Case Exchange Strategies in Multiagent Learning

Santiago Ontañón and Enric Plaza

IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia (Spain).
{santi,enric}@iiia.csic.es, http://www.iiia.csic.es

Abstract. Multiagent systems offer a new paradigm to organize AI applications. We focus on the application of Case-Based Reasoning to Multiagent systems. CBR offers the individual agents the capability of autonomously learn from experience. In this paper we present a framework for collaboration among agents that use CBR. We present explicit strategies for case retain where the agents take in consideration that they are not learning in isolation but in a multiagent system. We also present case bartering as an effective strategy when the agents have a biased view of the data. The outcome of both case retain and bartering is an improvement of individual agent performance and overall multiagent system performance. We also present empirical results comparing all the strategies proposed.

Keywords: Cooperative CBR, Multiagent CBR, Collaboration Policies, Bartering, Multiagent Learning.

1 Introduction

Multiagent systems offer a new paradigm to organize AI applications. Our goal is to develop techniques to integrate CBR into applications that are developed as multiagent systems. CBR offers the multiagent system paradigm the capability of autonomously learn from experience.

The individual case bases of the CBR agents are the main issue here. If they are not properly maintained, the overall system behavior will be suboptimal. These case bases must be maintained having in mind that the agents are not isolated, but inside a multiagent system. This enables the agent not to learn only from its own experience, but collaborating with the other agents in the system. The lacks in the case bases of some agents can be compensated by the experience of other agents in the system. In a real system, there will be agents that can very often obtain certain kind of cases, and that will very seldom obtain other types of cases. It will be beneficial for two agents if they reach an agreement to trade cases. This is a very well known strategy in the human history called *bartering*. Using case bartering, agents that have a lot of cases of some kind will give them to another agents in return to more interesting cases for them, and both will profit by improving their performance.

Our research focuses on the scenario of separate case bases that we want to use in a decentralized fashion by means of a multiagent system, that is to say a collection of CBR agents that manage individual case bases and can communicate (and collaborate)

with other CBR agents. Separate case bases make sense for different reasons like privacy or efficiency. If the case bases are owned by some organizations, perhaps they are not willing to donate the contents of its case bases to a centralized one where CBR can be applied. Moreover, in the case that the case bases where not private, more problems can arise from having all the cases in a single one, such as efficiency, storage or maintenance problems [5]. All these problems suggest difficulties that may be avoided by having separate case bases.

In this paper we focus in multiagent systems, where individual agents learn from its own experience using CBR. We show how the agents can improve its learning efficiency by collaborating with other agents, and show results comparing several strategies.

The structure of the paper is as follows. Section 2 presents the collaboration scheme that the agents use. Section 3 explains the strategies used by the agents to retain cases in their individual case bases . Then, section 4 presents a brief description of the bartering process. Finally, The experiments are explained in section 5 and the paper closes with related work and conclusion sections.

2 Multiagent Learning

A multiagent CBR ($\mathcal{M}$AC) system $\mathcal{M} = \{(A_i, C_i)\}_{i=1\ldots n}$ is composed on *n* agents, where each agent A_i has a case base C_i. In this framework we restrict ourselves to analytical tasks, i.e. tasks (like classification) where the solution is achieved by selecting from an enumerated set of solutions $\mathcal{K} = \{S_1 \ldots S_K\}$. A case base $C_i = \{(P_j, S_k)\}_{j=1\ldots N}$ is a collection of pairs problem/solution.

When an agent A_i asks another agent A_j help to solve a problem the interaction protocol is as follows. First, A_i sends a problem description *P* to A_j. Second, after A_j has tried to solve *P* using its case base C_j, it sends back a message that is either `:sorry` (if it cannot solve *P*) or a solution endorsement record (SER). A SER has the form $\langle S_k, P, A_j \rangle$, meaning that the agent A_j has found S_k as the most plausible solution for the problem P.

Voting Scheme. The voting scheme defines the mechanism by which an agent reaches an aggregate solution from a collection of SERs coming from other agents. Each SER is seen as a vote. Aggregating the votes from different agents for each class, we can now obtain the winning class as the class with the maximum number of votes. We will show now the $Committee$ collaboration policy that uses this voting scheme (see [6] for a detailed explanation and comparison of several collaboration policies, and a generalized version of the voting scheme to allow more complex CBR methods).

