Strategy-based Learning Through Communication With Humans

Nguyen-Thinh Le and Niels Pinkwart

Clausthal University of Technology
Germany
{nguyen-thinh.le,niels.pinkwart}@tu-clausthal.de

Abstract. In complex application systems, there are typically not only autonomous components which can be represented by agents, but humans may also play a role. The interaction between agents and humans can be learned to enhance the stability of a system. How can agents adopt strategies of humans to solve conflict situations? In this paper, we present a learning algorithm for agents based on interactions with humans in conflict situations. The learning algorithm consists of four phases: 1) agents detect a conflict situation, 2) a conversation takes place between a human and agents, 3) agents involved in a conflict situation evaluate the strategy applied by the human, and 4) agents which have interacted with humans apply the best rated strategy in a similar conflict situation. We have evaluated this learning algorithm using a Jade/Repast simulation framework. An evaluation study shows two benefits of the learning algorithm. First, through interaction with humans, agents can handle conflict situations, and thus, the system becomes more stable. Second, agents adopt the problem solving strategy which has been applied most frequently by humans.

Keywords: Agent-Human learning, multi-agent systems, machine learning, evaluation

1 Introduction

In complex application systems, there exist not only autonomous components which can be represented by agents, but humans may also play an important role. Usually, in a multi-agent system, agents have specific pre-defined abilities to perform a certain task. One of the challenges of a multi-agent system is to develop agents with the ability to learn from human behavior. Current research on multi-agent learning exploits machine learning techniques to adapt to preferences or behaviors of human users. In this paper, we present an algorithm that allows autonomous agents to learn problem solving strategies in conflict situations through communication with humans.

Our learning algorithm assumes that humans may have several strategies when encountering a conflict situation. Researchers suggested that experts have some kind of knowledge about problem categories and associated solution strategies which lead to correct solutions [7]. When solving a problem, an expert solver will identify the problem characteristics by associating them with previously solved problems. The problem will be assigned to a solution strategy which can then be applied to solve similar problems. In this paper, the term *strategy* is noted as a particular way of solving a problem as done by human experts. Under this assumption, we propose a strategy-based learning algorithm for autonomous agents. The algorithm consists of four phases. First, the human meets agents in a conflict situation. Then, in the second phase, the human initiates a conversation with involved agents and chooses a strategy to solve the conflict. Third, the agents which are involved in the conflict situation rate the effectiveness of the proposed strategy. In the last phase, based on an aggregated rating score, agents apply the most effective strategy they have learned in similar future conflict situations. To show the benefits of this learning algorithm, we will test the following two hypotheses:

1. Agents applying the strategy-based learning algorithm will adopt the strategy applied most frequently by humans in a similar conflict situation.
2. The more agents applying this algorithm interact with humans in conflict situations, the more stable the system is.

2 State of The Art

There have been numerous attempts on developing algorithms for multi-agent learning. They can be classified into three main approaches which differ from each other in terms of the type of feedback provided to the learner [11]: supervised, unsupervised, and reward-based learning. In supervised learning, the feedback provides the correct output and the objective of learning is to match this desired action as closely as possible. In unsupervised learning, no feedback is provided and the objective is to seek useful activities on a trial-and-error basis. In reward-based learning, the feedback provides a quality assessment (the *reward*) to the learner's action and the objective is to maximize the reward. The class of reward-based learning approach is divided into reinforcement learning methods (which estimate value functions) and stochastic search methods (which directly learn behaviors without explicit value functions). Among the three learning approaches, the reward-based one is most frequently used to enable agents to learn from experience. The supervised learning and unsupervised learning approaches may be difficult to develop. Reinforcement learning techniques [14], an instance of reward-based learning, have been successfully applied in several applications. For instance, Saggar et al. [12] developed a learning algorithm for agents to optimize walks for both speed and stability in order to improve a robot's visual object recognition. The reinforcement learning approach has also been applied for designing a controller for autonomous helicopters, i.e., a helicopter learns how to hover to a place and how to fly a number of maneuvers while considering a learned policy [10]. Schneider et al. [13] introduced a reinforcement learning approach for managing a power grid.

