Cautious Cooperative Learning with Distributed Case-Based Reasoning

Leen-Kiat Soh
Department of Computer Science and Engineering
University of Nebraska, 115 Ferguson Hall, Lincoln, NE 68588-0115 USA
lksoh@cse.unl.edu

Abstract

In this paper, we propose a cautious cooperative learning approach using distributed case-based reasoning. Our approach consists of two learning mechanisms: individual and cooperative learning. Normally, an agent conducts individual learning to learn from its past behavior. When the agent encounters a problem that it has failed to solve (satisfactorily), it triggers cooperative learning, asking for help from its neighboring agents. To avoid corrupting its own casebase and incurring costs on itself and other agents, our agent employs an axiomatic, cautious strategy that includes the notion of a chronological casebase, a profile-based neighbor selection, and a case review and adaptation before adopting an incoming case. Here we report on the approach and some results in a real-time negotiation domain.

Introduction

In distributed reasoning among agents, there is a balance to achieve such that autonomy and uniqueness among agents are preserved while exchanging and sharing experiences and knowledge to improve the system performance. For autonomy, agents solve the problems they encounter individually, with minimal coordination to avoid dependencies and consequent synchronization delays. For uniqueness, agents maintain heterogeneous expertise, with minimal overlap in capabilities to improve individual utility. On the other hand, it may be useful for agents to exchange their different experiences and knowledge to help each other improve the understanding of the environment and each other's constraints and preferences. Though the sharing of such reasoning-level information may lead to better system performance, it adds to the processing cost and reduces the autonomy of the agents. This concern was evident in (Marsella et al. 1999) in which the authors warned that some multi-agent environments could lead to a significant role specialization of individuals, and that sharing experiences of individuals in different roles or equivalently training individuals by letting them execute different roles could sometimes be significantly detrimental to team performance.

In view of this, we propose a multi-strategy learning methodology that practices cautiousness in an agent's adoption of other agents' reasoning outcomes, within a case-based reasoning (CBR) framework in a multiagent environment. Each agent is capable of individual and cooperative learning. Individual learning refers to learning based on an agent's perceptions and actions, without communicating directly with other agents in the environment. Cooperative learning refers to learning through interaction among agents. The objective here is to learn something better to solve a problem that an agent has failed to solve or solve satisfactorily. Our methodology employs an axiomatic, cautious adaptive mechanism to combine the two, an interaction protocol for soliciting and exchanging information, a profile-based neighbor selection and the idea of a chronological casebase.

In our multiagent domain, agents negotiate to collaborate on real-time tasks such as multi-sensor target tracking and CPU resource allocation. When agents negotiate, each follows a dynamically generated negotiation strategy. Each agent derives its negotiation strategy for each negotiation using CBR. That is, each agent has two casebases—one as an initiating agent, one as a responding agent. When an agent encounters a negotiation problem, it takes a snapshot of the environment parameters and forms a problem description. It then searches one of the casebases for the most similar case. After finding the best case, it adapts the solution (the negotiation strategy) of the best case to the current problem. It then uses the new negotiation strategy to carry out its negotiation. When the negotiation completes, the agent documents the negotiation task. If the agent has failed to successfully negotiate for a particular problem regularly, it triggers cooperative learning.

This paper outlines our cautious cooperative learning approach and its axiomatic design and implementation, and reports on two sets of experiments of the approach.

Related Work

There has been research in distributed and cooperative case-based reasoning (CBR). In (Prasad and Plaza, 1996), the authors proposed treating corporate memories as distributed case libraries. Resource discovery was achieved

Copyright © 2003, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

through (1) negotiated retrieval that dealt with retrieving and assembling case pieces from different resources in a corporate memory to form a good overall case, and (2) federated peer learning that deal with distributed and collective CBR in (Plaza *et al.*, 1997). In (Martin *et al.*, 1999), Martin extended the model using the notion of competent agents. In (Martin and Plaza 1999), Martin and Plaza employed an auction-based mechanism that focused on agent-mediated systems. The objective here was to determine the best case from the bid cases.

