Learning to Form Negotiation Coalitions in a Multiagent System

Leen-Kiat Soh
Computer Science and Engineering
University of Nebraska
115 Ferguson Hall
Lincoln, NE
(402) 472-6738
lksoh@cse.unl.edu

Costas Tsatsoulis
Dept. of Electrical Engineering and Computer Science
Information and Telecommunication Technology Center
University of Kansas
Lawrence, KS 66044
(785) 864-7749
tsatsoul@ittc.ukans.edu

Abstract

In a multiagent system where agents are peers and collaborate to achieve a global task or resource allocation goal, coalitions are usually formed dynamically from the bottom-up. Each agent has high autonomy and the system as a whole tends to be anarchic due to the distributed decision making process. In this paper, we present a negotiation-based coalition formation approach that learns in two different ways to improve the chance of a successful coalition formation. First, every agent evaluates the utility of its coalition candidates via reinforcement learning of past negotiation outcomes and behaviors. As a result, an agent assigns its task requirements differently based on what it has learned from its interactions with its neighbors in the past. Second, each agent uses a case-based reasoning (CBR) mechanism to learn useful negotiation strategies that dictate how negotiations should be executed. Furthermore, an agent also learns from its past relationship with a particular neighbor when conducting a negotiation with that neighbor. The collaborative learning behavior allows two negotiation partners to reach a deal more effectively, and agents to form better coalitions faster.

Introduction

We have developed a multiagent system that uses negotiation between agents to form coalitions for solving task and resource allocation problems. All agents of this system are peers, and decisions are made autonomously at each agent, from the bottom up. Each agent maintains its own knowledge and information base (and thus has a partial view of the world) and is capable of initiating its own coalitions. Due to the incomplete view and the dynamics of the world, an agent is unable or cannot afford to rationalize to form an optimal coalition. Hence, we have designed a negotiation-based coalition formation methodology, in which an agent forms a sub-optimal initial coalition based on its current information and then conducts concurrent 1-to-1 negotiations with the members of the initial coalition to refine and finalize the coalition. To form better coalitions faster, we have incorporated two learning mechanisms. First, each agent maintains a history of its relationships with its neighbors, documenting the negotiation experiences and using reinforcement learning to form potentially more viable and useful coalitions. Consequently, an agent assigns its task requirements differently based on what it has learned from its interactions with its neighbors in the past. Second, each agent has a case-based reasoning (CBR) module with its own case base that allows it to learn useful negotiation strategies. In this manner, our agents are able to form better coalitions faster.

An agent maintains its own knowledge base that helps determine its behavior—who to include in a coalition, how to negotiate to convince a neighbor to be in the coalition, and when to agree to join a coalition. As the agent continues reasoning, it adapts its behavior to what it has experienced. This in turn impacts its negotiation partners and collaborative learning is transferred implicitly, allowing a negotiation partner to update its relationship with the agent. This dynamic, negotiation-based coalition formation strategy can be viewed as an effective and efficient tool for collaborative learning. The efficiency comes from the fact that information is exchanged only when necessary during a negotiation.

The driving application for our system is multisensor target tracking, a distributed resource allocation and constraint satisfaction problem. The objective is to track as many targets as possible and as accurately as possible using a network of fixed sensors under real-time constraints. A "good-enough, soon-enough" coalition has to be formed quickly since, for example, to track accurately a target moving at half a foot per second requires one measurement each from at least three different sensors within a time interval of less than 2 seconds. Finally, the environment is noisy and subject to uncertainty and errors such as message loss and jammed communication channels.

In this paper, we first discuss present our coalition formation model. Then we briefly talk about our argumentative negotiation approach. After that, we discuss how reinforcement learning and case-based learning are used in the selection and ranking of coalition members, the determination of negotiation strategies, the task assignment among coalition candidates, and the evaluation of an on-going negotiation for evidence support. Then, we present and discuss some results. Before we conclude the paper, we outline some future work and interesting issues.