While current work on multi-agent learning often makes use of machine learning techniques, research on agent learning from humans mostly adopts learning approaches for humans or animals. Approaches for providing agents with the ability to learn from humans can be divided into three classes: learning from advice, learning from demonstration, and learning from reinforcement (also referred to as shaping) [4]. The scenario of learning from advice is similar to learning activities of humans, where a learner gets hints from a tutor to perform an action which leads to a correct solution for a given problem. In order to be able to learn from humans, an advice can be expressed using either natural language or a scripting language. When using natural language to give advice, it is challenging to transform an advice into an understandable form for agents. Kuhlman et al. [6] created a domain-specific natural language interface for giving advice to a learner. Using a formal scripting language, coding an advice can be difficult for human trainers [9]. Both approaches, using natural language or a scripting language, are thus challenging.

Learning from demonstration, also referred to as imitation learning or apprenticeship learning, is a technique which aims at extending the capabilities of an agent without explicitly programming the new tasks or behaviors for the agent to perform. Instead, an agent learns a policy from observing demonstrations [1]. Learning from demonstration is comparable to the approach of learning from examples in an educational setting for humans. This approach can be infeasible for some tasks which require special expertise of humans to control the agent (e.g., controlling an helicopter). Taylor et al. [15] proposed a human-agent transfer (HAT) approach which combines transfer learning, learning from demonstration and reinforcement to achieve fast learning. Reinforcement learning techniques require a large amount of training data and high exploration time. Applying learning from demonstration techniques, agents learn directly from humans without explorations, and thus less time would be required compared to the reinforcement approach. However, the quality of demonstrations heavily depends on the ability of the human teacher. The work reported that combining learning from demonstration, reinforcement learning, and transfer learning to transfer knowledge from a human to an agent results in better learning performance than applying each single one.

Learning from reinforcement (or shaping) adopts the approach called clicker training for animals. In a clicker training scenario, an audible clicking device is used as a reinforcement signal when animals perform a correct action (positive assessment). The shaping approach allows a human trainer to shape an agent by reinforcing successively through both positive or negative signals. The TAMER framework is an example of the shaping approach which makes use of positive or negative assessment from humans to teach a good behavior [4]. There exist attempts for mixing the shaping approach with reinforcement learning (which provides an environmental reward within an Markov Decision Process (MDP) [14]). Isbell et al. [3] developed a social software agent which uses reinforcement learning to pro-actively take action and adapt its behavior based on multiple sources of human reinforcement to model human preferences. Knox and Stone [5] attempted to combine TAMER (which uses only human reinforcement) with autonomous learning based on a coded reward. A study showed that this combination (human reinforcement and autonomous reinforcement) leads to better performance than a TAMER agent or an reinforcement learning agent alone.

An approach based on decision tree learning which cannot be classified into the three approaches for agent-human learning above, was introduced by Thawonmas et al. [16]. The authors used a RoboCup simulation system to enable a human player to play soccer against two agents. Based on log data provided by the system, condition-action rules are derived and then applied to the agent. This way, the agent adapts decision-making behaviors of the human player. The evaluation of the system showed that the agent can show almost human decision-making behaviors in a small scenario of playing soccer after five games between a human and agents.

The learning algorithm for agents to be presented in this paper is distinct from previous work on agent-human learning in that it will deploy communication to transfer knowledge from humans to agents.

3 Case study: Smart Airport

In order to illustrate the learning approach pursued in this paper, we use an airport departure scenario as a representative for a complex application system. The airport consists of static objects (single lanes, two-way roads, entrances, check-in counters, gates, plane parking positions, and charging stations) and moving objects (autonomous transportation vehicles (ATVs), and human-controlled vehicles (HCVs)). When there is a request of a passenger to be transported, an agency will provide vehicles (ATVs or HCVs) and manage the passenger's order. A transportation order consists of a start and an end position, pickup time, and a latest time for drop-off. The start and end positions form a route, e.g., from an entrance to a check-in counter. Since both ATVs and HCVs need energy to move, they are equipped with batteries which need to recharge regularly at charging stations. In this airport scenario, conflicts of different types can occur. We will focus on resource conflicts, i.e., two or more participants compete for one resource. Typical resource conflict situations are:

