

Expertness Based Cooperative Q-Learning

Majid Nili Ahmadabadi and Masoud Asadpour

Abstract—By using other agents' experiences and knowledge, a learning agent may learn faster, make fewer mistakes, and create some rules for unseen situations. These benefits would be gained if the learning agent can extract proper rules out of the other agents' knowledge for its own requirements. One possible way to do this is to have the learner assign some expertness values (intelligence level values) to the other agents and use their knowledge accordingly.

In this paper, some criteria to measure the expertness of the reinforcement learning agents are introduced. Also, a new cooperative learning method, called weighted strategy sharing (WSS) is presented. In this method, each agent measures the expertness of its teammates and assigns a weight to their knowledge and learns from them accordingly. The presented methods are tested on two Hunter-Prey systems.

We consider that the agents are all learning from each other and compare them with those who cooperate only with the more expert ones. Also, the effect of the communication noise, as a source of uncertainty, on the cooperative learning method is studied. Moreover, the Q-table of one of the cooperative agents is changed randomly and its effects on the presented methods are examined.

Index Terms—Cooperative learning, expertness, multi-agent systems, Q-learning.

I. INTRODUCTION

IN HUMAN societies, it can be observed that, the more one learns from another's experiences, a higher chance he has to succeed. In fact, people take advice, consult with each other, receive unprocessed information, and observe others to learn from their activities and experiences. In other words, people cooperate to learn.

In almost all of the present artificial multi-agent teams, agents learn individually and cooperative learning has not been deeply investigated. However, similar to human beings, agents are not required to learn everything from their own experiences (see Fig. 1). In fact, due to having more knowledge and information acquisition resources, cooperation in learning in a multi-agent system may result in a higher efficiency compared to individual learning [17]. Improvements in learning have been shown in different researches even when simple cooperative learning methods are used [30].

As the learner agents are not capable of representing their knowledge properly and observing the other agents requires a high level of sensing and intelligence, the agents cannot advise each other or automatically learn by passively observing

the other agents. Therefore, they are required to communicate their experiences and information.

In almost all of the multi-agent learning published papers, cooperation is unidirectional between a fixed trainer agent and a learner. However, all agents may learn something from each other provided that, some proper measures and methods are implemented.

One of the most important issues for a learner agent is the assessment of the behavior and the intelligence level of the other agents. In addition, the learner agent must assign a relative weight to the other agents' knowledge and use it accordingly.

In general, these three issues are very complex and need careful attention. Therefore, in this paper, as well as in [22], attention has been paid to find some solutions for homogeneous, independent, and cooperative Q-learning agents.

In [22], a new cooperative learning strategy, called weighted strategy sharing (WSS) and some expertness measuring methods are introduced. In that paper, it is assumed that the learner agents cooperate only with the more expert agents. Also, it is assumed that, the communication is perfect and all of the agents are reliable. In this paper, it is considered that all of the agents could learn from each other and the obtained results are compared with the results of the algorithm presented in [22]. In addition, effects of the communication noise as a source of uncertainty on the cooperative learning are studied. Moreover, the Q-table of one of the cooperative agents is changed randomly and its effects on the presented method are examined.

Related researches are reviewed in the next section. Then, WSS is briefly introduced and some expertness measures are presented. Also, some weight assigning methods are established. WSS, the effects of implementing the expertness measures, and the role of weight assigning methods are tested in the fourth section. In that section, effects of uncertainty and wrong knowledge are also studied. A conclusion and some directions for future research are given in the last section.

II. RELATED RESEARCHES

Samuel [26] used the competitive Learning algorithm to train a checker game player. In his method, the cooperator agent acts as an enemy or an evaluator and tries to find the weak points of the learned strategy. Hu and Wellman [12] proposed a framework for multi-agent Q-learning when the competitor agents have incomplete information about other agents' payoff functions and state transition probabilities.

In the ant colony system [6], some ants learn to solve the traveling salesman problem by nonverbal communication through the pheromones on the edges of a graph.

Imitation [16] is one of the cooperative learning methods. In this method, the learners watch the actions of a teacher, learn

Manuscript received September 28, 2000; revised September 9, 2001. This paper was recommended by Associate Editor A. Bensaid and Editor L. O. Hall.

M. N. Ahmadabadi is with the Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran, and also with the Intelligent Systems Research Center, Institute for Studies on Theoretical Physics and Mathematics (IPM), Tehran, Iran (e-mail: mnili@ut.ac.ir).

M. Asadpour is with the Intelligent Systems Research Center, Institute for Studies on Theoretical Physics and Mathematics (IPM), Tehran, Iran (e-mail: asadpur@karun.ipm.ac.ir).

Publisher Item Identifier S 1083-4419(02)00123-1.

them, and repeat these actions in similar situations. This method does not affect the teacher performance [3] and the learning process is unidirectional. For example, in [16], a robot perceives a human doing a simple assembly task and learns to repeat it in different environments. Hayes and Demiris [10] built a robotic system in which a learner robot imitates a trainer moving in a maze and learns to escape from it.

Yamaguchi and others [33] developed a robotic imitation system to train Q-learning ball-pusher robots. In this system, agents learn individually and imitate each other using simple mimetism, conditional mimetism, and adaptive mimetism methods.

In simple mimetism [35], all agents imitate each other when they are neighbors. In this method, two neighbors may wait for each other forever. This problem is solved by applying conditional mimetism [35]. In conditional mimetism, only the low performance agent (performance is measured based on the sum of the rewards and punishments received in past n actions) imitates the other one. Adaptive mimetism [33], [34] is similar to conditional mimetism, but the imitation rate is adjusted based on the performance difference of two neighbor robots.

The robots cooperate to learn when they share their sensory data and play the role of scout for each other [30]. Episode sharing [14], [30] can be used to communicate the state, action, and reward triples between the reinforcement learners. Tan showed that, sharing episodes with an expert agent could improve the group learning significantly [30]. In [15], the state, action, and value pairs are communicated among the agents. No measure is used to evaluate the received rules by the learners.

In [4], a blackboard is used as a global information system for improving the individual learning and coordination in a multi-agent team.

In the collective memory method, learners put learned strategy or experienced episodes on a shared memory [8] or they have a single memory and update it cooperatively [30].

A cooperative ensemble learning system [17] has been developed as a new method in neural network (NN) ensembles [1], [11], [17], [25], [27]. In these studies, a linear combination of the concurrent learning NN's outputs are used as a feedback to add a new penalty term to the error function of each network.

Provost and Hennessy [24] developed a cooperative distributed learning system for systems with huge training sets. The training set is divided into k smaller training subsets and k rule-learning agents learn the local rules. The rules are transmitted to the other agents for evaluation; if the rule satisfies the evaluation criteria, it is accepted as a global one.

High attention is paid to the advice taking method in recent years [9], [13], [19], [21], [23]. Mostow [21] wrote a program that accepts high-level advices to play the card game. Gordon and Subramanian [9] developed a system that translates high-level advices into the concrete actions and evaluates them by genetic algorithm (GA).