Coalition Formation

In dynamic, distributed problem solving or resource allocation, an initiating agent (or initiator) must form a coalition with other agents, so that it can perform its task. The members of a coalition are selected from a *neighborhood*, a set of other agents known to the initiator. In our application domain, an agent a_i is a neighbor to an agent a_j if a_i's sensor coverage area overlaps that of a_j. An initiator first selects members in the neighborhood that are qualified to be part of an initial coalition. Second, it evaluates these members to rank them in terms of their potential utility value to the coalition. Third, it initiates negotiation requests to the top-ranked members, trying to convince them to join the coalition. In the end, the coalition may fail to form because of the members refusing to cooperate, or may form successfully when enough members reach a deal with the initiating agent. Finally, the agent sends a confirmation message to all coalition members involved to announce the success or failure of the proposed coalition[1]. If it is a success, then all coalition members that have agreed to join will carry out their respective tasks at planned time steps.

To establish who can provide useful resources, the initiator calculates the position and the velocity of the target it is tracking and establishes a potential future path that the target will follow. Next, the initiator finds the radar coverage areas that this path crosses and identifies areas where at least three radars can track the target (remember that tracking requires almost simultaneous measurement from at least three sensors). The agents controlling these radars become members of the initial coalition.

Since computational resources are limited, and negotiating consumes CPU and bandwidth, the initiator does not start negotiation with all members of the coalition, but first ranks them and then initiates negotiation with the highest-ranked ones. Ranking of the coalition members is done using a multi-criterion utility-theoretic evaluation technique. There are two groups of evaluation criteria, one problem-related, and another one experience-based. The problem-related criteria are: (1) the target's projected time of arrival at the coverage area of a sensor: there has to be a balance between too short arrival times which do not allow enough time to negotiate and too long arrival times which do not allow adequate tracking; (2) the target's projected time of departure from the coverage area of a sensor: the target needs to be in the coverage area long enough to be illuminated by the radar; (3) the number of overlapping radar sectors: the more sectors that overlap the higher the chance that three agents will agree on measurements, thus achieving target triangulation; and (4) whether the initiator's coverage overlaps the coverage area of the coalition agent: in this case the initiator needs to convince only two agents to measure (since it is the third one), which may be easier than convincing three.

We will discuss further our experience-based criteria in the form of distributed case-based learning and utility-driven reinforcement learning in the following sections. In summary, at the end of the evaluation all coalition members are ranked and the initiator activates negotiations with as many high-ranked agents as possible (there have to be at least two and the maximum is established by the negotiation threads available to the initiator at the time, since it may be responding to negotiation requests even as it is initiating other ones).

Argumentative Negotiations

Our agents use a variation of the *argumentative negotiation model* (Jennings *et al.* 1998) in which it is not necessary for them to exchange their inference model with their negotiation partners. Note that after the initial coalition formation, the initiator knows who can help. The goal of negotiations is to find out who is willing to help. To do so, first the initiator contacts a coalition candidate to start a negotiating session. When the responding agent (or responder) agrees to negotiate, it computes a *persuasion threshold* that indicates the degree to which it needs to be convinced in order to free or share a resource (alternatively, one can view the persuasion threshold as the degree to which an agent tries to hold on to a resource). Subsequently, the initiator attempts to convince the responder by sharing parts of its local information. The responder, in turn, uses a set of domain-specific rules to establish whether the information provided by the initiator pushes it above a resource's persuasion threshold, in which case it frees the resource. If the responder is not convinced by the evidential support provided by the initiator, it requests more information that is then provided by the initiator. The negotiation continues based on the established strategy and eventually either the agents reach an agreement, in which case a resource or a percentage of a resource is freed, or the negotiation fails. Note that, motivated to cooperate, the responder also counter-offers when it realizes that the initiator has exhausted its arguments or when time is running out for the particular negotiation. How to negotiate successfully is dictated by a negotiation strategy, which each agent derives using case-based reasoning (CBR). CBR greatly limits the time needed to decide on a negotiation strategy, which is necessary in our real-time domain since the agent does not have to compute its negotiation strategy from scratch.