1. At least two (max. four) vehicles are approaching a crossing. One of the vehicles needs the priority to pass through the crossing first. In this situation, the resource required by the vehicles is the crossing.
2. Several vehicles are running out of energy and need to be recharged, while the charging station might be occupied. The resource required by the vehicles is the charging station.
3. Vehicles have to take passengers to unoccupied check-in counters. The resource is a check-in counter. In reality, a check-in counter usually is occupied by one or two staff members, and thus has a maximal capacity of two units.

This application scenario can be described by a multi-agent system in which autonomous vehicles are implemented by autonomous agents and human-controlled vehicles by (non-autonomous) agents. The latter can be controlled by humans, they represent (inter-)actions of humans in the system.

4 Strategy-based Learning

Under the assumption that humans have a set of strategies for a certain conflict situation, this paper proposes a strategy-based learning approach which consists of four phases:

Phase 1: Recognizing a conflict situation According to [17], a conflict is an opportunity for learning, because there occurs a social pressure to solve a conflict when two individuals disagree in a situation. Through resolving a conflict, individuals may change the viewpoint and their behaviors. To detect resource conflicts, a special conflict type, a conflict model and a conflict detection algorithm are required. This conflict model assumes that an agent is able to see its peers within its limited view scope. That is, each agent has information about the last, current, and next possible position of other participants existing in its scope. Thus, the conflict detection mechanism exploits the agent's *belief* about the world state within its scope. Based on this belief, an agent is able to identify other agents that will release/require a resource (an environment element) which is also required by itself. The definition for a *potential conflict* for an agent is taken from [8] as follows.

Definition 1 *Let A be an agent, its current position is $\langle X, Y\rangle$ and its next action is to require an environment element E at position $\langle X_{next}, Y_{next}\rangle$. Let α_E be the set of agents that are occupying E, $\alpha_{release,E}$ and $\alpha_{require,E}$ be the sets of agents (excluding A) that will release/require E, respectively. A has a* ***potential conflict****, denoted as $conflict(E, scope(A), \alpha_E, \alpha_{release,E}, \alpha_{require,E})$, iff $|\alpha_E| - |\alpha_{release,E}| + |\alpha_{require,E}| + 1 > C$, where C is the capacity of E and $scope(A)$ is the scope of A.*

Using Definition 1, an agent which intends to consume an environment element in the smart airport scenario, e.g., a crossing, a charging station, or a check-in counter, is able to detect potential conflicts.

Phase 2: Learning through communication Given a conflict situation C, there exists a set of possible strategies $\{S_1, .., S_n\}$ possibly applied by a human. A strategy is defined formally as follows:

Definition 2 *A strategy is a sequence of requests and replies $\{Q_1A_1, ..., Q_nA_n\}$, where requests Q_i are carried out by a* teacher *and replies A_i are given by a* learner. *A request has one of the types: performing an action, querying data, checking a predicate, or confirming an information.*

We assume that a human has more experience than an autonomous agent, and thus a human plays the role of a *teacher* and an autonomous agent is a *learner*. The human sends requests to an agent and the agent replies to these requests. This way, the agent learns the sequence of requests which have been performed by the human in conflict situations. To establish a conversation between a human and an agent, a communication ontology is required.