McGinty and Smyth (2001) utilized collaborative CBR for personalized route planning where an agent requests for help when it is not able to solve a particular problem. An agent determines the quality of a peer based on the coverage of a peer's capability in solving the problem, and the similarity between the agent and the peer. The similarity is computed based on the number of problems the agents have in common and on the similarity of their solutions to these common problems. Leake and Sooriamurthi (2001, 2002a, 2002b) proposed a multi-case-base reasoning architecture involving (a) case-base characterization, (b) problem dispatching, (c) case selection, (d) solution merging, (e) cross-case-base adaptation, and (d) multi-case-base maintenance. The underlying approach is multiple casebases working together, dispatching problems to each other, and adopting cases for local use. Leake and Sooriamurthi (2002a) presented three adoption strategies: (a) linear interpolation based on the extrema of the external and local casebases, (b) local approximation with the starting point from the external casebase, and (c) local approximation with the starting point from the local casebase.

Recently, Ontañón and Plaza (2003) proposed a cooperative case retention strategy combing ideas from CBR and active learning techniques. The basic idea in active learning is that the learner receives a set of unlabeled examples and decides which of them are interesting to learn from; then the teacher labels the examples that the learner has found interesting and they are used for learning. In the MAQbC algorithm proposed, when an agent *A* asks another agent *B* to help solve a problem, the interaction protocol is as follows. First, *A* sends a problem description *P* to *B*. After *B* has tried to solve *P* using its casebase, it either informs *A* that (a) it cannot solve the problem, or (b) a solution endorsement record (SER). A SER is a vector of cases in a casebase that endorse a solution for *P*. After *A* receives all SERs from the agents that it has asked for help, it conducts a vote to combine the information. The solution class with the most endorsing cases wins the ballot. *A* then either keeps the solution, or sends copies to other agents, if the problem *P* is deemed to be interesting.

Our proposed multi-strategy methodology emphasizes individual learning while cautiously adopting cooperative learning when necessary. Cooperative learning differs from collective CBR in that it does not merge case pieces into one as it considers entire cases. In addition, our research focus here is to define a utility-based cautious mechanism that combines individual and cooperative learning. Our methodology also differs from the multi-case-base reasoning framework of Leake and Sooriamurthi (2001) and focuses on a cautious approach *integrating* the monitoring of problematic cases, careful selection of which agent (or casebase) to approach for help, neighborhood profiling, and adaptation based on the differences between the local and external cases.

Methodology

Our innovative approach to cooperative learning is cautious using the notion of a chronological casebase, a profile-based neighbor selection, and a case review and adaptation before adopting an incoming case. As indicated earlier, our system practice cautiousness in sharing reasoning outcomes because of cost in processing and risk in knowledge. The additional communication and coordination overhead may be too expensive or too slow for cooperative learning to be cost-effective or timely. Moreover, since an agent learns from its experience and its view of the world, its solution to a problem may not be applicable for another agent facing the same problem. This injection of *foreign* knowledge may also be risky as it may add to the processing cost without improving the solution quality of an agent.

In our domain, our agents learn how to negotiate better. The learned knowledge is encapsulated in cases: the negotiation task is the case problem; the negotiation strategy is the case solution; and the outcome of the negotiation is the case result. Our negotiation approach is argumentative (Soh and Tsatsoulis 2001), in which every agent may assume two different roles in its negotiation tasks. As an initiator, an agent tries to convince the responder to agree to give up certain resources or to help perform a task. As a responder, an agent evaluates the request against its own constraints. If the arguments supporting the request are convincing enough, then the responder will agree to a deal. Therefore, every agent maintains two casebases, one for each role. Under normal CBR operations, when an agent confronts a negotiation task, it retrieves the best case that matches the current problem, and adapts the solution of the best case as the negotiation strategy. The agent then uses the negotiation strategy for its negotiation. When the negotiation ends, it records the outcome of the negotiation. Then the agent determines whether to learn the new case—whether to store it in its casebases.

Cautious Axioms

Our cautious approach is based on a set of axioms, designed to prevent foreign reasoning outcomes to be accepted by an agent, specified in the following in terms of case-based reasoning.

Axiom 1 Self-Forgetting Casebase *A casebase must forget cases that have not been used recently to cut down case evaluation and search time.*

Axiom 1 is necessary to allow for the addition of new cases from other agents in two ways. First, if the new cases are useful, then we want to remove existing cases that have not been used to maintain a manageable casebase size. Second, if the new cases are useless, then we want to remove them from the casebase. This axiom allows our agents to do just that.

Axiom 2 Problematic Case *A casebase must be able to evaluate and identify problematic cases such that it can ask for help to solve the problem.*

Axiom 2 is necessary for an agent to monitor and track all its cases systematically and identify the case that it has not been able to solve or solve satisfactorily. This axiom also implies that an agent does not casually ask for new reasoning outcomes from other agents without a useful purpose.