[1] Due to unreliable inter-agent communications, a confirmation message may not arrive at the coalition members on time. We implemented one mechanism and are considering another to improve the fault tolerance of the system. First, a coalition member will hold on to a commitment (though not yet confirmed) until it is no longer feasible to do so—for example, when the duration of the task is not cost-effective. The second improvement is to make persistent communication depending on the time between now and the start of the planned coalition. If the agent has more time, then it will make more attempts to communicate the confirmation.

Reinforcement Learning for Coalition Formation

Each agent keeps a profile of its neighborhood[2], and current and past relationships with its neighbors, and the selection of the potential members of an initial coalition is based on this profile. The current relationship is based on the negotiation strains and leverage between two agents at the time when the coalition is about to be formed. The past relationship, however, is collected over time and enables an agent to adapt to form coalitions more effectively via reinforcement learning. This we will discuss in this section. First, we define the past relationship between an agent a_i and a candidate α_k. Suppose that the number of negotiations initiated from an agent a_i to α_k is $\Sigma_{negotiate}(a_i \rightarrow \alpha_k)$, the number of successful negotiations initiated from an agent a_i to α_k is $\sum_{negotiate}^{success}(a_i \rightarrow \alpha_k)$, the number of negotiation requests from α_k that a_i agrees to entertain is $\sum_{negotiate}^{entertain}(\alpha_k \rightarrow a_i)$, the total number of all negotiations initiated from a_i to all its neighbors is $\Sigma_{negotiate}(a_i \rightarrow \eta_{a_i})$, and the total number of all successful negotiations initiated from a_i to all its neighbors is $\sum_{negotiate}^{success}(a_i \rightarrow \eta_{a_i})$. In our model, $rel_{past,a_i}(\alpha_k,t)$ includes the following:

(a) the helpfulness of α_k to a_i:

$$\frac{\sum_{negotiate}^{success}(a_i \rightarrow \alpha_k)}{\Sigma_{negotiate}(a_i \rightarrow \alpha_k)},$$

(b) the importance of α_k to a_i:

$$\frac{\sum_{negotiate}(a_i \rightarrow \alpha_k)}{\Sigma_{negotiate}(a_i \rightarrow \eta_{a_i})},$$

(c) the reliance of a_i on α_k:

$$\frac{\sum_{negotiate}^{success}(a_i \rightarrow \alpha_k)}{\sum_{negotiate}^{success}(a_i \rightarrow \eta_{a_i})}.$$

Similarly, the agent knows of how useful it has been to its potential coalition partner α_k:

(d) the friendliness of a_i to α_k:

$$\frac{\sum_{negotiate}^{entertain}(\alpha_k \rightarrow a_i)}{\sum_{negotiate}(\alpha_k \rightarrow a_i)},$$

(e) the helpfulness of a_i to α_k:

$$\frac{\sum_{negotiate}^{success}(\alpha_k \rightarrow a_i)}{\sum_{negotiate}^{entertain}(\alpha_k \rightarrow a_i)},$$

(f) the relative importance of a_i to α_k:

$$\frac{\sum_{negotiate}(\alpha_k \rightarrow a_i)}{\sum_{negotiate}(a_i \rightarrow \alpha_k)}, \text{ and}$$

(g) the reliance of α_k on a_i.

$$\frac{\sum_{negotiate}^{success}(\alpha_k \rightarrow a_i)}{\sum_{negotiate}^{success}(\alpha_k \rightarrow \eta_{\alpha_k})}.$$