In the airport departure case study, for instance, when a HCV meets an ATV at a crossing, a potential conflict occurs as described in Section 3. In this conflict situation, the human (represented by an HCV) may apply one of the following strategies: 1) calculating the priority based on the urgency of transport tasks, 2) calculating priority based on energy states of the HCV and the ATV, or 3) the strategy of politeness, i.e., give way to the participant without requesting to calculate the priority. Table 1 illustrates how the HCV initiates a conversation with the ATV when applying the task-based strategy (left column) and the politeness-based strategy (right column). For the energy-based strategy, the conversation will be similar to the task-based one, the only difference is the first request to calculate the priority based on energy states of HCV and ATV, instead. The agent distinguishes strategies applied by the human through the action requests: *calculate task priority*, *calculate energy priority*, or *give way.* As a result, the agent can "memorize" and adopt these sequences of requests in later similar conflict situations. In general, humans may use multi-modal interactions to indicate their strategy: e.g., using a common language, or through non-verbal acts (gestures or movements). Inferring a humans' strategy from non-verbal acts is beyond the scope of this paper. In the approach pursued in this paper, depending on the sequence of requests initiated by the human to perform actions, the agent (which has no knowledge about strategies on beforehand) can learn strategies used by humans. For instance, if the HCV sends requests to the ATV to compare the priority of two tasks, then the ATV knows that the HCV is pursuing the strategy to compare the urgency of two tasks.

Table 1. Conversation between human-controlled and autonomous vehicles

Task-based	**Politeness-based**
HCV → ATV: *calculate task priority* ATV → HCV: My *priority is higher* HCV → ATV: I *give way* ATV → HCV: confirm *OK*	HCV → ATV: I *give way* ATV → HCV: confirm *OK*

Phase 3: Evaluating human's strategies After the human has communicated with the agent, the conflict should have been solved, i.e., the resource can be allocated according to the conversation between the participants. However, participants might not be delighted with the strategy proposed by the human. For instance, an ATV might have to give way to other participants, because it has lower priority in the context of comparing transport tasks, but this ATV needs to be recharged as soon as possible because its energy state is low. Thus, this raises the need to evaluate the effectiveness of the strategy proposed by the human in each conflict situation. For this purpose, each agent involved in a conflict situation has the opportunity to rate a proposed strategy. Let the rating scalar be from the interval [0;N] where N is the best rating. The total rating for the strategy X which has been initiated by a human is $rating(X) = \sum R_A$, where $R_A \in$ [0;N] is the rating

by an agent A involved in a conflict situation. The total rating for each strategy applied by the human should be maintained and be available for all existing agents. The agents involved in the same conflict situation need to share their ratings. Here, we have to make a trade-off between a centralized coordination and intensive peer communication. Using peer communication, each agent has to send its rating to its peers. However, this solution is very communication intensive. An alternative is using a database to maintain the strategies for different conflict situations and each agent updates the total rating for each strategy. In the approach followed in this paper, we choose the second option.

Phase 4: Applying the best strategy When an autonomous agent detects a conflict with other agents in a situation in which it had a conflict with humans before, the agent takes the set of strategies which have been collected by learning from humans to compare. The best strategy of this set is determined by querying the rating in the strategy database. The strategy which has the highest rating is taken as the best one which can be used. Once again, after the agent has applied that strategy, its peers have the possibility to update the total rating in the strategy database. A question is which agent should initiate a conversation in case of a conflict situation where no human is involved. For this purpose, either an agent who has acquired knowledge should initiate the conversation or one of the involving agents is selected randomly.