Maclin and Shavlik [18] used the advice taking scheme to help a connectionist reinforcement learner. The learner accepts advice in the form of a simple computer program, compiles it, represents the advice in some NNs and adds them to its current network.

In most of the reviewed researches, cooperation is unidirectional from a prespecified trainer to a preselected learner

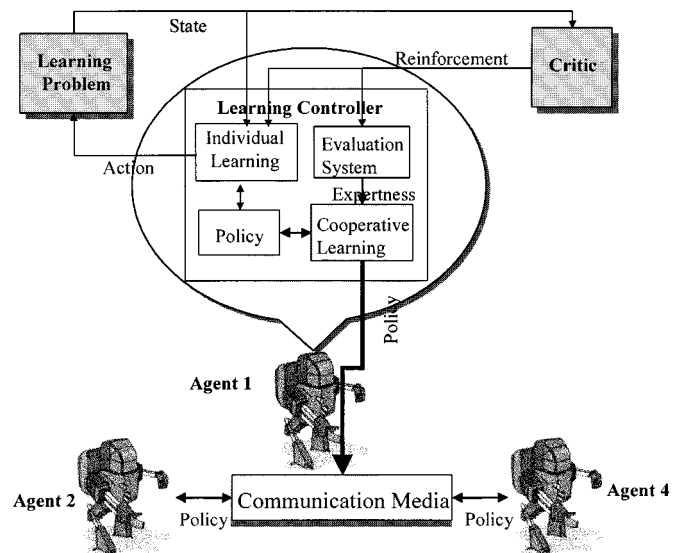


Fig. 1. Weighted strategy sharing (WSS) architecture.

agent. In the real world, cooperative learning is bidirectional and all of the agents learn something from each other (even from the nonexpert ones). In the strategy sharing method [30], each Q-learning agent learns from all of its teammates. The agents learn individually and at some special instants, each agent gathers the Q-tables of the other agents and takes the average of the tables as its own new strategy. In this system, the agents do not have the ability to find good teachers. It seems that, simple averaging of the Q-tables is nonoptimal when the agents have different skills and experiences. Additionally, the Q-tables of the agents become equal after each cooperation step. This decreases the agents adaptability to the environment changes [33].

To overcome the described problems, a new strategy sharing method based on the expertness level of the other agents was proposed in [22].

III. WSS METHOD

In the WSS method [22] (Fig. 1), it is assumed that n homogeneous one-step Q-learning agents [28], [29], [31], [32] learn in some distinct environments and no hidden state is produced [7].

The agents learn in two modes: individual learning mode and cooperative learning mode (see Algorithm 1). At first, all of the agents are in individual learning mode. Agent i executes t_i learning trials. Each learning trial starts from a random state and ends when the agent reaches the goal. After a specified number of individual trials, all agents switch to cooperative learning mode.

Algorithm 1. Weighted Strategy Sharing

Algorithm for agent A_i

- (1) Initialize
- (2) while not End Of Learning do
- (3) begin
- (4) If In Individual Learning Mode then
- (5) begin Individual Learning

```

(6)  $x_i := \text{Find Current State } ()$ 
(7)  $a_i := \text{Select Action } ()$ 
(8)  $\text{Do Action } (a_i)$ 
(9)  $r_i := \text{Get Reward } ()$ 
(10)  $y_i := \text{Go To Next State } ()$ 
(11)  $V(y_i) := \text{Max}_{b \in \text{actions}} Q(y_i, b)$ 
(12)  $Q_i^{\text{new}}(x_i, a_i) := (1 - \beta_i)Q_i^{\text{old}}(x_i, a_i) + \beta_i(r_i + \gamma_i V(y_i))$ 
(13)  $e_i := \text{Update Expertness } (r_i)$ 
(14) end
(15) else Cooperative Learning
(16) begin
(17) for  $j := 1$  to  $n$  do
(18)  $e_j := \text{Get Expertness } (A_j)$ 
(19)  $Q_i^{\text{new}} := 0$ 
(20) for  $j := 1$  to  $n$  do
(21) begin
(22)  $W_{ij} := \text{Compute Weights } (i, j, e_1 \dots e_n)$ 
(23)  $Q_j^{\text{old}} := \text{Get } Q(A_j)$ 
(24)  $Q_i^{\text{new}} := Q_i^{\text{new}} + W_{ij} * Q_j^{\text{old}}$ 
(25) end
(26) end
(27) end

```

In cooperative learning mode, each learning agent assigns some weights to the other agents' Q-tables with respect to their relative expertness. Then, each agent takes the weighted average of the others' Q-tables and uses the resulted table as its new Q-table¹

$$Q_i^{\text{new}} \leftarrow \sum_{j=1}^n (W_{ij} * Q_j^{\text{old}}). \quad (1)$$

A. Some Expertness Criteria

In the WSS method, W_{ij} is a measure of agent i reliance on the knowledge and the experiences of agent j . Here we argue that this weight is a function of the agents relative expertness.

In the strategy sharing method [30], expertness of the agents are assumed to be equal. Nicolas Meuleau [20] used the user judgment for specifying the expert agent. This method requires continuous human supervision.

In [2], different but fixed expertness values are assumed for the agents. However, differences in the expertness values may change during the learning process and are not constant.

Yamaguchi *et al.* [33] specified the expert agents by means of their successes and failures during current n moves and considered the expertness criterion as an algebraic sum of the reinforcement signals in that time interval. This means that more successes and fewer failures are considered a sign of a higher degree of expertness. This expertness measuring method is not optimal in some situations.

For example, the agent that has faced many failures has some useful knowledge to be learned from it. In other words, it is possible that this agent does not know the ways arriving at the goal,

¹Multiplication (*) and summation (+) operators must be specified based on the knowledge representation method.

but it is aware of those not leading to its target and can avoid them. Also, an agent at the beginning of its learning process is less expert than those learned for a longer time and naturally has confronted more failures.

Considering the discussions, six expertness measures are introduced. These measures include the following.

1) **Normal (Nrm)**: An algebraic sum of the reinforcement signals

$$e_i^{\text{Nrm}} = \sum_{t=1}^{\text{now}} r_i(t). \quad (2)$$

2) **Absolute (Abs)**: A sum of the absolute value of the reinforcement signals

$$e_i^{\text{Abs}} = \sum_{t=1}^{\text{now}} |r_i(t)|. \quad (3)$$

3) **Positive (P)**: A sum of the positive reinforcement signals

$$e_i^{\text{P}} = \sum_{t=1}^{\text{now}} r_i^+(t)$$

$$r_i^+(t) = \begin{cases} 0, & \text{if } r_i(t) \leq 0 \\ r_i(t), & \text{otherwise.} \end{cases} \quad (4)$$

4) **Negative (N)**: A sum of the absolute value of the negative reinforcement signals

$$e_i^{\text{N}} = \sum_{t=1}^{\text{now}} r_i^-(t)$$

$$r_i^-(t) = \begin{cases} 0, & \text{if } r_i(t) > 0 \\ -r_i(t), & \text{otherwise.} \end{cases} \quad (5)$$

5) **Gradient (G)**: Changes in the received reinforcement signals since the last cooperation time

$$e_i^{\text{G}} = \sum_{t=c}^{\text{now}} r_i(t) \quad (6)$$

where c is the start time of the individual learning mode.