Note that the above attributes are based on data readily collected whenever the agent a_i initiates a request to its neighbors or whenever it receives a request from one of its neighbors. The higher the value of each of the above attributes, the higher the potential utility the agent a_i may contribute to the coalition. The first three attributes tell the agent how helpful and important a particular neighbor has been. The more helpful and important that neighbor is, the better it is to include that neighbor in the coalition. The second set of attributes tells the agent the chance of having a successful negotiation. The agent expects the particular neighbor to be *grateful* and more willing to agree to a request based on the agent's friendliness, helpfulness and relative importance to that neighbor. To further the granularity of the above attributes, one may measure them along different event types: for each event type, the initiating agent records the above six attributes. This allows the agent to better analyze the utility of a neighbor based on what type of events that it is currently trying to form a coalition for. In that case, an event type would qualify all the above attributes.

An agent a_i can readily compute and update the first six attributes since they are based on the agent's encounter with its neighbors. However, an agent is not able to compute the last attribute, the reliance of α_k on a_i, since the denominator is based on α_k 's negotiation statistics. This attribute is updated only when agent a_i receives arguments from

[2] In our multiagent system, each agent maintains a neighborhood. An agent knows and can communicate to all its neighbors directly. This configuration of overlapping neighborhoods centered around each agent allows the system to scale up. Each agent, for example, only needs to know about its local neighborhood, and does not need to have the global knowledge of the entire system. In this manner, an agent can maintain its neighborhood with the various measures discussed in this section.

α_k. Thus, an agent updates the attributes from two sources. First, the update is triggered by the negotiation outcome directly. Second, the update comes from the information pieces contained in the arguments (when the initiator reminds the responder of the responder's reliance on the initiator). With the argumentative negotiation between agents, and the ability of an agent to conduct multiple, concurrent, coordinated 1-to-1 negotiations, the above profiled attributes facilitate a convergence of useful coalitions, allowing multiagent system as a whole to learn to collaborate more efficiently.

From Learning to Selecting Better Coalition Members

An agent evaluates its coalition candidates and ranks them based on their potential utility value. The potential utility, PU_{α_k,a_i}, of a candidate α_k is a weighted sum of the past relationship, i.e., $rel_{past,a_i}(\alpha_k,t)$, the current relationship, i.e., $rel_{now,a_i}(\alpha_k,t)$, and the ability of the candidate to help with the task e_j, i.e., $ability_{a_i}(\alpha_k,e_j,t)$. The ability value is domain-dependent and predefined (e.g. an IR sensor is better suited for night detection than an optical one). The current relationship between an agent a_i and its neighbor α_k is defined as follows: Suppose the number of concurrent negotiations that an agent can conduct is N_{avail}, and the number of tasks that the agent a_i is currently executing as requested by α_k is $\sum_{execute}(task : initiator(task) = \eta_{k,a_i})$. Suppose $\sum_{negotiate}^{success}(a_i \rightarrow \alpha_k)$ is the number of ongoing negotiations initiated from a_i to α_k. Then, $rel_{now,a_i}(\alpha_k,t)$ includes the following:

(a) negotiation strain between a_i and α_k: $\dfrac{\sum_{negotiate}^{ongoing}(a_i \rightarrow \alpha_k)}{N_{avail}}$,

(b) negotiation leverage between a_i and α_k: $\dfrac{\sum_{negotiate}^{ongoing}(\alpha_k \rightarrow a_i)}{N_{avail}}$, and

(c) degree of strain on a_i from α_k: $\dfrac{\sum_{execute}(task : initiator(task) = \alpha_k)}{\sum_{execute}(task : initiator(task) \in \eta_{a_i})}$.

The first attribute is inversely proportional to $rel_{now,a_i}(\alpha_k,t)$ and the other two are proportional to $rel_{now,a_i}(\alpha_k,t)$. The first attribute approximates how demanding the agent is of a particular neighbor. The more negotiations an agent is initiating to a neighbor, the more demanding the agent is and this strains the relationship between the two and the negotiations suffer. The last two attributes are used as a leverage that the agent can use against a neighbor that it is negotiating with, about a request initiated by the neighbor.