5 Implementation

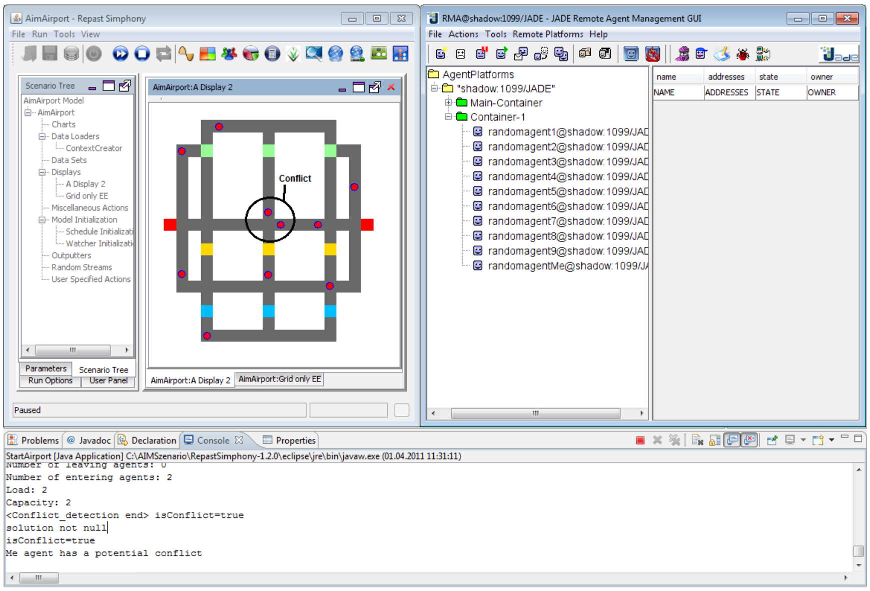
We implemented the airport scenario using the JRep simulation platform [2]. JRep is an integration of Repast Symphony and the JADE Framework. Repast provides a toolkit for visual simulations of multi-agent systems. JADE supports developing intelligent behaviors for agents and provides communication protocols according to FIPA-ACL[1]. In order to enable a conversation between agents and a human, we need to define a communication ontology. We extended JRep with the possibility to add human-controlled vehicles (HCVs) which can be controlled via the four arrow keys of the keyboard. Note, if there exist several HCVs in the airport setting, only one HCV can be moved at a time (while other HCVs cannot perform any action). This is a limitation of our current simulation framework.

The left hand side of Figure 1 illustrates the simulation of a conflict situation at a crossing in a smart airport. The right hand side shows the existence of current agents in the system and the agent management GUI (e.g., to control the communication between agents) provided by JADE.

6 Evaluation

The goal of the evaluation study is to prove that the strategy-based learning algorithm is beneficial for multi-agent systems. For this purpose, we test the two hypotheses mentioned in Section 1.

[1] `http://www.fipa.org/specs/fipa00061`



name	addresses	state	owner
NAME	ADDRESSES	STATE	OWNER

StartAirport [Java Application] C:\AIMSzenario\RepastSimphony-1.2.0\eclipse\jre\bin\javaw.exe (01.04.2011 11:31:11)

```
Number of leaving agents: 0
Number of entering agents: 2
Load: 2
Capacity: 2
<Conflict_detection end> isConflict=true
solution not null
isConflict=true
Me agent has a potential conflict
```

Fig. 1. Enhanced Simulation Platform

6.1 Design

To carry out the evaluation study, we used the simulation framework described in the previous section. In the first round, we carried out six simulations, each with 30 ATVs. The number of HCVs was varied between zero and five. For each simulation, the system was run with 100 ticks. Each tick represents a movement step of an agent or a conversation (consisting of several message exchanges) between an HCV and an ATV (or between ATVs). Then, in the second round, similarly, six other simulations were executed, each with 50 ATVs (and with zero to five HCVs). Results of the second round were used to verify the outcomes of the first round. Each ATV can move randomly around the airport map. If an ATV detects a resource conflict on the road, i.e., the road cell capacity is reached, then the ATV chooses another movement direction. In our study, autonomous agents (ATVs) that have interacted with humans (HCVs) in conflict situations and have thus learned problem solving strategies were able to apply those strategies in the similar conflict situation. Agents without any strategy for solving a conflict situation did not survive conflicts.

6.2 Results

Table 2 shows statistical results of six simulations with 50 ATVs and six simulations with 30 ATVs. The third and the fifth columns of the table show the number of interactions between ATVs and HCVs during 100 simulation ticks. As mentioned in Section 5, during one tick, only one human could interact with agents. In average, we simulated between 44 and 46 ATV-HCV interactions. In order to test whether a higher amount of ATV-HCV interactions resulted in a higher stability of the system, we computed the correlation between the number of ATV-HCV interactions and the number of survived ATVs using the statistics software provided by the Office for Research Development and Education[2]. From the table, we can notice that, in general, through interactions with humans, the number of survived ATVs is higher than in simulations without the existence of humans (indicated by the first row). The Spearman's coefficient shows that for simulations with 30 ATVs, a strong correlation between the number of ATV-HCV interactions and the number of survived ATVS (ρ=0.9) can be identified. For simulations with 50 ATVs, the correlation coefficient (ρ=0.66) is lower, but still relatively high, and shows a moderate positive correlation between the number of ATV-HCV interactions and the number of survived ATVs. As a conclusion, the hypothesis that the more opportunities agents learn problem solving strategies from humans, the less agents will die, can be confirmed.