6) **Average Move (AM)**: A reverse number of moves each agent does to reach the goal

$$e_i^{\text{AM}} = \left(\sum_{\text{trial}=1}^{n_{\text{trial}}} m_i(\text{trial}) / n_{\text{trial}} \right)^{-1} \quad (7)$$

where trial is the trial number, n_{trial} is the current trial, and $m_i(\text{trial})$ is the number of moves that each agent has done to reach the goal.

Nrm criterion gives more credit to those who have more successes and fewer failures. Abs considers both rewards and punishments as a sign of being experienced. P disregards experiences not resulted in achieving the goal and considers the successful experiences only. N formula looks at unsuccessful tries only and assigns a higher expertness value to those experiencing more failures. AM is an indirect way to measure the expertness and considers the average number of actions the agent does to reach the goal. G looks at the trend of improvement in the agent performance in its recent actions and does not look directly at its

past experiences. It is noteworthy that some of these expertness measures are similar to those used by the human beings.

B. Weight Assigning Mechanisms

1) *Learning From All (LA)*: It can be said that all agents have some valuable knowledge to be learned. When using all agents' knowledge, the simplest formula to assign weight to agent j knowledge by learner i could be

$$W_{ij} = \frac{e_j}{n} \quad (8)$$

$$\sum_{k=1}^n e_k$$

where n is the number of the agents and e_k is the amount of the expertness of agent k . In this method, effects of agent j knowledge on all learners are equal, i.e., $W_{1j} = W_{2j} = \dots = W_{nj}$. Also all of Q-tables become homogeneous after each cooperation step.

2) *Learning From All With Positive Weights (LAP)*: If $e_{\min} = \min\{e_k | k = 1 \dots n\}$ and $c > 0$ is a constant, then $e_j - e_{\min} + c > 0$. So, the following weight assigning method can be introduced:

$$W_{ij} = \frac{e_j - e_{\min} + c}{n} > 0, \quad (9)$$

$$\sum_{k=1}^n (e_k - e_{\min} + c)$$

The weight of the least expert agent is

$$W_{i,\min} = \frac{c}{n} \quad (10)$$

$$\sum_{k=1}^n (e_k - e_{\min} + c)$$

If $c \rightarrow \infty$, then $W_{ij} = 1/n$ and WSS converges to SA.

3) *Learning From Experts (LE)*: To decrease the amount of communication required to exchange the Q-tables, the learner may use only the Q-tables of the more expert agents. Learner i assigns the weights based on its expertness difference with the more expert agents using the following formula:

$$W_{ij} = \begin{cases} 1 - \alpha_i, & \text{if } i = j \\ \alpha_i \frac{e_j - e_i}{n} & \text{if } e_j > e_i \\ \sum_{k=1}^n (e_k - e_i) & \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where α_i is the impressibility factor and indicates how much each agent relies on the others knowledge. Partial weights of the others knowledge are zero if they are less expert than the learner agent i . Substituting this equation in the weighted averaging formula results in

$$Q_i^{new} \leftarrow (1 - \alpha_i) * Q_i^{old} + \alpha_i * \sum_{j \in \text{Exprt}(i)} \left(\frac{e_j - e_i}{n} * Q_j^{old} \right) \sum_{k=1}^n (e_k - e_i)$$

$$\text{Exprt}(i) = \{j | e_j > e_i\} \quad (12)$$

where $\text{Exprt}(i)$ is the set of the agents that are more expert than agent i . As the WSS equation mathematically resembles the reinforcement learning formula, the mathematical approaches

taken to the reinforcement learning method are also applicable to the WSS technique.

C. Special Cells Communication

Two mechanisms called positive only (PO) and negative only (NO) are also introduced to reduce interagent communication. In PO, the learner uses the other agents Q-table cells having positive value and assumes that the other cells are zero. In PO, the expertness values of the agents are measured by positive criterion.

In NO, agents only send negative-value cells of their Q-tables to the others and their expertness is measured by negative criterion.

IV. SIMULATION ON HUNTER-PREY PROBLEM

The hunter-prey problem [30] is one of the classical testbeds to study and compare different learning processes (Fig. 2). In this paper, there are three hunters independently searching a 10×10 environment to capture a prey agent. The moving speed of the hunters and the prey is positive and less than 1 and 0.5 units, respectively. The prey is captured when its distance to the hunter is less than 0.5 units. Upon capturing the prey, the hunter receives $+R$ reward and $-P$ punishment otherwise.

Each agent has a visual field to locate the other agents. The visual field of the hunters and the prey are two and three, respectively. The hunter states are specified with respect to the prey position (x, y) in its local coordinate frame. If the prey is not inside its visual field, a default state is considered. The hunters actions consist of rotation and velocity change: $a = (v, \theta)$. The distance, the velocity difference, and the angle difference of the hunter and the prey are divided into sections of one distance unit, 0.5 velocity unit, and 45° , respectively.

For complicating the learning problem and in order to show the differences in efficiency of the cooperative learning algorithms more clearly, a simple and a complicated version of the hunter-prey problem are used. In the simple case, similar to the other researches, the prey moves randomly. In the other case, the prey moves based on the potential field model and escapes from the hunter. We call this agent the intelligent prey.

In the potential field model, four walls around the environment, the prey, and the hunter are assumed to be electropositive materials and repulse each other. The repulsive force of the hunter is considered 1.5 times that of the wall. The hunter and the prey are modeled as spot loads and the walls as linear ones.