Committee Policy. In this collaboration policy the agent members of a $\mathcal{M}$AC system $\mathcal{M}$ are viewed as a committee. An agent A_i that has to solve a problem *P*, sends it to all the other agents in $\mathcal{M}$. Each agent A_j that has received *P* sends a solution endorsement record $\langle S_k, P, A_j \rangle$ to A_i. The initiating agent A_i uses the voting scheme above upon all SERs, i.e. its own SER and the SERs of all the other agents in the multiagent system. The problem's solution is the class with maximum number of votes.

In a single agent scenario, when an agent has the opportunity to learn a new case, the agent only has to decide whether the new case will improve its case base or not. Several retain policies exist to take this decision [2, 9]. But when we are in a multiagent

scenario, new factor must be considered. Imagine the following situation: an agent A_i has the opportunity to learn a new case C, but decides that C is not interesting to him. But there is another agent A_j in the system, that could obtain a great benefit from learning the case C. It would be beneficial for both agents, that the agent A_i retained the case, and then give or sell it to A_j.

Two different scenarios may be considered, when there are ownership rights over the cases and when the agents are free to make copies of the cases to send them to other agents. We will call the first scenario the *non-copy* scenario, and the second one the *copy* scenario. Several strategies for retaining cases and bargaining with the retained cases can be defined for each scenario.

The learning process in our agents has been divided in two subprocesses: The *case retain* process, and the *case bartering* process. During the *case retain* process the agent that receives the new case decides whether to retain or not the new case, and whether to offer the new case to the other agents or not. An alternative to offer the cases to other agents for free, is to offer them in change of more interesting cases. This is exactly what the *case bartering* process consists on. Thus, when an agent has recopilated some cases that are not interesting for him, he can interchange them for more interesting cases during a bartering process. This bartering process has not to be engaged necessary each time an agent learns a new case, but just when the agents decide that they have enough cases to trade with.

In the following sections we will first describe all the strategies that we have experimented with, and then we will give a brief description of the bartering process.

3 Case Retain

In this section, we will explain with detail all the strategies for the *case retain* process used in the experiments. This *case retain* strategies are used when an agent has the opportunity to learn a new case, and has to decide whether to retain it or not.

Each *case retain* strategy is composed on two different policies: the *individual retain* policy and the *offering* policy. For the *individual retain* policy, we have experimented with three options: *Never retain* (*NR*), where the agent that has the opportunity to learn a new case never retains the case; *Always retain* (*AR*), where the agent always retains the case and *When-Interesting retain* (*WIR*), where the agent only retains cases founded interesting. Notice that we can define the *interestingness* of a case in several ways. In our experiments the criterion for a case being *interesting* for an agent is when that case is incorrectly solved by the agent.

And for the *offering* policy, we have only two options: *Never offer* (*NO*), where the agent that has the opportunity to learn a new case never offers it to any other agent in the system and *Always offer* (*AO*), where the agent always asks if any of the other agents is interested in each case the agent has the opportunity to learn.

Now, combining all these options, we can define all the possible *case retain* strategies for both scenarios: *copy* scenario and *non-copy* scenario. In the following subsections, all the possible combinations that have sense for both scenarios are explained.

***Non-Copy* Scenario Strategies.** The following combinations have sense for the *non-copy* scenario:

- *Never retain - Never offer* strategy (*NR-NO*): The agents never retain the cases neither offer it to any other agent. Therefore, this is equivalent to a system where the agents do not learn from its experience.
- *Always retain - Never offer* strategy (*AR-NO*): The agent has the opportunity to learn a new case always retains it, but never offer it to the other agents. In this case, every agent works as if learning in isolation. And all the collaborative work is delegated to the *case bartering* process.
- *When-Interesting retain - Never offer* strategy (*WIR-NO*): Equivalent to the previous one, but the agent only retains the case if it is interesting for him.
- *When-Interesting retain - Always offer* strategy (*WIR-AO-non-copy*): In this strategy, the agent A_i that has the opportunity to learn a new case, retains the case only if deemed interesting. If the case is not retained, it is offered to the other agents. Then, as we are in the *non-copy* scenario, the agent has to choose just one of the agents that have answered requesting the case to send only one copy of it. Several strategies can be used to make this selection, but in the experiments this is made randomly.

