Towards Large-scale Conflict Resolution: Initial Results

Hyuckchul Jung, Milind Tambe, Weixiong Zhang, Wei-min Shen
University of Southern California/Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292
{jungh,tambe,zhang,shen}@isi.edu

Abstract

With the increasing interest in distributed and collaborative multi-agent applications, conflict resolution in large-scale systems becomes an important problem. Our approach to collaborative conflict resolution is based on argumentation. To understand the feasibility and the scope of the approach, we first implemented the process in a system called CONSA and applied it to two complex, dynamic domains. We then modeled this approach in distributed constraint satisfaction problems (DCSP) to investigate the effect of different conflict resolution configurations, such as the degree of shared responsibility and unshared information, and their effects in large-scale conflict resolution via argumentation. Our results suggest some interesting correlations between these configurations and the performance of conflict resolution.

1 Introduction

Distributed, collaborative agents are promising to play an important role in large-scale multi-agent applications. While there is now considerable progress in such collaborative agents [2], the issue of collaborative conflict resolution has remained largely unaddressed, particularly in large scale system. For collaborative conflict resolution, we have developed a system called CONSA (COllaborative Negotiation System based on Argumentation)[3]. CONSA is based on argumentation[1] and has been applied to two complex, dynamic domains: battlefield simulations and distributed sensor systems.

CONSA's practical applications revealed different conflict resolution configurations. One key parameter of such configuration is *responsibility* of agents in conflict resolution. The second key parameter is the degree of unshared information available only to local agents. To investigate the implications of different configurations in large scale, we have modeled the CONSA approach using distributed constraint satisfaction problems (DCSP)[4]. DCSP provides a formal framework to study argumentation with various parameters and understand the impact of configurations on scalability. Our results show interesting correlation between the configurations and the conflict resolution performance.

2 Background and Motivations

In this section, we briefly describe the negotiation process of CONSA via argumentation and illustrate the types of conflict resolution which CONSA addresses.

CONSA's argumentation involves an agent (sender) making a proposal to the agent-team (receivers) with an attached justification (argument). The receivers evaluate the proposal by taking the justification into account, and either accept or refute it. If refuting the proposal, a receiver may send back a counter-proposal to the team, who may continue this cycle of proposals and counter-proposals.

In our helicopter combat simulation domain[2], CONSA is applied in allocating *firing positions* for a team of pilots. Each firing position must be at least one kilometer apart from others. To locate firing positions, agents must negotiate, rather than relying on a centralized pre-planner because each pilot may have different information of enemy, friendly vehicles and own state.

In CONSA's distributed sensor system, multiple sensor agents are required to work together to track events (with deadlines) in their environment. Each agent has its own specific capabilities, and multiple agents with similar capabilities are available. With multiple targets appearing at different intervals, and with each agent possessing its own prior commitments of what to track and when, conflicts arise over the agents' limited resources. Once again, sensor agents must negotiate to ensure targets are cooperatively tracked before deadlines.

3 Formalizing Argumentation as DCSP

Our two applications of CONSA illustrate two different conflict resolution configurations. One key parameter of such configuration is *responsibility* for conflict resolution. If this parameter indicates *fully shared responsibility*, each agent assumes shared responsibility for resource allocation to all of the agents. For instance, in the firing position example, each agent assumes shared responsibility for firing position allocation to all of the agents. In contrast, with *non-shared responsibility*, each agent allocates non-shared portion of the resources. For instance, in distributed sensor application, each agent allocates its own resources to a subtask without considering resource allocation of the other agents. Fully shared responsibility is likely to generate an acceptable solution but requires complex local computation. In contrast, non-shared responsibility has a simpler local problem but tends to generate incompatible solution.

Another key parameter of the configuration is the degree of unshared information available only to local agents. In the firing position example, different agents may have information about the different enemy positions. Thus, with no information sharing, each agent has local information which is not available to the other agents.