The potential utility is then:

$$PU_{\alpha_k,a_i} = W_{\Lambda_{ini}(a_i,e_j)} \bullet \left[rel_{past,a_i}(\alpha_k,t) \quad rel_{now,a_i}(\alpha_k,t) \quad ability_{a_i}(\alpha_k,e_j,t)\right]$$

where $W_{\Lambda_{ini}(a_i,e_j)} = \begin{bmatrix} w_{past,a_i,e_j} \\ w_{now,a_i,e_j} \\ w_{ability,a_i,e_j} \end{bmatrix}$ and $w_{past,a_i,e_j} + w_{now,a_i,e_j} + w_{ability,a_i,e_j} = 1$. Note that ultimately these weights may be dynamically dependent on the current status of a_i and the task e_j. And $rel_{past,a_i}(\alpha_k,t)$ is a weighted sum of the seven attributes previously discussed. As a result, an agent is reinforced to go back to the same neighbor (for a particular task) that it has had good past relationship in their interactions. In this manner, our agents are able to form more effective coalitions via reinforcement learning of which neighbors are useful for which tasks.

From Learning to Arguing More Effectively

As we mentioned earlier, when an initiating agent argues, it sends over different information classes. One of them is the world class which includes a profile of the neighbors. This information includes the past relationship attributes. Upon receiving these arguments, the responding agent evaluates the evidence support that they bring forth. Here we show an actual CLIPS rule that our agents use:

```
(defrule world-help-rate
  (world (_helpRate ?y&:(> ?y 0.6667)))
=>
  (bind ?*evidenceSupport* (+ (* 0.05 ?y)
  ?*evidenceSupport*))
```

This rule says if the attribute helpfulness is greater than 0.6667, then add 5 percent of that rate to the evidence support. This means that the responding agent is reinforced to agree to a negotiation by its previous interactions with the initiating agent. And the more successes the initiating agent has with a particular neighbor, the more effectively it can argue with that neighbor due to its reinforcement learning.

As a result, both agents learn to conduct better negotiations. Note also that this collaborative learning via inter-agent negotiations allow both agents to improve their performance because both agents are motivated to cooperate to achieve global goals.

Case-Based Learning for Better Negotiations

Before a negotiation can take place, an agent has to define its negotiation strategy, which it learns from the retrieved, most similar old case. Note that a better negotiation does not mean one that always results in a deal. A better negotiation is one that is effective and efficient, meaning that a quick successful negotiation is always preferred, but if a negotiation is going to fail, then a quickly failed negotiation is also preferred.

A negotiation process is situated. That is, for the same task, because of the differences in the world scenarios, constraints, evaluation criteria, information certainty and completeness, and agent status, an agent may adopt a different negotiation strategy. A strategy establishes the types of information to be transmitted, the number of communication acts, the computing resources needed, and so on. To represent each situation, we use cases. Each agent maintains two case bases, one for the agent as an initiator, the other for the agent as a responder. In general, an initiator is more conceding and agreeable and a responder is more demanding and unyielding. Each agent learns from its own experiences and thus evolves its own case bases.

In our work a case contains the following information: a situation space, a solution space, and an outcome. The situation space documents the information about the world, the agent, and the neighbors when the task arises. The solution space is the negotiation strategy that determines how an agent should behave in its upcoming negotiation. In an initiator case, the situation features are the list of current tasks, what the potential negotiation partners are, the task description, and the target speed and location. The negotiation parameters are the classes and descriptions of the information to transfer, the time constraints, the number of negotiation steps planned, the CPU resource usage, the number of steps possible, the CPU resources needed, and the number of agents that can be contacted. In a responder case, the situation feature set consists of the list of current tasks, the ID of the initiating agent, the task description, the power and data quality of its sensor, its CPU resources available, and the status of its sensing sector. The negotiation parameters to be determined are the time allocated, the number of negotiation steps planned, the CPU usage, the power usage, a persuasion threshold for turning a sensing sector on (performing frequency or amplitude measurements), giving up CPU resources, or sharing communication channels. Finally, in both cases the outcome records the result of the negotiation.