***Copy* Scenario Strategies.** The *NR-NO*, *AR-NO* and *WIR-NO* strategies are the same than in the *non-copy* scenario. Thus, the only new strategy that can be applied in the *copy* scenario is: *When-Interesting retain - Always offer* strategy (*WIR-AO-copy*) where the agent that has the opportunity to learn a new case, retains the case only if deemed interesting. Then, it is offered to the other agents. The, a copy of the case is sent to each agents that answers requesting a copy. Notice that now this is possible because we are in the *copy* scenario.

There is another combination of policies that generates a new strategy: *Always retain - Always offer* strategy, where the cases are always retained by the agent. And then, offered to the other agents. This is a non interesting strategy although, because all the agents in the system will have access exactly to the same cases and will retain all of them. Therefore, as all the agents will have exactly the same case bases there is no reason to use a multiagent system instead of a single agent that centralizes all the cases.

4 Case Bartering

In the previous section, we have explained the *case retain* process strategies used by the agents. Now, in this section, we will give a brief description of the case bartering process, see [7] for a complete description.

Previous results [6] have shown that agents can obtain better results using the *Committee* collaboration policy than working alone. However, those experiments assumed that every agent had a representative (with respect to the whole collection of cases) sample of cases in its individual case base. When one agent's case base is not representative we say it is biased, and the *Committee* accuracy starts decreasing. Case bartering addresses this problem, and each agent will exchange cases with other agents in order to improve the representativeness (i.e. diminish the bias) of its case base.

4.1 Case Base Bias

The first thing we have to define is the way that the agents measure its case base *bias*. Let be $d_i = \{d_i^1, \ldots, d_i^K\}$ the individual distribution of cases for an agent A_i, where d_i^j is the number of cases with solution $S_j \in \mathcal{K}$ in the the case base of A_i. Now, we can estimate the overall distribution of cases $D = \{D^1, \ldots, D^K\}$ where $D^j = (\sum_{i=1}^n d_i^j)/(\sum_{i=1}^n \sum_{l=1}^K d_i^l)$ is the estimated probability of the class S_i.

To measure how far is the case base C_i of a given agent A_i of being a representative sample of the overall distribution we will define the *Individual Case Base* (*ICB*) bias, as the square distance between the distribution of cases D and the (normalized) individual distribution of cases obtained from d_i:

$$ICB(C_i) = \sum_{k=1}^{K} \left(D^k - \frac{d_i^k}{\sum_{j=1}^K d_i^j} \right)^2$$

It has been empirically shown [7] that when the *ICB* bias is high (and thus the individual case base representativeness is low), the agents using the *Committee* policy obtain lower accuracies.

4.2 Bartering Offers

The way bartering has to reduce the *ICB* bias of a case base is through case interchanging. In order to interchange cases between two agents, they must reach a bartering agreement. Therefore, there must be an offering agent A_i that sends an offer to another agent A_j. Then A_j has to evaluate whether the offer of interchanging cases with A_i is interesting or not, and accept or reject the offer. If the offer is accepted, we say that A_i and A_j have reached a bartering agreement, and they will interchange the cases in the offer. Formally an offer is a tuple $o = \langle A_i, A_j, S_{k_1}, S_{k_2} \rangle$ where A_i is the offering agent, A_j is the receiver of the offer, and S_{k_1} and S_{k_2} are two solution classes, meaning that the agent A_i will send one of its cases with solution S_{k_2} and A_j will send one of its cases with solution S_{k_1}.

The Case Bartering protocols do not force to use some concrete strategy to accept or to send offers, so each agent can have its own strategy. However, in our experiments each agent follow the same strategy.

Let us start with the simpler one. When an agent receives a set of offers, it has to choose which of these offers to accept and which not. In our experiments the agents use the simple rule of accepting every offer that reduces its own *ICB* bias. Thus, we will define the set of interesting offers *Interesting*(O, A_i) of a set of offers O for an agent A_i as those offers that will reduce the *ICB* bias of A_i.

