.

3. W.L. Maguire, "The Application of Hyper-Resolution to Distributed Automated Reasoning", M.S. Thesis, Department of Electrical and Computer Engineering, Clarkson University, Potsdam, NY 13699 (May 1992).

4. R.P.Pope, "DCONSA: A Distributed Constraint-Based Planner for Semi-Autonomous Agents", Ph.D. Dissertation, Department of Electrical and Computer Engineering, Clarkson University, Potsdam, NY 13699 (August 1992).

5. P. Nangsue, P. Pillay, and S.E. Conry, “Evolutionary Algorithms for Induction Motor Parameter Determination”, to appear in the *IEEE Transactions on Power Systems*

.

6. P. Nangsue and S.E. Conry, “Internet-Based Genetic Programming Platform”, *Late Breaking Papers at the Genetic Programming 1997 Conference,* Stanford Bookstore, Stanford University.

7. P. Nangsue and S.E. Conry, “Fine-Grained Multiagent Systems for the Internet”, to appear in *the Proceedings of the Third International Conference on Multiagent Systems*, Paris France, 1998.

8. P. Nangsue and S.E. Conry, “An Agent-Oriented, Massively Distributed Parallelization Model of Evolutionary Algorithms”, submitted to Genetic Programming 1998.