Axiom 3 Profiling Neighbors *An agent keeps a profile of each of its neighbors through its interactions with them to evaluate their abilities in solving similar problems.*

Axiom 3 is necessary for an agent to target specific neighbors for help. Otherwise, an agent would have to approach all neighbors, leading to more communication and processing. With this axiom, an agent will be able to rank the best neighbor to ask for help. Note also that this axiom assumes that (1) the agents interact and (2) the agents are able to perceive how the other agents solve similar problems through these interactions.

Axiom 4 Neighbor Selection *An agent selects specific neighbors from which to ask for help.*

Axiom 4 requires an agent to be selective (and thus responsible) in its request for help. It should not blindly ask for help from all the agents that it knows. This axiom discourages an agent from sending requests to neighbors that will not be able to help or to provide good solutions. It also reduces the computational load for an agent—it does not have to manage too many requests to solve a single problem.

Axiom 5 Interaction Protocol *An agent must be able to communicate to its neighbors for help related to a specific problematic case. Conversely, an agent must be able to respond to a request for help to solve a specific problematic case.*

Axiom 5 is necessary to facilitate cooperative learning. It basically states that an interaction protocol has to be in place for the learning to take place. Note that it emphasizes on a specific problematic case—meaning that the *requesting* agent does not blindly ask for good cases from its neighbors and that the *helping* agent does not blindly disseminate good cases to its neighbors. This axiom prevents agents from spamming its neighbors with requests.

Axiom 6 Review and Adaptation *An agent must review and adapt if necessary incoming cases before adopting them into its own casebase.*

Axiom 6 is the final cautionary step before finally adding foreign cases to an agent's casebase. This axiom allows an agent to evaluate the incoming reasoning outcomes, thus (1) reducing the cognitive load on its neighbors—the neighbors do not have to help with the correct solution, as long as the neighbors think they are providing help with the best solution that they have, and (2) adapting the incoming reasoning outcomes to better suit an agent's problem. Note that coupling this axiom with axiom 3, an agent increases the chance of receiving a suitable solution from its neighbors by approaching only the neighbors that are potentially helpful. Also, with this axiom, a *helping* agent is more likely to help as it knows that (1) it does not need to interact with the *requesting* agent to investigate the problematic case in details, and (2) it does not need to guarantee that its solution is useful to the *requesting* agent.

With these axioms, we introduce the notion of a chronological casebase, case usage history, neighbor profiling, and profile-based neighbor selection.

Chronological Casebases & Usage History

We utilize the notion of a chronological casebase in which each case is stamped with a time-of-birth (when it was created) and a time-of-membership (when it joined the casebase). All initial cases are given the same time-of-birth and time-of-membership. A foreign case, however, may have a much earlier time-of-birth than a time-of-membership when imported to a local casebase. In addition, we profile each case's usage history (Table 1).

Parameters	**Description**
_timesOfUsed	the number of times the case has been used
_timesOfSuccessUsed	the number of times the case has been used in a successful negotiation
_timesOfIncurNewCase	the number of times the usage of the case has led to a new case getting added to the casebase
_timesOfRequest	the number of times the case has been designated as a problematic case, i.e., with very low utility
_timeStamp	the last time that the case was used or the time when the case was added

Table 1 The usage history that an agent profiles of each case

An agent evaluates the performance of a case based on its usage history. If a case is deemed to have a problematic performance, then a cooperative learning will be triggered and the case will be replaced. The following supports the Self-Forgetting Casebase Axiom and Problematic Case Axiom. Here are some heuristics we use in tandem with the chronological casebase:

H1 Currency: If a case has not been used in a long time, then this case is more likely to be replaced.

H2 Evolution: With everything else equal, an old case is more likely to be replaced than a young case.

H3 Usefulness: If a case's *_timesOfSuccessUsed* is significantly small, then the case is more likely to be replaced.

H4 Solution Quality I: If a case has a high *_timesOfUsed* but a low *_timesOfSuccessUsed*, then it is problematic.

H5 Solution Quality II: If a case has a low *_timesOfSuccessUsed*, and a high *_timesOfIncurNewCase*, then the solution of this case is probably not suitable for the problems encountered by the agent and it is problematic.