The strategy to make offers in our experiments is slightly more complicated. In [7] the agents used a deterministic strategy to make offers, but for the experiments reported here, we have chosen a probabilistic strategy which obtains better results. Each agent A_i decide which offers to make in the following way: from the set possible solution classes $\mathcal{K}$, each agent choose the set $\mathcal{K}' \subseteq \mathcal{K}$ of those solution classes they are interested in (i.e. those classes that incrementing the number of cases with that solution class, will diminish the *ICB* bias measure). For each class $S_{k_1} \in \mathcal{K}'$, the agent will send one

bartering offer to an agent $A_j \in \mathcal{A}$. This agent A_j is chosen probabilistically, and the probability of an agent to be chosen as A_j is a function of the number of cases that the agent have with solution class S_{k_1} (as more cases, higher probability). Now, the agent A_i has to decide which solution class $S_{k_2} \in \mathcal{K}$ will offer to A_j in change for the class S_{k_1}. The solution class $S_{k_2} \in \mathcal{K}''$ (where $\mathcal{K}'' \subseteq \mathcal{K}$ is the subset of solution classes that decreasing the number of cases with that solution class, will diminish the *ICB* bias measure) is also chosen probabilistically, and the probability of each solution class to be chosen is a function of the number of cases that A_i has of that solution class (as more cases, higher probability).

4.3 Case Bartering Protocol

Using the previous strategies, two different protocols for Case Bartering have been experimented. The first one is called the *Simultaneous Case Bartering Protocol*, and the second one the *Token-Passing Case Bartering Protocol*. However, since the experiments presented in this paper use only the second one, only the *Token-Passing* protocol is going to be explained here.

When an agent member of the $\mathcal{M}$AC wants to enter in the bartering process, it sends an initiating message to all the other agents in the $\mathcal{M}$AC. Then all the other agents answer whether or not they enter the bargaining process. This initiating message contains the parameters for bartering: a parameter t_O, corresponding to the time period that the agents have to make offers; a parameter t_A, corresponding to the time period that the agents have to send the accept messages; the number n of agents taking part in the bartering, and R_{max}, the maximum number of bartering rounds that the bartering will have. Once the agents have answered to this initial message, the bartering starts.

The main characteristic of this protocol is the Token-Passing mechanism, so that only the agent who has the Token can make offers to the others.

1. The initiating agent sends a start message containing the the protocol parameters (t_O, t_A, and R_{max}).
2. Each agent broadcasts its local statistics d_i.
3. When all agents have send d_i, they are able to compute the overall distribution estimation D.
4. Each agent computes the *ICB* bias of all the agents taking part in the bartering (including itself), and sorts them in decreasing order. This defines the order in which the Token will be passed through.
5. The agent with higher *ICB* bias is the first to have the Token. So, the initiating agent gives the token to him.
6. The agent who has the Token sends its bartering offers.
7. When the time t_O is reached each agent chooses the subset of accepted offers from the set of received offers from the owner of the token and sends accept messages.
8. When the maximum time t_A is over, all the unaccepted offers are considered as rejected.
9. Each agent broadcasts its new individual distribution d_i.
10. When all agents have send d_i, three different situations may arise:

(a) If there are agents that still haven't owned the token in the current round, the owner of the token gives it to the next agent and the protocol moves to state 6.
(b) If every agent has owned the token once in this round, there have been some interchanged cases and the maximum number of iterations R_{max} still has not been reached, the protocol moves to state 4.
(c) If every agent has owned the token once in this round, but there have been no interchanged cases or the maximum number of iterations R_{max} has been reached, the protocol moves to state 11.

11. If there have been no interchanged cases, the Case Bartering Protocol ends, otherwise the protocol moves to state 4.

Notice that the procotol does not specify when the agents have to barter the cases. It only defines a way to reach bartering agreements. It's a matter of the agents when they will really interchange the cases.

5 Experimental results

In this section we want to compare the classification accuracy of the *Committee* collaboration policy using all the strategies presented in this paper. We also present results concerning case base sizes.

We use the marine sponge identification (classification) problem as our test bed. Sponge classification is interesting because the difficulties arise from the morphological plasticity of the species, and from the incomplete knowledge of many of their biological and cytological features. Moreover, benthology specialists are distributed around the world and they have experience in different benthos that spawn species with different characteristics due to the local habitat conditions.