When a task is formulated, an agent composes a new problem case with a situation space. Then it retrieves the most similar case from the case base by a weighted, parametric matching on the situation spaces. Given the most similar case, the agent adapts that solution or negotiation strategy to more closely reflect the current situation. Equipped with this modified negotiation strategy, the agent proceeds with its negotiation. Finally, when the negotiation completes, the agent updates the case with the outcome.

Incremental and Refinement Learning

After a negotiation is completed (successfully or otherwise), the agent updates its case base using an incremental and a refinement learning step.

During incremental learning the agent matches the new case to all cases in the case base and if it is significantly different from all other stored cases, then it stores the new case. When the agent computes the difference between a pair of cases, it emphasizes more the case description than the negotiation parameters since its objective is to learn a wide coverage of the problem domain. This will improve its future case retrieval and case adaptation. So, an agent learns good, unique cases incrementally. The difference measure is:

$$Diff(a,b)=\frac{\begin{pmatrix} w_{sit}\sum_{i=1toM} w_{sit,i}\left|f^{i}_{sit,a}-f^{i}_{sit,b}\right| + \\ w_{sol}\sum_{j=1toN} w_{sol,j}\left|f^{j}_{sol,a}-f^{j}_{sol,b}\right| \end{pmatrix}}{w_{sit}\sum_{i=1toM} w_{sit,i}+w_{sol}\sum_{j=1toN} w_{sol,j}},$$

where $f^{i}_{sit,k}$ is the ith situation feature for case k, and $f^{j}_{sol,k}$ is the jth solution feature for case k, $w_{sit,i}$ is the weight for $f^{i}_{sol,k}$, and $w_{sol,j}$ is the weight for $f^{j}_{sol,k}$, w_{sit} is the weight for the situation space, and w_{sol} is the weight for the solution space. Also, w_{sit} + w_{sol} = 1, and each $w_{sit,i}$ or $w_{sol,j}$ is between 0 and 1. Finally, the agent computes $\min_{b=1toK} Diff(a,b)$ for the new case a, and if this number is greater than a pre-determined threshold, then the case is learned.

Second, since we want to keep the size of the case base under control, especially for speed of retrieval and maintenance (since our problem domain deals with real-time target tracking), the agent also performs refinement learning. If the new case is found to be very similar to one of the existing cases, then it computes (1) the sum of differences (e.g., $\sum_{b=1toK} Diff(a,b)$) between that old case and the entire case base minus the old case; and (2) the sum of differences between the new case and the entire case base minus the old case. This establishes the *utility* of the new case and the old case (that the agent considers to replace with the new case). The agent chooses to keep the case that will increase the diversity of the case base. Thus, if the second sum is greater than the first sum, then the agent replaces the old case with the new one. In this manner, we are able to gradually refine the cases in the case base while keeping the size of the case base under control.

Results

We have built a fully-integrated multiagent system with agents performing end-to-end behavior. In our simulation, we have any number autonomous agents and Tracker modules. A Tracker module is tasked to accept target measurements from the agents and predict the location of the target. The agents track targets, and negotiate with each other. We have tested our agents in a simulated environment and with actual hardware of up to eight sensors and two targets. For the following experiments, we used four sensors and one target moving about 60 feet between two points.

We ran two basic experiments: In the first one we investigated the effect of reinforcement learning in the quality of the resulting coalition, and in the second one we looked into the quality of the negotiation as a function of learning new cases of negotiation strategies. The quality was based in the number of successful negotiations, since when the agents reach a negotiated deal to jointly track a target, the overall system utility increases. Initial results indicate that the agents form coalitions with partners who are more willing to accommodate them in negotiation, and that the cases learned are being used in future negotiations. Agents that use learning of negotiation cases have between 40% and 20% fewer failed negotiations. Agents that use reinforcement learning to determine future coalition partners tend to prefer neighbors who are more conceding.