Individual Learning

When a negotiation completes, if the *new* case is useful and adds to the casebase's diversity, the agent learns it. This constitutes the basis of the individual learning strategy of our methodology. If the casebase's size has reached a preset limit, then the agent considers replacing one of the existing cases with the new case. This feature enables our agents to perform both incremental and refinement learning. Interested readers are referred to (Soh and Tsatsoulis 2002) for details. For the refinement learning, we use heuristics ***H1***, ***H2***, and ***H3***. Hence, our individual learning satisfies Axiom 1.

Cooperative Learning

However, if an agent consistently fails to negotiate successfully given a particular problem description, it needs to look for a better solution (i.e., a better negotiation strategy). This motivates the cooperative learning strategy of our methodology. As previously mentioned, we have adhered to a cautious approach to cooperative learning.

Problematic Case. The agent evaluates the case to determine whether it is problematic. To designate a case as problematic, we use heuristics ***H4*** and ***H5***: a (frequently used) case is problematic if it has a low success rate (*_timesOfSuccessUsed/_timesOfUsed*) and a high incurrence rate (*_timesOfIncurNewCase/_timesOfUsed*). That means the case addresses an appropriate problem but does *not* provide a satisfactory solution. With this, we satisfy Axiom 2.

Neighborhood Profiling. An agent keeps a profile of every neighbor that documents the negotiation relationships between the agent and the neighbor (Soh and Tsatsoulis 2002). When an agent initiates a negotiation to one of its neighbors, it increases the number of requests to that neighbor. When the agent concludes a negotiation to that neighbor, it updates accordingly the number of successful negotiations or failed negotiations. When an agent receives a negotiation request and agrees to negotiate, it also increments the number of requests from that neighbor. When the agent concludes the negotiation, it updates accordingly the number of successful or failed negotiations. Note that the problems of our domain are negotiation tasks, which inherently require interactions. Thus, our neighborhood profiling naturally satisfies Axiom 3 of the cautious approach. In general, in a multiagent system where agents interact, agents can profile the interactions by tagging each interaction with an eventual outcome status. If the agents do not interact, then it seems natural that the agents should not learn cooperatively.

Profile-Based Neighbor Selection. The agent only requests help from another agent that it thinks is good at a particular problem. The idea here is that we want to approach neighbors who have initiated successful negotiations with the current agent, with the hope that the agent may be able to learn how those neighbors have been able to be successful. Among the profiled parameters is *_helpRate*. This is the percentage of times that the agent has agreed to a request by a particular neighbor. The agent selects the neighbor with the highest *_helpRate* to ask for help, for example. This increases the chance that the neighbor may have a better solution than the agent's. With this selection, we satisfy Axiom 4. Note that *_helpRate* approximates the notion of *authority.* If agent *B* has been successful in getting *A* to perform certain tasks, then from the viewpoint of *A*, *B* is a very good source to obtain ask for help in getting other agents to perform certain tasks. We are currently investigating the notion of utility and task-specific resolutions for coalition formation. We aim to tie each selection to the outcome of request for help in a utility measure, and to profile a neighbor along different task types.

Interaction Protocol. The case exchange follows an interaction protocol. Briefly, when agent *A* wants to learn from agent *B*, it sends a *CASE_REQUEST* message that includes the problematic case to agent *B*. Agent *B* receives this message and retrieves the most similar case to the problematic case from its casebase. It then replies a *CASE_RESPONSE* message to *A* with the two cases attached. This protocol allows agent *A*, for example, to issue multiple requests for different problematic cases concurrently. It simplifies the internal management of problematic cases but imposes additional costs on communication especially when each case is large. This satisfies Axiom 5.

Review and Adaptation. The agent adapts the foreign case before adopting it into its casebase. This adaptation is similar to the adaptation that the CBR module performs after retrieving the most similar case from the casebase to the current problem at hand. At the same time, the usage history parameters of the foreign case—regardless of the number of times that it has been used or used successfully by the helping neighbor—are reset. This is done to prevent the case from dominating the casebase without first getting used at least once by the agent. With this mechanism, we satisfy Axiom 6. Along the line of authority, we are also investigating the review of the foreign case to refine the profiling and selection process. That is, if a foreign case has been used successfully by a neighbor, that means the neighbor has provided a good solution from its point of view. If this case undergoes significant adaptation, that means the case may not be useful to the *requesting* agent. That means the authority of the neighbor is high but the utility of the neighbor may be low. If a foreign case has not been used successfully by a neighbor yet is received as help, that means the neighbor has provided a bad solution from its point of view but a solution that matches the requested problem. If, subsequently, the case is useful to the *requesting* agent, then that means the neighbor has a low authority but a high utility to the *requesting* agent. We aim to use this information to improve neighbor selection; that is, to become even more cautious.