We have designed an experimental suite with a case base of 280 marine sponges pertaining to three different orders of the *Demospongiae* class (*Astrophorida*, *Hadromerida* and *Axinellida*). In each experimental run the whole collection of cases is divided in two sets, a training set (that contains a 10% of the cases), and a test set (that contains a 90% of the cases). The training set is distributed among the agents, and then incremental learning is performed with the test set. Each problem in the test set arrive randomly to one agent in the $\mathcal{M}$AC. The goal of the agent receiving a problem is to identify the correct biological order given the description of a new sponge. Once an agent has received a problem, the $\mathcal{M}$AC will use the *Committee* collaboration policy to obtain the prediction. Since our experiments are supervised learning ones, after the committee has solved the problem, there is a supervisor that tells the agent receiver of the problem which was the correct solution. After that, the retain policy is applied. In order to test the generality of the strategies, we have tested them using systems with 3, 5 and 8 agents. Each agent apply the nearest neighbor rule to solve the problems. The results presented here are the average of 50 experimental runs.

For experimentation purposes, the agents do not receive the problems randomly. We force biased case bases in every agent by increasing the probability of each agent to receive cases of some classes and decreasing the probability to receive cases of some other classes. This is done both in the training phase and in the test phase. Therefore, each agent will have a biased view of the data.

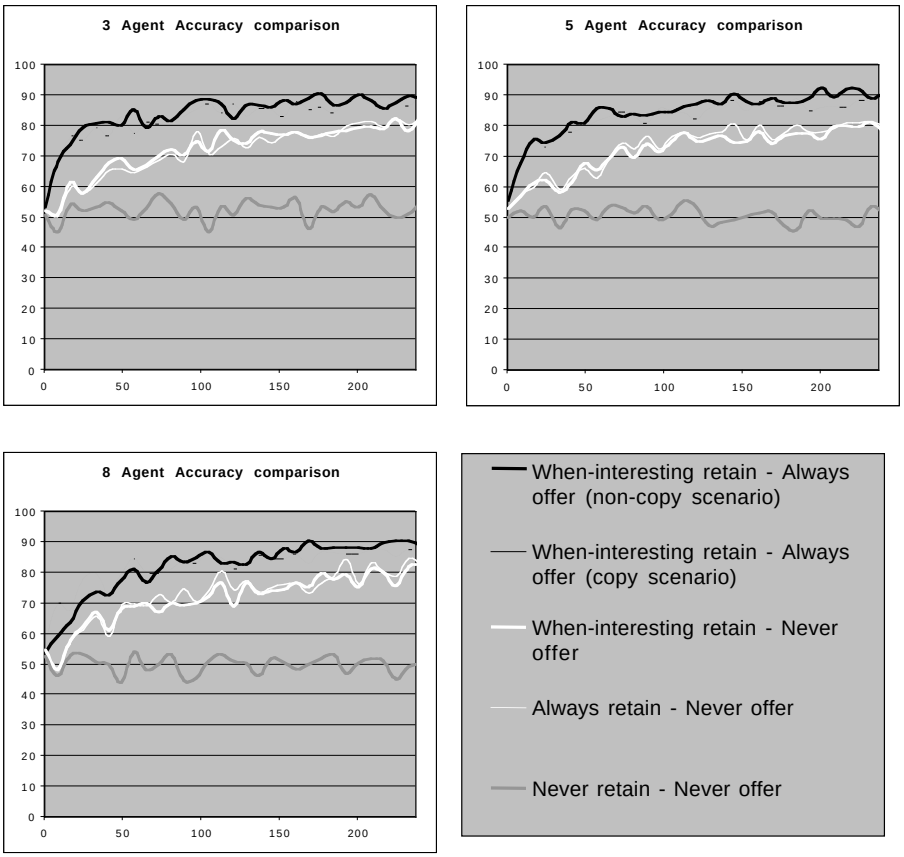

Fig. 1. Accuracy comparison for several configurations without using bartering.

Figure 1 shows the learning curve for several multiagent systems and using several retain strategies and without using bartering. The three charts shown in Figure 1 correspond to multiagent systems composed of 3, 5 and 8 agents respectively. For each multiagent system, 5 strategies have been tested: *NR-NO*, *AR-NO*, *WIR-NO*, *WIR-AO-copy* and *WIR-AO-non-copy*. The figure shows the learning curve for each strategy. The horizontal axis of Figure 1 represents the number of problems that the agents have received of the test set. The baseline for the comparison is the *NR-NO* strategy, where the agents do not retain any cases, and therefore (as we can see in the figure) the agents do not learn, resulting in an horizontal learning curve around an accuracy of 50% in all the multiagent systems. This is because the training set is extremely small, 28 cases. The *Committee* collaboration policy has been proven to obtain results above 88% when the agents have a reasonable number of cases [6].