We also conducted experiments with four versions of learning: (1) both case-based learning and reinforcement learning (CBLRL), (2) only reinforcement learning (NoCBL), (3) only case-based learning (NoRL), and (4) no learning at all (NoCBLRL). Figure 1 shows the result in terms of the success rates for negotiations and coalition formations. As can be observed from the graph, the agent design with both case-based learning and reinforcement learning outperformed others in both its negotiation success rate and coalition formation success rate. That means with learning, the agents were able to negotiate more effectively (and perhaps more efficiently as well) that led to more coalitions formed. Without either case-based learning or reinforcement learning (but not both), the negotiation success rates remained about the same but the coalition formation rate tended to deteriorate. This indicates that without one of the learning mechanisms, the agents were still able to negotiate effectively, but may be not efficiently (resulting in less processing time for the initiating agent to post-process an agreement). Without both learning mechanisms, there was significant drop in the negotiation success rate. This indicates that the learning mechanisms did help improve

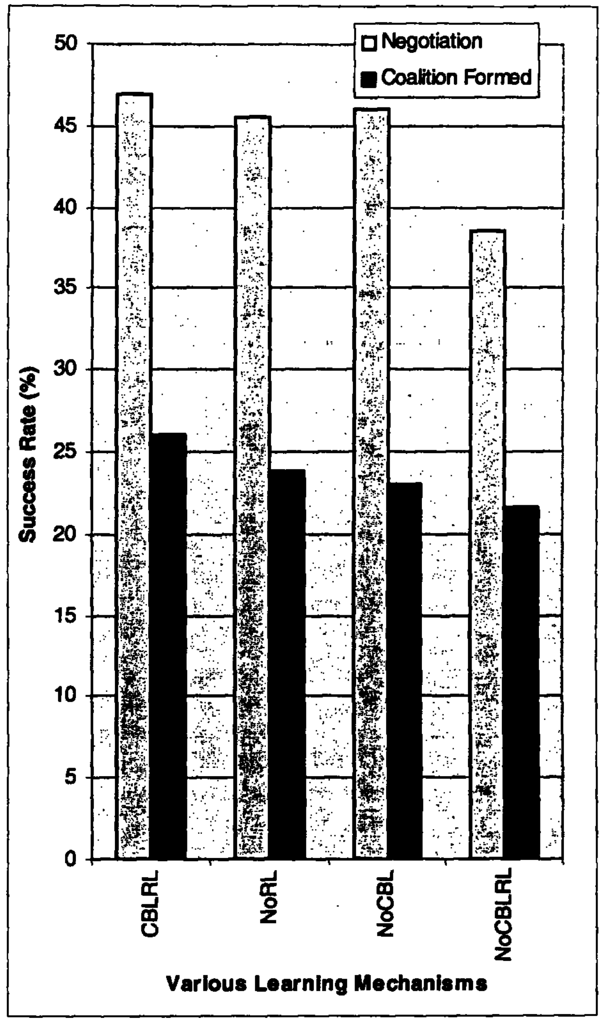

Figure 1 Success rates of negotiations and coalition formations for different learning mechanisms.

Note that the entire agent system is complicated with the end-to-end-behavior affected by various threads and environments. The results reported here need to be scrutinized further to isolate learning, monitoring, detection, reasoning, communication, and execution components in the evaluation: An agent busy tracking will not entertain a negotiation request that requires it to give up the resources that it is using to track. That refusal leads to a failure on the initiator's side. Another point worth mentioning is the real-time nature of our system and experiments. Added learning steps may cause an agent to lose valuable processing time to handle a coalition formation problem. More frequent coalition formations may flat out prevent other negotiations to proceed as more agents will be tied up in their scheduled tasks, part of the commitments to the coalitions.