Implementation

Our application involves multi-sensor target tracking and CPU reallocation (Soh and Tsatsoulis 2001). Four agents live in a noisy environment (simulated by a Java-based program called RADSIM for sensors and targets, and C++ methods for process delays due to CPU starvation). When an agent detects a moving target, it tries to form a tracking coalition. A tracking coalition requires at least three members or agents. Thus, the initiating agent will try to recruit at least two neighbors to help out, and that prompts the agent to conduct negotiations. Meanwhile, when an agent detects a CPU shortage, it tries to obtain additional resource from other agents through negotiations. Each agent has 3 + *N* threads. It has a core thread that does the decision-making, manages the tasks, performs coalition formation, and oversees the negotiations. It has a communication thread that sends and receives messages. It has an execution thread that actuates the physical sensor: calibration, search-and-detect for a target, etc. It also has *N* active negotiation threads, allowing the agent to conduct multiple concurrent negotiations.

Experiments and Results

We conducted two sets of experiments, Comprehensive Experiment A (CEA) and Comprehensive Experiment B (CEB). We carried out CEA to study the effects of individual learning in subsequent cooperative learning, the roles of cooperative learning in agents of different initial knowledge and the feasibility of our multi-strategy learning methodology. We performed CEB to investigate the effects of the environment on the agents' learning. For details on our experiments and results, please refer to (Soh and Luo 2003; Luo 2003).

Comprehensive Experiment A (CEA)

We conducted four sets of experiments in CEA as shown in Table 2. The design of these experiment sets was to investigate how learning differed given different casebase sizes. Note that for the following experiments, case replacement started to take place after the size of the casebase reached 30.

Further, for each experiment set, we had two sub-experiments: (1) combine-first-combine-later (Exp1), and (2) individual-first-combine-later (Exp2). Each sub-experiment had two stages. In combine-first-combine-later, each stage employed both individual and cooperative learning. In individual-first-combine-later, the first stage employed individual-only learning and the second stage combined both. After the first stage, the casebases were *cleaned manually*: cases not used were deleted. This design was to create different initial casebases (for the second stage) for different agents, and to obtain initial casebases (for the second stage) of different diversity—the first sub-experiment should generate casebases that were more diverse since it employed cooperative learning in the first stage.

Experiment Set	A1	A2	A3	A4
ES1	16	16	16	16
ES2	2	16	16	16
ES3	16	16	16	28
ES4	2	10	20	28

Table 2 Experiment sets. For example, in ES1, every agent has 16 cases in its casebase; and so on.

We used two main parameters to evaluate the casebases: utility and diversity. To obtain utility, each type of outcomes is given a score. For example, a successful negotiation is scored 10 points. A failed negotiation due to a jammed communication channel is worth 6 points. Thus, the average utility of a case base is simply the average outcome utility of the cases. The diversity measure of a casebase is compiled from the average parametric difference of the cases in the casebase.

Briefly, we observe the following: Cooperative learning brings more utility and diversity per learning occurrence than individual learning. Cooperative learning brings more utility and diversity to an initial casebase previously grown using individual-only learning than one previously grown using both learning. A small casebase learns more effectively in terms of utility and diversity, but not faster since our learning is problem-driven. A large casebase learns in a similar manner as an average casebase except when it is greater than the preset limit that triggers case replacement. Our multi-strategy learning is able to adapt to the environment dynamically.

Comprehensive Experiment B (CEB)

The objective of CEB was to see how the learning results changed in different environments. In our problem domain, there are two kinds of coalitions formed among the agents: (1) multi-sensor target tracking, and (2) CPU re-allocation. A tracking coalition requires at least three agents in order to track a moving target. A CPU coalition requires at least only one neighbor to help. Moreover, a tracking task is durational such that it takes time to actually carry out the tracking task. However, a CPU re-allocation task is carried out at a point in time. In addition, a tracking task is highly time-constrained. A coalition has to be formed in time to *catch* the target before the target moves out of the sensor coverage area. Thus, negotiations related to tracking are more difficult to manage and handle.