To create some agents with different expertnesses, the hunter agents have different learning times (t_i). The first hunter learns six trials, then the second one is permitted to do three trials, and the last hunter does one learning trial. The total number of individual learning trials is 1000 and the cooperation time is after every 50 individual learning trials of all hunters together. The reward and the punishment signals are one of the following six pairs:

$$(10, -0.01), (10, -0.1), (10, -1), \\ (5, -0.01), (5, -0.1), (5, -1).$$

An individual learning trial ends when the hunter captures the prey. The one-step Q-learning parameters are set to $\beta =$

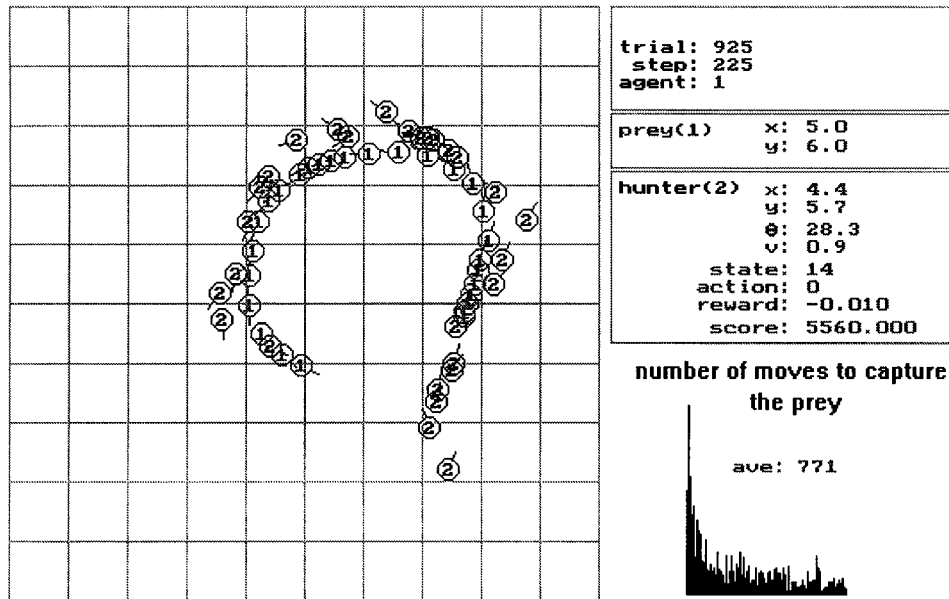


Fig. 2. Hunter-Prey problem simulator.

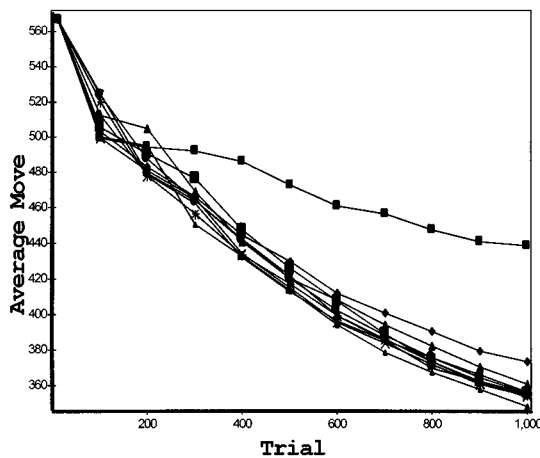


Fig. 3. Average number of moves in random-prey and equal experience case for LE.

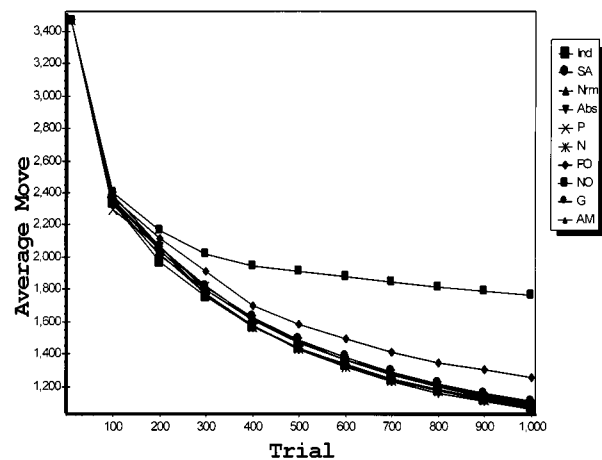


Fig. 4. Average number of moves in intelligent-prey and equal experience case for LE.

0.01, $\gamma = 0.9$, and $T = 0.4$. The Q-table values are initialized to zero and all agents have $\alpha_i = 0.9$. Also, for the trial n , the average number of the hunter actions to capture the prey over the past n trials is measured.

Four types of experiments are performed to check the effectiveness of the WSS method. In the first and the second experiments, agents have equal and different learning trials, respectively.

The last two sets of experiments are designed to study the sensitivity of WSS to uncertainties in the agents' knowledge. The third experiments are performed in the presence of noise in the communication medium. One of the agent's Q-table is filled by incorrect and random data in the fourth experiments.

A. Equal Experiences

In this set of experiments, all of the agents have equal chances for individual learning. Figs. 3 and 4 show the average number

of moves to reach the goal with the six prescribed expertness measures in the random and intelligent prey cases using LE weighting mechanism. Similar graphs are obtained using LA and LAP methods. Table I shows the improvement percentage of these methods over independent learning, i.e., without cooperation in learning.

In the random-prey case, Nrm, AM, P, N, and G methods sometimes have small positive effects on the learning, but in the intelligent-prey case, all of the methods have small negative or approximately zero influence on the learning. So, cooperative learning is not effective in this case.

As expected, in equal experience experiments, the weighted strategy sharing converges to the simple averaging and almost all of the graphs, especially in the intelligent-prey cases, approximately show the same results except for the PO and NO.

PO and NO have significant negative effects on the learning (see improvement percents in Table I). In these two methods, each learner uses only one part of the other agents knowledge.

TABLE I
IMPROVEMENT PERCENTS IN EQUAL EXPERIENCE CASE

| | | SA | Nrm | Abs | P | N | PO | NO | G | AM |
|---------------|-----|-------|-------|-------|-------|--------|--------|--------|-------|-------|
| random | LA | -0.05 | 1.88 | -1.03 | 0.8 | -1.6 | -20.6 | -26.03 | 0.14 | 1.6 |
| | LAP | -0.05 | -3.66 | -0.05 | -1.36 | -0.38 | -23.08 | -25.19 | -1.6 | -0.05 |
| prey | LE | -0.05 | -1.36 | -0.14 | 0.38 | 0.24 | -4.97 | -23.36 | 0.70 | 2.11 |
| | LA | -3.14 | -3.43 | -4.3 | -1.34 | -3.21 | -42.3 | -85.07 | -2.97 | -2.9 |
| prey | LAP | -3.14 | -6.76 | -1.67 | -0.67 | -1.84 | -42.49 | -84.04 | -7.3 | -3.17 |
| | LE | -3.14 | -3.34 | 0.94 | -1.95 | -0.156 | -17.66 | -65.1 | -1.26 | -0.45 |

In PO, failures of the other agents are ignored, and, in NO, successful actions of others are not considered. A study of the Q-tables of the learning hunters shows that most cells of the Q-tables have negative values and, consequently, a small number of the cells are transferred in PO. But, in all cases, PO has better results than NO and the successful actions have been more important. This is due to two facts. First, as the number of the successful actions is small, each learner possibly cannot find a good action without cooperation with the other learners. Secondly, in Boltzmann probability distribution, the selection probability of a positive-value action (hereafter called a positive action) is increased exponentially when its value is increased, but, the selection probability difference of two negative-value (negative actions) actions is small; even their values are considerably different.

In the PO and NO methods, unlike other methods, the other agents negative and positive actions values are considered zero (more and less than the real values, respectively). As a result, the differences of the positive and the negative actions are decreased. Therefore, the agents make more mistakes in their action selection.