Future Research Issues

There are other learning issues that we plan investigate. Of immediate concerns to us are the following

(1) We plan to investigate the learned cases and the overall 'health' of the case bases. We plan to measure the diversity growth of each case base (initiating and responding) for each agent. We also plan to examine the relationships between an agent's behavior (such as number of tracking tasks, the number of negotiation tasks, the CPU resources allocated, etc.) with the agent's learning rate. For example, if an agent is always busy, does it learn more or less, or learn faster or more slowly?

(2) We plan to investigate the learning of "good-enough, soon-enough" solutions. Under time constraints, agents cannot afford to perform complicated learning. Should the agent learn *hastily* sub-optimal solutions to its coalition formation and negotiation ventures? Or should it keep track of its coalition formation success rates and its negotiation success rates? For example, if one of the rates drops below an acceptable level, should an agent ask for help from more successful agents and perhaps learn their cases? If so, then we may have to incorporate time issues in this behavior to trade-off between the loss of the target tracking time, and the gain of higher coalition formation rates.

(3) We plan to investigate further the effects of reinforcement learning on the agents' coalition formation capabilities. Will each agent learn to approach a static coalition formation for each particular task? That is, will the reinforcement learning inhibit an agent to look to some other neighbors for help instead of the same old group of neighbors? If so, this could be harmful to the system?

(4) We plan to investigate cooperative and distributed case-based learning (Martin and Plaza 1998; Martin *et al.* 1998; Plaza *et al.* 1997). However, instead of evaluating the worth of such learning in terms of tasks getting executed, we want to concentrate on coalitions getting formed. We believe that combining case-based learning with negotiations allows the agents to exchange experiences selectively—a negotiation occurs only when necessary. This facilitates a more conservative but potentially more powerful learning behavior as the learning is reinforced through subsequent agent processes.

Conclusions

We have described the combination of reinforcement and case-based learning in negotiating multiagent systems. Each agent keeps a profile of its interactions with its neighbors and keeps a log on their relationships. This allows an agent to compute a potential utility of each coalition candidate and only approach those with high utility values to increase the chance of a successful coalition formation. This is a form of reinforcement learning as agents are able to learn to go back to the same agents that have cooperated before and to argue more effectively. The approach of negotiation-based coalition facilitates collaborative learning efforts among the agents as the negotiation outcomes influence each agent's perception of its neighbors, which, in turn, plays an important role in forming better coalitions faster. This learning is derived from the outcome of the negotiations and also from the information exchanged during a negotiation. Each agent also uses case-based reasoning to derive a negotiation strategy. When a negotiation completes, the resultant new case is learned if it increases the diversity of the case base. The integration of these learning mechanisms leads to more effective negotiations and deals between agents. Initial results support this avenue of research and show that learning agents perform better and more successful negotiations.

Acknowledgments

The work described in this paper is sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-99-2-0502. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory, or the U.S. Government.

References

Jennings, N. R., Parsons, S., Noriega, P., and Sierra, C. 1998. On Argumentation-Based Negotiation, in *Proceedings of Int. Workshop on Multi-Agent Systems*, Boston, MA.

Martin, F. and Plaza, E. 1998. Auction-Based Retrieval, in *Proceedings of the 2nd Congres Catala d'Intelligencia Artificial*, Girona, Spain, 136-145.

Martin, F., Plaza, E. and Arcos, J. L. 1998. Knowledge and Experience Reuse through Communication among Competent (Peer) Agents, *International Journal of Software Engineering and Knowledge Engineering*, 9(3):319-341.

Plaza, E., Arcos, J. L., and Martin, F. (1997). Cooperative Case-Based Reasoning, in Weiss, G. (ed.) *Distributed artificial intelligence meets machine learning, Lecture Notes in Artificial Intelligence number 1221*, Springer Verlag, 180-201.

Soh, L.-K. and Tsatsoulis, C. 2001. Reflective Negotiating Agents for Real-Time Multisensor Target Tracking, in *Proceedings of IJCAI'01*, Seattle, WA, 1121-1127.