Considering the other four strategies we can see that, in all the multiagent systems, there are two pairs of strategies with similar learning curves. Specifically, the *AR-NO* and *WIR-NO* have nearly the same learning curve, and therefore we cannot distinguish

	without bartering			*with bartering*		
	3 Agents	*5 Agents*	*8 Agents*	*3 Agents*	*5 Agents*	*8 Agents*
NR-NO	9.33	5.60	3.50	9.33	5.60	3.50
AR-NO	93.33	56.00	35.00	93.33	56.00	35.00
WIR- NO	23.80	14.32	10.70	29.13	19.50	13.64
WIR- AO-copy	58.66	57.42	56.60	59.43	57.42	57.09
WIR- AO-non-copy	45.00	34.42	25.90	44.33	35.14	26.55

Table 1. Average case base size of each agent at the end of the learning process for agents using and not using bartering.

them. They both start from an accuracy of 50% and end with an accuracy around 81%. Therefore, they are significantly better than the *NR-NO* strategy.

The *WIR-NO-copy* and *WIR-NO-non-copy* also have very similar learning curves. Both starting at around 50% in all the scenarios, and arriving to 90% in the case of the *WIR-NO-non-copy* and 88% in the *WIR-NO-copy* also in all the scenarios.

Summarizing we can say that (when the agents do not use case bartering) the strategies that use *When-interesting retain* and *Always retain* policies are not distinghishable in terms of accuracy. The strategies that use the *Always offer* policy (*WIR-AO-copy* and *WIR-AO-non-copy*) obtain higher accuracy than the strategies that use the *Never offer* policy (*AR-NO* and *WIR-NO*). Thus, it is always better for the *Committee* collaboration policy that the agents that receive cases offer them to the other agents; the reason is that these cases perhaps are not interesting to agent receiving the problem, but may be another agent in the system that founds some of those cases interesting.

We can also compare the case base sizes reached after the learning process. The left part of Table 1 shows the average size of each individual case base at the end of the learning process (i.e. when all the 252 cases of the test set have been sent to the agents) when the agents do not use bartering. In all the experiments just 28 cases (the training set) are owned by the agents at the beginning. When the agents use the *NR-NO* strategy, since they do not retain any new cases, they just keep the initial cases. For instance, we can see in the 3 agents scenario, where the agents have in average a case base of 9.33 cases, that 3 times 9.33 is exactly 28, that is the number of cases initially given to them.

Comparing the *AR-NO* strategy with the *WIR-NO* strategy (that achieved undistinguishable accuracies), we can see that the case bases sizes obtained with *WIR-NO* are four times smaller than the case base sizes obtained with *AR-NO* for the 3 and 5 agents scenarios, and about 3.2 times smaller for the 8 agents scenario. Thus, we can conclude that the *WIR-NO* strategy is better than *AR-NO* strategy because achieves the same accuracy but with a smaller case base size.

A similar comparison can be made with the *WIR-AO-copy* and *WIR-AO-non-copy*. Remember that *WIR-AO-non-copy* and *WIR-AO-copy* have a similar learning curve, but *WIR-AO-non-copy* obtains slightly better results (90% vs 88% at the end of the test phase). The case base sizes reached are smaller for the *WIR-AO-non-copy* than for the *WIR-AO-copy* strategy. Thus, *WIR-AO-non-copy* achieves higher accuracy with a smaller case base size. The explanation is that when allowing multiple copies of a case in the system (in *WIN-AO-copy*), we are increasing the correlation between the