To understand the computational efficiency and the impact of competing configurations particularly on larger-scale systems, we need to abstract argumentation into a formal well-understood framework. To this end, we have modeled CONSA using DCSP with multiple local variables per agent[4]. To formalize the argumentation process as a DCSP, variables, domains, and constraints should be defined. In addition to this problem, there is another issue about how to represent arguments in this framework. The key here is that each agent can divide its local variables into two categories: *public* and *private*. The relationships among public and private variables constitute the constraints in DCSP and, their values and domains are known to other agents. On the other hand, the private variables influence the value assignment of the public variables but their values need not to be known to other agents beforehand. Given these definitions, the problem of argumentation can be defined as a DCSP with public and private variables.

In the firing position example, the helicopter's firing positions are the public variables. The enemy's positions are private because they influence the firing positions, but they are not in conflict and may not be known to other agents.

4 Effects of Argumentation

Based on formalization of argumentation into DCSP, we experimentally study argumentation, and evaluate the effects of responsibility and information sharing on the quality of argumentation. For this experiment motivated by firing position example, we used Yokoo's distributed constraint satisfaction algorithm from [4].

Considering the group structure and responsibility sharing, we set up three realistic configurations: (i) full responsibility sharing (FRS), (ii) partial responsibility sharing (PRS), and (iii) no responsibility sharing (NRS). In PRS, agents are grouped into four partitions where each partition is a FRS . In our experiments, each agent has one private variable which simulates the position of enemies.

The performance of negotiation is measured by the total number of negotiation cycles, the total number of exchanged messages , the total number of constraints checks, and the number of constraints checks per negotiation cycle.

Figure 1 shows the results where one thirds of the agents have unshared private variables(i.e. the 1/3 of pilots are aware of extra enemy positions). The results indicate that tradeoff exists in the number of cycles and the number of constraint checks per cycle for NRS, FRS, and PRS. In particular, NRS has more cycles but low local computation which is opposite to FRS. The PRS provides a good compromise among all these factors.

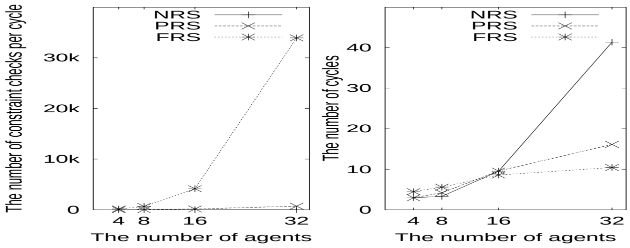

Figure 1. configuration and performance.

In the above experiments, the private variables constrained the public variables to interval values. To further investigate the impact of private variables on the performance of negotiation, we have also tightened the constraints of private variables to constrain the public variables into constants. Here, the negotiation processes were shortened. This is mainly because the possible choices for the public variables are reduced as they are more tightly constrained. When private variables tightly constrain the public variables, NRS is the best configuration. This has implications for change of conflict resolution method. In firing position example, if agents are in a situation with tightly constrained public variables, they decide to resolve conflicts in NRS.

5 Conclusion

We implemented a system called CONSA for collaborative conflict resolution and applied it to two complex, dynamic domains: battlefield simulations and distributed sensor systems. We then modeled our argumentation-based approach in distributed constraint satisfaction problems (DCSP) to investigate the effect of different conflict resolution configurations, such as the degree of shared responsibility and unshared information, and their effects on scalability. Our results suggest some interesting correlations between these configurations and the performance of conflict resolution.

6 Acknowledgements

This research is funded by DARPA ITO award number F30602-99-2-0507.

References

[1] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104:1–70, 1998.

[2] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research (JAIR)*, 7:83–124, 1997.

[3] M. Tambe and H. Jung. The benefits of arguing in a team. *AI Magazine*, 20(4), 1999.

[4] Makoto Yokoo and Katsutoshi Hirayama. Distributed constraint satisfaction algorithm for complex local problems. In *Proc. of the Third Intl. Conf. on Multi-Agent Systems(ICMAS)*, July 1998.