Expertness measuring methods approximately have the same or better results when the weight assigning mechanism is changed from LA to LAP, except for Nrm and G. A problem with the LA method is that, if $e_i < 0$, e.g., in gradient and normal methods, according to (8) we may have

$$\sum_{k=1}^n e_k = 0$$

or $W_{ij} < 0$. Also, if $e_i < e_j < 0$ we have $W_{si} > W_{sj}$, which contradicts the main idea of the expertness-based weight assignment method.

More studies on Nrm and G methods showed that, because of receiving many punishments, the expertness values of the agents become negative. Therefore, the more experienced agents get more negative expertness values, but in LA, the numerator and denominator are both negative. Consequently, the more expert agents are assigned more positive weights.

On the other hand, LAP prevents a division by zero and wrong weight assignment problems by shifting the agents expertness values. So, the more expert agents (having negative expertness value with respect to Nrm and G) get smaller weights compared to the less expert agents. Therefore, LA works better than LAP for Nrm and G measures.

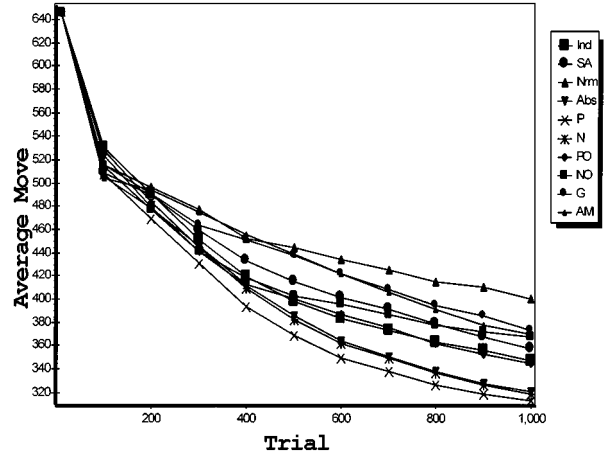


Fig. 5. Average number of moves in random-prey and different experiences case for LE.

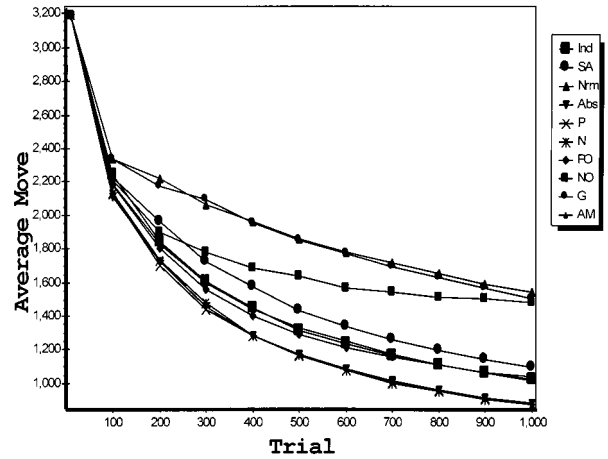














Fig. 6. Average number of moves in intelligent-prey and different experiences case for LE.

Another disadvantage of LA and LAP is that $W_{1j} = W_{2j} = \dots = W_{nj}$ and the Q-tables become the same after cooperation steps. This feature decreases the adaptability of a multi-agent system [33].

Comparing simple averaging strategy sharing (SA) [30] and individual learning shows that, on average, the SA method has no positive effects on learning (Table I). More detailed studies showed that, in experiments with high convergence rate, e.g., when the rewards and punishments were $(10, -1)$ or $(5, -1)$, SA

TABLE II
IMPROVEMENT PERCENTS IN DIFFERENT EXPERIENCE CASE

| | | SA | Nrm | Abs | P | N | PO | NO | G | AM |
|--------------------|-----|-------|--------|------|-------|-------|-------|--------|--------|-------|
| Random | LA | -2.74 | 9.98 | 5.71 | 3.84 | 4.08 | -11.2 | -26.73 | 3.36 | 0.34 |
| Prey | LAP | -2.74 | -8.64 | 7.82 | 6.57 | 8.78 | -10.8 | -25.24 | -8.16 | -2.26 |
| | LE | -2.74 | -15.3 | 7.58 | 9.93 | 8.30 | 0.85 | -5.78 | -7.49 | -6.29 |
| Intelligent | LA | -7.57 | 4.82 | 3.91 | 4.22 | 3.57 | -30.3 | -89.44 | 5.02 | -3.29 |
| Prey | LAP | -7.57 | -33.67 | 9.98 | 10.01 | 7.88 | -29.8 | -88.32 | -29.67 | -7.72 |
| | LE | -7.57 | -51.15 | 13.9 | 14.24 | 14.44 | -1.68 | -44.84 | -47.49 | -0.65 |

| Reward and Punishment at beginning of learning | Abs | P | N | Nrm |
|--|---|---|--|---|
| Rewards >> Punishments |  |  |  |  |
| Rewards << Punishments |  |  |  |  |
| Rewards = Punishments |  |  |  |  |




 Very Good
  Very Bad
  Good

Fig. 7. Summary of a comparison among Abs, P, N, and Nrm.

caused little improvement in the group learning. Also, when the learning problem became simpler, the results improved gradually, e.g., SA has better results in the random-prey case.

B. Different Experiences

In these experiments, the hunters perform a different number of learning trials and, as a result, have different experiences. Figs. 5 and 6 show the average number of moves to hunt, using six prescribed reinforcement functions and an LE weighting mechanism in random- and intelligent-prey cases. Similar graphs are obtained for LAP and LA. Table II shows the average improvement percentage relative to individual learning.

It is the same with the previous section results; PO and NO have negative effects on the learning. The results also show that SA has a worse performance compared with the performance of the equal experience case. The reason for such an outcome is that the method assigns nonoptimal equal weights to the agents having different expertnesses.

AM has a negative or a very small effect. In the AM method, due to using the $1/m_i$ function, the difference of the agents' expertness values is small, even when they have a considerably different number of moves.