Table 2. Results of interactions between agents and humans

HCVs	With 50 ATVs		With 30 ATVs	
	Survived ATVs	**Interactions**	**Survived ATVs**	**Interactions**
0	16	0	7	0
1	17	9	17	21
2	30	56	19	42
3	25	47	20	52
4	24	57	20	60
5	29	54	24	57
		m=44.6 (s.d.=20.3)		m=46.4 (s.d.=15.8)
	Spearman's ρ=0.66		Spearman's ρ=0.9	

Table 3 shows how agents adapted their bahavior through interactions with humans. The left part and the right part of the table present the relative frequencies of the strategies applied by HCVs and ATVs, respectively. In boldface, the most frequently used strategy is highlighted. During the simulations with 30 ATVs, we can recognize that the strategy adopted by most agents is always consistent with the strategy chosen most frequently by HCVs, although the relative frequencies differ (ATVs tend to focus on one primary strategy, while HCVs exhibited a more varied behavior). For the simulation with 50 ATVs, the table shows the same tendency. One exception: in the case of the simulation with the existence of 5 HCVs, it is not clear which strategy (politeness or task-based) was favored by HCVs, while the ATVs have decided for the task-based strategy. Hence, the hypothesis that as a result of applying the strategy-based learning algorithm, agents will adopt the strategy most humans applied, can be confirmed.

[2] Wessa, P. (2011), Free Statistics Software, version 1.1.23-r7, URL http://www.wessa.net/

Table 3. Adaptation of agents

HCVs	Strategy applied by HCV (%)			Strategy applied by ATV (%)		
	Politeness	Task-based	Energy-based	Politeness	Task-based	Energy-based
Simulations with 30 ATVs						
1	**100**	0	0	**100**	0	0
2	42.9	**57.1**	0	2.4	**97.6**	0
3	**40.4**	30.8	28.8	**97.9**	0	2.1
4	**46.7**	33.3	20	**77.8**	22.2	0
5	**57.9**	29.8	12.3	**96.5**	3.5	0
Simulations with 50 ATVs						
1	**100**	0	0	**100**	0	0
2	33.9	**66.1**	0	6.5	**93.5**	0
3	27.7	**38.3**	34	21.7	**72.2**	6.1
4	**43.9**	14	42.1	**73.9**	0	26.1
5	**40.7**	**40.7**	18.5	0	**100**	0

7 Conclusion

In this paper, we have presented a strategy-based learning algorithm for agents through communication with humans. The learning process consists of four phases: 1) detecting conflict situations, 2) humans initiating a conversation with agents and deciding on a conflict solving strategy, 3) agents involved in the conflict situation rate the effectiveness of the proposed strategy, and 4) the agent applies the most effectively rated strategy in a similar situation. We have conducted a pilot study to evaluate the benefits of this learning algorithm. The evaluation shows that the multi-agent system with interactions between humans and agents becomes more stable than a system without interactions with humans, and that agents adopt the problem solving strategy applied most frequently by humans. Note, the evaluation assumed that the agents have no knowledge how to solve conflicts as long as they have not interacted with humans. This learning approach for agents is novel in that it exploits the communication ability of agents (using the FIPA-ACL protocols) to be instructed by humans, whereas most existing work is based on machine learning techniques. In the future, we will try to shorten the conversation steps between humans and agents, because a long conversation is not appropriate for traffic situations and is resource expensive.