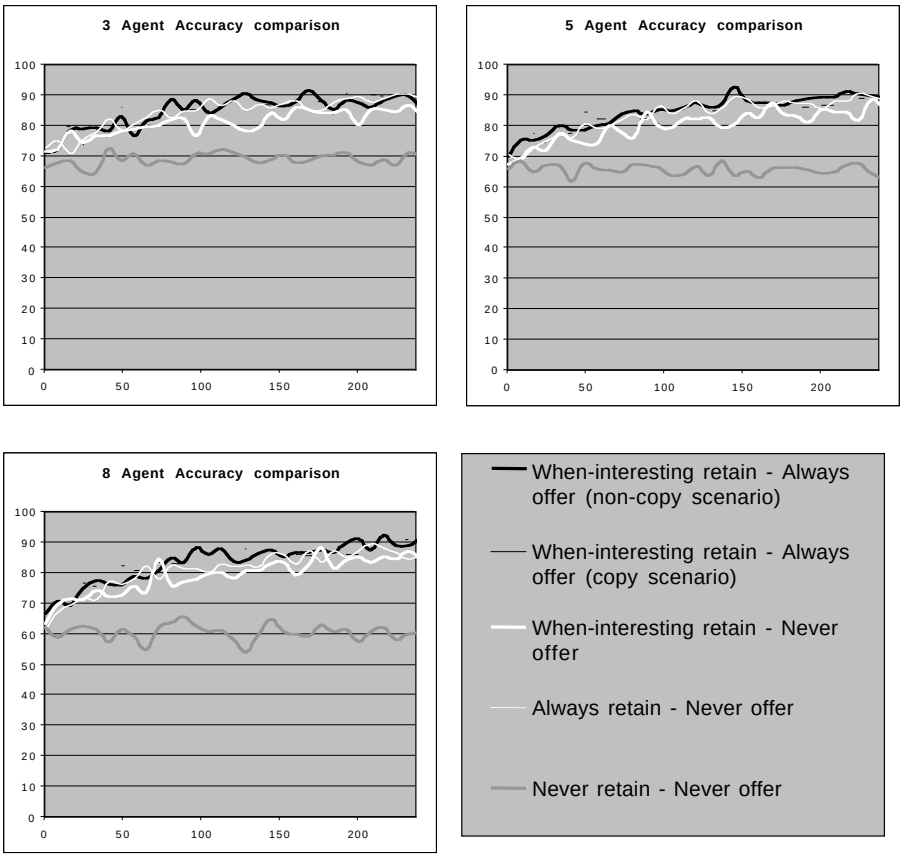

Fig. 2. Accuracy comparison for several configurations using bartering.

case bases of the agents. Moreover, It is known that the combination of uncorrelated classifiers has better results that the combination of correlated ones [4]; increasing the correlation is the cause of *WIR-AO-copy* achieving a lower accuracy than *WIR-AO-non-copy*.

Figure 2 shows exactly the same experiments as in Figure 1, but with agents using bartering. The agents in our experiments perform bartering each 20 cases of the test phase. Figure 2 shows that with the use of bartering, the accuracy of all the different strategies is increased. The *NR-NO* strategy gets boosted from an accuracy of 50% to 70% in the 3 agents scenario and to 67% and 60% in the 5 and 8 agents scenario respectively. Notice that when the agents use the *NR-NO*, no cases are retained and thus their case base sizes are very small, and that just reducing the bias of the individual case bases, the agents obtained a great improvement. This shows the benefits that bartering can provide to the multiagent system.

Figure 2 also shows that the *WIR-AO-copy* and *WIR-AO-non-copy* strategies still achieve the highest accuracies, and that their accuracies are not distinguishable. The

accuracies of the *WIR-NO* and *AR-NO* strategies also improved and are now closer to *WIR-AO-copy* and *WIR-AO-non-copy* than without bartering. Moreover, the *AR-NO* strategy achieves now higher accuracy than *WIR-NO*. The agents retain more cases in the *AR-NO* strategy than in the *WIR-NO*, thus they have more cases to trade with in the bartering process. Therefore, when the agents use bartering, they have an incentive to retain cases because they can later negotiate with them in the bartering process.

The right part of Table 1 shows the average size of each individual case base at the end of the learning process when the agents use bartering. We can see that the case base sizes reached are very similar to the case base sizes reached without bartering. Therefore, with bartering we cannot say that *AR-NO* is better than *WIR-NO* (as happened without bartering), because *AR-NO* achieves higher accuracy but greater case base sizes, and *WIR-NO* has smaller case base size but with a slightly lower accuracy.