Nrm and G methods have a positive effect on the learning in the LA case but are noneffective in LAP and LE and have the worst results in these two cases. As explained in the previous section, the wrong expertness assignment to the agents is the main reason for this behavior. Considering the presented expertness computation formula, Nrm and G become equivalent when

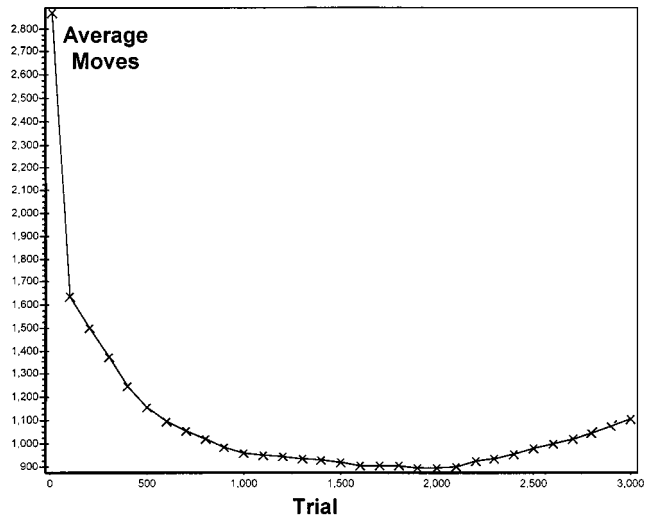
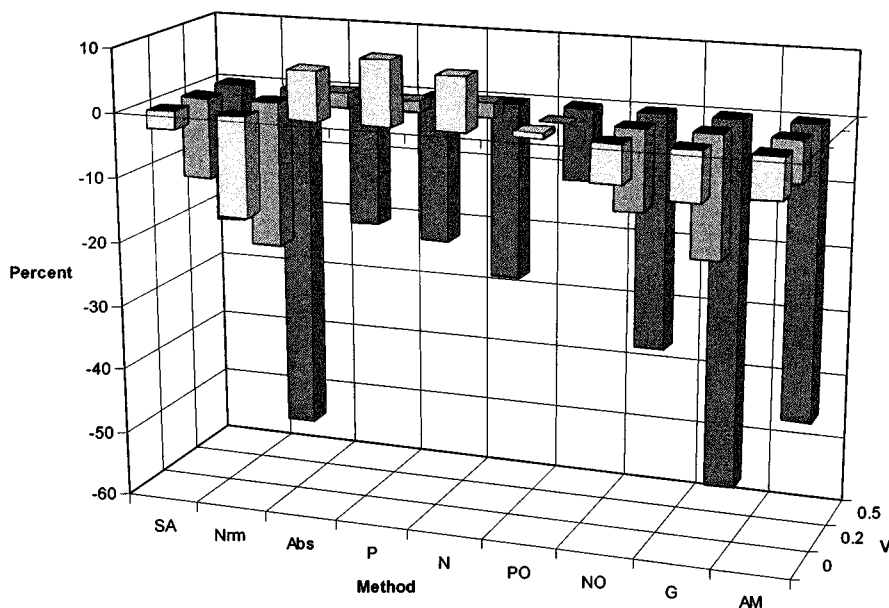


Fig. 8. Divergence of N after temporary convergence.

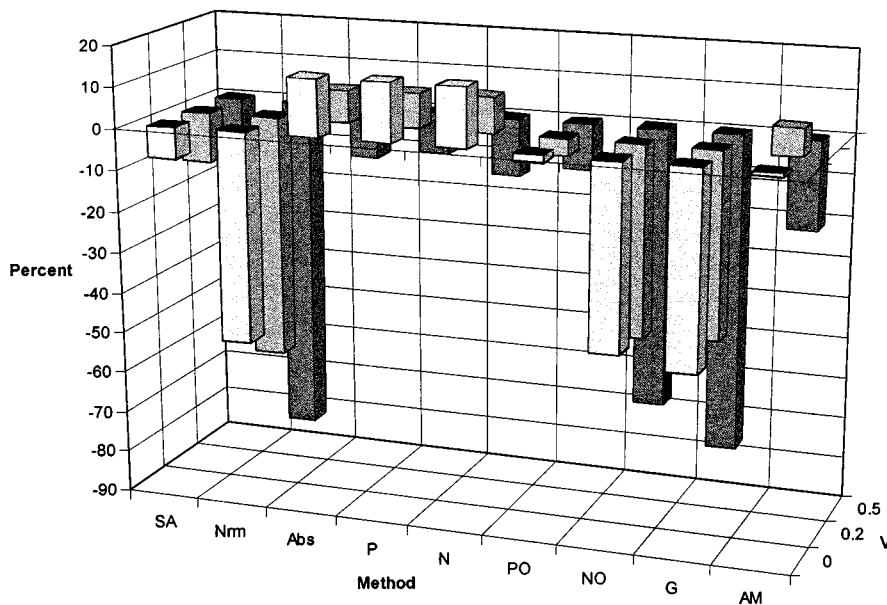
the learning time is increased. This claim is verified in the intelligent-prey case where the learning time is long and the results of Nrm and G are very close.

Table II shows that Abs, P, and N measures help the learning process in all of the cases. P has the best results in the LE case for the random-prey, but N criterion gives the best performance in the LE case for the intelligent-prey. The reason is that, since



| | SA | Nrm | Abs | P | N | PO | NO | G | AM |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | -2.74 | -15.3 | 7.58 | 9.93 | 8.3 | 0.85 | -5.78 | -7.49 | -6.29 |
| 0.2 | -12.7 | -22.5 | 2.3 | 1.7 | 2.3 | 0 | -12.4 | -18.7 | -6.3 |
| 0.5 | -21.9 | -54.5 | -20.7 | -22.5 | -27.4 | -11.2 | -36.6 | -57.1 | -45.5 |

Fig. 9. Uncertain data: Improvement percents relative to independent learning (random-prey).



| | SA | Nrm | Abs | P | N | PO | NO | G | AM |
|-----|-------|--------|------|-------|-------|-------|--------|--------|-------|
| 0 | -7.57 | -51.15 | 13.9 | 14.24 | 14.44 | -1.68 | -44.84 | -47.49 | -0.65 |
| 0.2 | -12 | -58.9 | 7.9 | 8.4 | 8.7 | -3.9 | -45.8 | -44.7 | 6.5 |
| 0.5 | -20.5 | -82.1 | -12 | -9.7 | -13.9 | -11.1 | -67.6 | -76.5 | -21.2 |

Fig. 10. Uncertain data: Improvement percents relative to independent learning (intelligent-prey).

hunting a random-prey is simpler, the learner gets less punishment, compared to the case where the agent hunts an intelligent-prey.

In [22], we showed that, when the received punishments are greater than the rewards, N is the best criteria and P is the worst one among Abs, P, N, and Nrm, but when the rewards are greater