References

1. Argall, B. D., Chernova, S., Veloso, M., Browning, B.: A Survey of Robot Learning From Demonstration. J. Robotics and Autonomous Systems, 57(5), pp. 469–483, Elsevier (2009)
2. Görmer, J., Homoceanu, G., Mumme, C., Huhn, M., Müller, J. P.: JRep: Extending Repast Simphony for Jade Agent Behavior Components. In Proceedings of the IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology, pp. 149–154 (2011)

3. Isbell, C., Kearns, M., Singh, S., Shelton, C., Stone, P., Kormann, D.: Cobot in LambdaMOO: A Social Statistics Agent. In: J. Autonomous Agents and Multiagent Systems, 13(3), pp. 327–354, Springer, Netherlands (2006)
4. Knox, W. B., Stone, P.: Interactively Shaping Agents via Human Reinforcement - The TAMER Framework. In: Proceedings of the 15th International Conference on Knowledge Capture, pp. 9–16, ACM, New York, USA (2009)
5. Knox, W. B., Stone, P.: Combining manual feedback with subsequent MDP reward signals for reinforcement learning. In: Proceedings of the 9th Int. Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp.5–12, AAMAS (2010)
6. Kuhlmann, G., Stone, P., Mooney, R. J., Shavlik, J. W.: Guiding a Reinforcement Learner With Natural Language Advice: Initial Results in RoboCup Soccer. In: Proceedings of the AAAI Workshop on Supervisory Control of Learning and Adaptive Systems (2004)
7. Le, N. T., Menzel, W., Pinkwart, N.: Considering Ill-definedness of Problems From The Aspect of Solution Space. In: Proceedings of the 23rd International Florida Artificial Intelligence Conference (FLAIRS), pp. 534–535, AAAI Press (2010)
8. Le, N. T., Mrtin, L., Pinkwart, N.: Learning Capabilities of Agents in Social Systems. In: Proceedings of The 1st International Workshop on Issues and Challenges in Social Computing (WICSOC), held at the IEEE International Conference on Information Reuse and Integration (IRI) (pp. 539 –544), NJ, IEEE (2011)
9. Moreno, D. L., Regueiro, C. V., Iglesias, R., Barro, S.: Using Prior Knowledge to Improve Reinforcement Learning in Mobile Robotics. In: Proceedings of Towards Autonomous Robotic Systems (TAROS), Technical Report Series, Report Number CSM-415, Department of Computer Science, University of Essex (2004)
10. Ng, A. Y., Kim, H. J., Jordan, M. I., Sastry, S.: Inverted Autonomous Helicopter Flight Via Reinforcement Learning. In: International Symposium on Experimental Robotics, MIT Press (2004)
11. Panait, L., Luke, S.: Cooperative Multi-agent Learning: The State of The Art. J. Autonomous Agents and Multi-Agent Systems, 11(3), pp. 387–434 (2005)
12. Saggar, M., DSilva, T., Kohl, N., Stone, P.: Autonomous Learning of Stable Quadruped Locomotion. In RoboCup-2006: Robot Soccer World Cup X, LNAI, vol. 4434, pp. 98–109, Springer, Berlin (2007)
13. Schneider, J., Wong, W. K., Moore, A., Riedmiller, M.: Distributed Value Functions. In: Proceedings of the 16th International Conference on Machine Learning, pp. 371–378, Morgan Kaufmann (1999)
14. Sutton, R. S., Barto, A. G.: Reinforcement Learning: An Introduction. MIT Press (1998)
15. Taylor, M. E., Suay, H. B., Chernova, S.: Integrating Reinforcement Learning with Human Demonstrations of Varying Ability. In: Proceedings of the 10th Int. Conference on Autonomous Agents and Multiagent Systems, pp. 617–624, AAMAS (2011)
16. Thawonmas, R., Hirayama, J., Takeda, F.: Learning From Human Decision-making Behaviors - An application to Robocup Software Agents. In: Proceedings of the 15th International Conference on Industrial and Engineering, Applications of Artificial Intelligence and Expert Systems, LNCS, vol. 2358, pp. 136–145, Springer (2002)
17. Weiß, G., Dillenbourg, P.: What is 'multi' in Multi-agent Learning. In: Dillenbourg (ed.) Collaborative-learning: Cognitive, pp. 64–80, Pergamon Press, Oxford (1999)