There were three sets of experiments in CEB, as shown in Table 3. In ES1, the number of CPU coalitions attempted was greater than the number of tracking coalitions attempted. In ES2, the two numbers were about the same. In ES3, there were more tracking coalitions attempted than CPU coalitions.

	Combination of CPU and Tracking Coalitions
ES1	CPU coalitions more often than tracking
ES2	CPU and tracking coalitions similarly frequent
ES3	Tracking coalitions more often than CPU

Table 3 Sub-Experiments setup in CEB.

Briefly, we observe the following: Different environments affect agents' learning behavior. Depending on the frequency of a task and its characteristics, an agent may rely more on individual learning or cooperative learning. For example, if a type of tasks (tracking) is time consuming and durational, then increasing its frequency actually weakens the potential benefit of individual learning and encourages the agent to perform more cooperative learning. The environments impact the two initiating and responding roles differently, especially for negotiations associated with tough requirements. Since an initiating agent has to shoulder the coalition management and decision making, it is able to learn more and more diverse and useful cases. But, negotiating as a responder, an agent's responsibility is less and thus considers fewer issues—as a result the learning tends to be less impressive.

Conclusions

We have presented a multi-strategy learning methodology in a dynamic multiagent environment, where agents learn to negotiate better. The methodology consists of a cautious, axiomatic mechanism to combine individual and cooperative learning and an interaction protocol for soliciting and exchanging solutions. We have also introduced the notion of a chronological casebase together with a set of heuristics, a dynamic profiling of case usage history, and a relation-based neighbor selection. From our experiments, we concluded that our cautious cooperative learning brings more diversity and utility than individual learning and that different task types and negotiation roles affect the learning behavior of agents.

Acknowledgments

This work was partially supported by a grant from the DARPA ANTS project, subcontracted from the University of Kansas. The author would like to thank Juan Luo for the software implementation and experiments and the anonymous reviewers for their helpful comments.

References

Leake, D. B. and R. Sooriamurthi (2002a). Automatically Selecting Strategies for Multi-Case-Base Reasoning, *ECCBR 2002: Advances in Case-Based Reasoning*, Springer-Verlag, Berlin, 204-218.

Leake, D. B. and R. Sooriamurthi (2002b). Managing Multiple Case Bases: Dimensions and Issues, in *Proc. FLAIRS'2002*, pp. 106-115.

Leake, D. B. and R. Sooriamurthi (2001). When Two Case Bases are Better Than One: Exploiting Multiple Case Bases, in *Proc. ICCBR'2001*.

Luo, J. 2003. Case-Based Learning Behavior in a Real Time Multi-Agent System, Master's project report, Computer Science & Engineering Dept., Univ. of Nebraska.

Martin, F. J., Plaza, E., and Arcos, J. L. 1999. Knowledge and Experience through Communication among Competent (Peer) Agents, *Int. J. Software Engineering & Knowledge Engineering*, 9(3):319-341.

Martin, F. J. and Plaza, E. 1999. Auction-Based Retrieval, *Proc. 2nd Congres Catala d'Intel.ligencia Artificial*, 1-9.

Marsella, S., Adibi, J., Al-Onaizan, Y., Kaminka, G. A., Muslea, I., Tallis, M., and Tambe, M. 1999. On Being a Teammate: Experiences Acquired in the Design of RoboCup teams, *Proc. Agents'99*, 221-227.

McGinty, L. and B. Smyth (2001). Collaborative Case-Based Reasoning: Applications in Personalized Route Planning, *Proc. ICCBR'2001,*

Ontañón, S. and Plaza, E. 2003. Collaborative Case Retention Strategies for CBR Agents, *Proc. ICCBR*, Iceland, 392-406.

Plaza, E., Arcos, J. L., and Martin, F. 1997. Cooperative Case-Based Reasoning, in G Weiss (ed.) *Distributed Artificial Intelligence Meets Machine Learning*, LNAI, Springer Verlag, 1-21.

Prasad, M. V. N. and Plaza, E., 1996. Corporate Memories as Distributed Case Libraries, *Proc. KAW'96*, 1-19.

Soh, L.-K. and Luo, J. 2003. Multiagent Case-Based Reasoning through Individual and Cooperative Learning, *Proc. 18th IJCAI Workshop on Agents and Automated Reasoning*, 44-51.

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

Soh, L.-K. and Tsatsoulis, C. 2002. Learning to Form Negotiation Coalitions in a Multiagent System, *AAAI Spring Symp. Collaborative Learning Agents*.