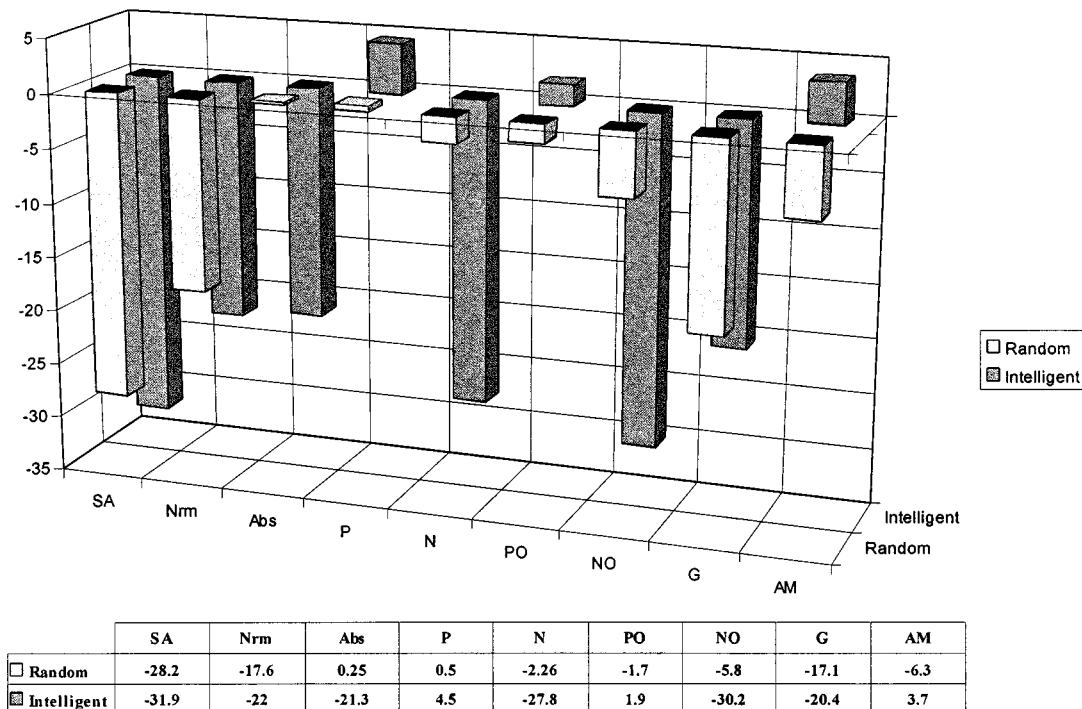


Fig. 11. Incorrect knowledge experiments: Improvement percents relative to independent learning.

than the punishments, P is the best measure and N is the worst one. In addition, when the received punishments and rewards are approximately equal, Abs gains the best results. Nevertheless, since any little difference in the rewards and punishments significantly affects the weights, Nrm criterion has the worst results in this case (Fig. 7).

1) *Divergence of N Criterion:* The N criterion helps the agents with different experiences to learn faster. However, it is important to note that, the learning process may start to diverge after a temporary convergence for N or NO criteria. The divergence time is closer when the absolute value of the punishments is increased. Fig. 8 shows the group learning curves for N where the punishments and rewards are -1 and 5 , respectively. Similar results are obtained for NO. This outcome can be explained as follows.

Before the learning curves converge, the agent having more failures is considered a more expert agent, but after the convergence of the learning process, the number of the expert agent failures decreases and the nonexpert agent receives more punishments. Consequently, it gets a higher expertness value. Therefore, the expert agent is forced to learn from the nonexpert agent and the group learning starts to diverge. As a result, the cooperative learning process must be stopped after convergence if N or NO criterion is used and the punishment value is high.

C. Effects of Uncertainty

In the third type of experiments, it is assumed that there are some uncertainties in the agents' knowledge. To simulate these uncertainties, a normal random number $N(0, v)$ is added to the communicated Q-table cells (this could be assumed as a communication noise). The cooperative learning is done for the agents which have different experiences. The improvement percent of the methods relative to the independent learning

(without uncertainty) for $v = 0.2$ and $v = 0.5$ are compared to $v = 0$ (without noise) and the results are shown in Figs. 9 and 10.

It could be observed that, although PO and NO methods are mainly inefficient, they are two of the least sensitive methods due to less communication. SA is the second method (after PO) in sensitivity because positive and negative random values added to the Q-table cells may, on average, cancel out each other. Also, the Abs method is less sensitive compared to P, N, and Nrm methods.

It is observed that the sensitivity is lower in the intelligent-prey case because the Q-table cells have higher values and the effect of noise is less.

D. Incorrect Knowledge

In this set of experiments, the second agent Q-table is filled with some random numbers between 0 and 1. The improvement percents relative to independent learning are depicted in Fig. 11.

It can be seen that the SA, Nrm, Abs, and N methods are the most sensitive methods. P, PO, and AM methods have the least sensitivities.

V. CONCLUSION AND FUTURE WORKS

In this paper, three weight-assigning procedures for the weighted strategy sharing (WSS) method were introduced. Also, some criteria to measure the expertness of the agents were presented. The introduced methods were tested on the Hunter-Prey problem.

Results showed that the WSS method had no effect (or little effect) on the learning process when the agents had equal experiences. When the experiences of agents were different, the WSS algorithm improved the learning speed.

All of the introduced expertness measures were sensitive to the reinforcement signal value, but as the Abs consider both rewards and punishments as a sign of being experienced, it had the minimum sensitivity.

Obtained results indicated that the best expertness measure is different for different received reinforcement signals. Positive criterion was the best measure when the sum of the received rewards was greater than the punishments in the beginning of learning. On the other hand, when the sum of the received punishments was greater, Negative (N) criteria was the best. When the difference between the rewards and the punishments were little, the N method had the worst results and the Positive (P) and N method had approximately the same effects. Absolute (Abs) was the best measure in such situations.

Usage of incorrect knowledge and uncertain data decreases the cooperative learning quality. When facing uncertain data, PO, NO, and SA were the least sensitive methods and Abs had less sensitivity compared to P, N, and Nrm. In the incorrect knowledge case, SA, Nrm, Abs, and N were the most sensitive methods and P, PO, and AM were the least sensitive ones.

Results in Section IV showed that WSS is sensitive to the value of the reinforcement signal. One of the next steps is to implement a suitable mechanism to dynamically switch between the expertness criteria based on the value of the received reinforcement signals.

The proposed weight assigning method was created for reinforcement learning algorithm, however, we believe other learning methods may reveal more problems and give a better insight into the expertness-measuring subject. Also, a substantial mathematical examination of the proposed approach is one of the subjects of our future research.

In the experiments, the impressibility factors of the agents were equal and fixed over the learning period. This parameter could be dynamically changed according to the effectiveness of the others' knowledge.

Detection of the agents with incorrect knowledge and minimizing their effects on the cooperative group learning is another direction for future research.