Summarizing all the experiments presented (with and without bartering), we can say that using bartering the system always obtains an increased accuracy. We have also seen that the strategies where the agents use the *Always offer* policy also obtains higher accuracies. And that if we let each agent to decide when a case is interesting to be retained (*When-Interesting retain*) instead of retaining every case (*Always retain*), we can reduce significantly the case bases with practically no effect on the accuracy. Finally, we can conclude that *When-Interesting Retain - Always offer* strategy (with no copy) outperforms all the other strategies, since it obtains the higher accuracies with rather small case bases, and that the use of bartering is always beneficial.

6 Related Work

Related work can be divided in two areas: multiple model learning (where the final solution for a problem is obtained through the aggregation of solutions of individual predictors) and case base competence assessment. A general result on multiple model learning [3] demonstrated that if uncorrelated classifiers with error rate lower than 0.5 are combined then the resulting error rate must be lower than the one made by the individual classifiers. However, these methods do not deal with the issue of "partitioned examples" among different classifiers as we do—they rely on aggregating results from multiple classifiers that have access to *all* data. The meta-learning approach in [1] is applied to partitioned data. They experiment with a collection of classifiers which have only a subset of the whole case base and they learn new meta-classifiers whose training data are based on predictions of the collection of (base) classifiers.

Learning from biased data sets is a well known problem, and many solutions have been proposed. Vucetic and Obradovic [10] propose a method based on a bootstrap algorithm to estimate class probabilities in order to improve the classification accuracy. However, their method does not fit our needs, because it requires the availability of the entire test set.

Related work is that of case base competence assessment. We use a very simple measure comparing individual with global distribution of cases; we do not try to assess the areas of competence of (individual) case bases - as proposed by Smyth and McKenna [8]. This work focuses on finding groups of cases that are competent.

7 Conclusions and Future Work

We have presented a framework for cooperative Case-Based Reasoning in multiagent systems, where agents can cooperate in order to improve its performance. We have presented also that a market mechanism (bartering) can help the agents and improve the overall performance as well as the individual performance of the agents. The agent autonomy is maintained because all the agents are free, because if an agent do not want to take part in the bartering, he just have to reject all the offers and do not make any one.

We have also shown the problem arising when data is distributed over a collection of agents than can have a skewed view of the world (the individual bias). Case bartering shows that the problems derived from distributed data over a collection of agents can be solved using a market-oriented approach. We have presented explicit strategies for the agents to accumulate experience (retaining cases) and to share this experience with the other agents in the system. The outcome of this experience sharing is an improvement of the overall performance of the system (i.e. higher accuracy). But further research is needed in order to find better strategies that allow the agents obtain the higher accuracy with the smallest case base size.

Acknowledgements

The authors thank Josep-Lluís Arcos of the IIIA-CSIC for its support and for the development of the Noos agent platform. Support for this work came from CIRIT FI/FAP 2001 grant and projects TIC2000-1414 "eInstitutor" and IST-1999-19005 "IBROW".

References

[1] Philip K. Chan and Salvatore J. Stolfo. A comparative evaluation of voting and meta-learning on partitioned data. In *Proc. 12th International Conference on Machine Learning*, pages 90–98. Morgan Kaufmann, 1995.

[2] G. W. Gates. The reduced nearest neighbor rule. *IEEE Transactions on Information Theory*, 18:431–433, 1972.

[3] L. K. Hansen and P. Salamon. Neural networks ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (12):993–1001, 1990.

[4] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 231–238. The MIT Press, 1995.

[5] David B. Leake and Raja Sooriamurthi. When two case bases are better than one: Exploiting multiple case bases. In *ICCBR*, pages 321–335, 2001.

[6] S. Ontañón and E. Plaza. Learning when to collaborate among learning agents. In *12th European Conference on Machine Learning*, 2001.

[7] S. Ontañón and E. Plaza. A bartering aproach to improve multiagent learning. In *1st International Joint Conference in Autonomous Agents and Multiagent Systems*, 2002.

[8] B. Smyth and E. McKenna. Modelling the competence of case-bases. In *EWCBR*, pages 208–220, 1998.

[9] Barry Smyth and Mark T. Keane. Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In *IJCAI*, pages 377–383, 1995.

[10] S. Vucetic and Z. Obradovic. Classification on data with biased class distribution. In *12th European Conference on Machine Learning*, 2001.