REFERENCES

- [1] K. Ali, "A comparison of methods for learning and combining evidence from multiple model," Univ. California, Irvine, Dept. Inform. Comput. Sci., Tech. Rep. 95-47, Nov. 1995.
- [2] E. Alpaydin, "Techniques for combining multiple learners," in *Proc. Eng. Intell. Syst. Conf.*, 1998, vol. 2, pp. 6–12.
- [3] P. Bakker and Y. Kuniyoshi, "Robot see, robot do: An overview of robot imitation," in *Proc. AISB Workshop Learning Robots and Animals*, 1996, pp. 3–11.
- [4] H. R. Berenji and D. A. Vengerov. (2000) Learning, cooperation, and coordination in multi-agent systems. Tech. Rep. IIS-00-10, Intelligent Inference Systems Corp. [Online]. Available: <http://www.iis-corp.com/projects/multi-agent/>.
- [5] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *Proc. AAAI Workshop Multiagent Learning*, 1997.
- [6] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 53–66, Apr. 1997.
- [7] H. Friedrich, M. Kaiser, O. Ragalla, and R. Dillmann, "Learning and communication in multi-agent systems," in *Distributed Artificial Intelligence Meets Machine Learning*, G. Weiss, Ed. New York: Springer-Verlag, 1996, vol. 1221, pp. 259–275.
- [8] A. Garland and R. Alterman, "Multi-agent learning through collective memory," in *Proc. AIII Spring Symp. Adaptation, Co-evolution, Learning in Multi-agent Syst.*, 1996.
- [9] D. Gordon and D. Subramanian, "A multi-strategy learning scheme for agent knowledge acquisition," *Informatica*, vol. 17, pp. 331–346, 1994.
- [10] G. Hayes and J. Demiris, "A robot controller using learning by imitation," in *Proc. Second Int. Symp. Intell. Robot. Syst.*, Renoble, France, 1994, pp. 198–204.
- [11] D. Heath, S. Kasif, and S. Salzberg, "Committees of decision trees," in *Cognitive Technology*, B. Gorayska and J. Mey, Eds. Amsterdam, The Netherlands: Elsevier, 1996, pp. 305–317.
- [12] J. Hu and M. P. Wellman, "Multi-agent reinforcement learning: Theoretical framework and an algorithm," in *Proc. 15th Int. Conf. Machine Learning (ICML)*, Madison, WI, July 1998, pp. 242–250.
- [13] S. Huffman and J. Laird, "Learning procedures from interactive natural language instructions," in *Proc. 10th Int. Conf. Machine Learning*, Amherst, MA, 1993, pp. 143–150.
- [14] I. Kawaishi, S. Yamada, and J. Toyoda, "Experimental comparison of a heterogeneous learning multi-agent system with a homogenous one," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 1996.
- [15] I. D. Kelly, "The development of shared experience learning in a group of mobile robots," Ph.D. dissertation, Univ. Reading, Dept. Cybern., Reading City, U.K., 1997.
- [16] Y. Kuniyoshi *et al.*, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Trans. Robot. Automat.*, vol. 10, pp. 799–822, Dec. 1994.
- [17] Y. Liu and X. Yao, "A cooperative ensemble learning system," in *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN)*, Anchorage, AK, May 1998, pp. 2202–2207.
- [18] R. Maclin and J. W. Shavlik, "Creating advice-taking reinforcement learners," *Mach. Learn.*, vol. 22, pp. 251–282, 1996.
- [19] —, "Incorporating advice into agents that learn from reinforcements," in *Proc. 12th Nat. Conf. Artif. Intell.*, Seattle, WA, 1994, pp. 694–699.
- [20] N. Meuleau, "Simulating co-evolution with mimetism," in *Proc. First Euro. Conf. Artif. Life (ECAL)*, 1991, pp. 179–184.
- [21] J. D. Mostow, "Transforming declarative advice into effective procedures: A heuristic search example," in *Machine Learning: An Artificial Intelligence Approach*, R. Michalski, J. Carbonell, and T. Mitchell, Eds. Palo Alto, CA: Tioga, 1982, vol. 1.
- [22] M. N. Ahmadabadi *et al.*, "Expertness measuring in cooperative learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2000.
- [23] D. Noelle and G. Cottrell, "Toward instructable connectionist systems," in *Computational Architectures Integrating Neural and Symbolic Processes*, R. Sun and L. Bookman, Eds. Boston, MA: Kluwer, 1994.
- [24] F. J. Provost and D. N. Hennessy, "Scaling up: Distributed machine learning with cooperation," in *Proc. 13th Nat. Conf. Artif. Intell. (AAAI)*, 1996.
- [25] R. S. Renner, "Improving generalization of constructive neural networks using ensembles," Ph.D. dissertation, Florida State Univ., Coll. Arts Sci., Tallahassee, 1999.
- [26] A. Samuel, "Some studies in machine learning using the game of checkers," in *Computer and Thought*, E. A. Feigenbaum and J. Feldman, Eds. New York: McGraw-Hill, 1963.
- [27] A. J. C. Sharkey, "On combining artificial neural nets," *Connect. Sci.*, vol. 8, pp. 229–313, 1996.
- [28] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, no. 3, pp. 9–44, 1988.
- [29] R. S. Sutton, Ed., *Machine Learning: Special Issue on Reinforcement Learning*. Cambridge, MA: MIT Press, 1998, vol. 8, pp. 3–4.
- [30] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. Tenth Int. Conf. Machine Learning*, Amherst, MA, June 1993.
- [31] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, U.K., May 1989.
- [32] C. J. C. H. Watkins and P. Dayan, "Q-learning (technical note)," in *Machine Learning: Special Issue on Reinforcement Learning*. Cambridge, MA: MIT Press, 1998, pp. 55–68.
- [33] T. Yamaguchi, Y. Tanaka, and M. Yachida, "Speed up reinforcement learning between two agents with adaptive mimetism," in *Proc. IEEE Conf. Intell. Robot. Syst. (IROS)*, 1997, pp. 594–600.
- [34] T. Yamaguchi, M. Miura, and M. Yachida, "Multi-agent reinforcement learning adaptive mimetism," in *Proc. Fifth IEEE Int. Conf. Emerging Technol. Factory Automat. (ETFA)*, vol. 1, 1996, pp. 288–294.
- [35] —, "Learning cooperative behaviors with spontaneous mimetism," in *Proc. Sixth Int. Fuzzy Syst. Assoc. World Congr.*, vol. 1, 1995, pp. 101–104.



Majid Nili Ahmadabadi was born in Isfahan, Iran, in 1967. He received the B.S. degree in mechanical engineering from Sharif University of Technology, Tehran, Iran, in 1990 and the M.Sc. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 1994 and 1997, respectively.

In 1997, he joined the Advanced Robotics Laboratory at Tohoku University. Later, he moved to the Department of Electrical and Computer Engineering, University of Tehran, where he is currently the Head of the Robotics and AI Laboratory. He is also a Senior

Researcher at the Intelligent Systems Research Center, Institute for Studies on Theoretical Physics and Mathematics (IPM), Tehran. His main research interests are distributed robotics and artificial intelligence (AI), mobile robots, and cooperative learning in multi-agent systems.

Dr. Ahmadabadi initialized the Iranian National Robot Contests in 1999 and is the President of the Executive Committee of these games.



Masoud Asadpour was born in Lar, Iran, in 1975. He received the B.Sc. degree in computer software engineering from Sharif University of Technology, Tehran, Iran, in 1997. He received the M.Sc. degree in AI and robotics from the University of Tehran, in 1999.

Currently, he is a Researcher at the Intelligent Systems Research Center, Institute for Studies on Theoretical Physics and Mathematics (IPM), Tehran. His research interests are cooperative learning, cooperative robotics, and multi-agent systems.