Machine Learning Algorithms for Multi-Agent Systems

Conference Paper · November 2015

DOI: 10.1145/2816839.2816925

4 authors, including:

Khaled M. Khalil
Ain Shams University

Mohamed Abdelaziz
Ain Shams University

Abdel-Badeeh M.Salem
Ain Shams University

Machine Learning Algorithms for Multi-Agent Systems

Khaled M. Khalil, M. Abdel-Aziz, Taymour T. Nazmy, Abdel-Badeeh M. Salem
Faculty of Computer and Information Science Ain shams University Cairo, Egypt
kmkmohamed@gmail.com, mhaziz67@gmail.com, ntaymoor@yahoo.com, abmsalem@yahoo.com

ABSTRACT

Multi-agent systems are rapidly used in a variety of domains, including robotics, distributed control, telecommunications, collaborative decision support systems, and economics. The complexity of many tasks arising in these domains makes them difficult to solve with preprogrammed agent behaviors. The agents must instead discover a solution on their own, using learning. The heart of the problem is how the agents will learn the environment independently and then how they will cooperate to establish the common task. This paper will attempt to answer these commonly asked questions from the machine learning perspective. In addition, this paper discusses the current gap and challenges in adoption of machine learning algorithms in multi-agent systems.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence – *coherence and coordination, intelligent agents, multiagent systems*.

General Terms

Algorithms, Performance, Design, Theory.

Keywords

Machine Learning; Artificial Intelligence; Multi-Agent Systems.

1. INTRODUCTION

An agent is an entity that perceives its environment through sensors and acting upon that environment through effectors [1]. In multi-agent systems, agents are independent in that they have independent access to the environment. They need to adapt to new circumstances and to detect and extrapolate patterns. Therefore, each agent should incorporate a learning algorithm to learn and explore the environment. In addition and due to interaction among the agents in multi-agent systems, they should have some sort of communication between them to behave as a group. In other words, they must organize themselves to act together [2].

The advantage of multi-agent learning is that the performance of the agent or of the completely multi-agent system gradually improves. Multi-agent learning is not a matter of straight learning, but it is involving complex patterns of social interactions. This leads to complex collective functions [3]. Many of algorithms developed in machine learning can be transferred to settings where there are multiple, interdependent, interacting learning agents. However, they may require modification to account for the other agents in the environment [4, 5]. Furthermore, multi-agent systems present a set of unique learning opportunities over and above single-learner approach.

In this paper, we elaborate different machine learning techniques in multi-agent systems. Then we explore the current gap and challenges faced by researchers when they adopt machine learning algorithms in multi-agent systems. We organized this paper as follows: Section 2 introduces the machine learning in multi-agent systems. While, section 3 discusses machine learning techniques and functional perspectives for multi-agent systems. At the end, we discuss machine learning gap and challenges in multi-agent systems in section 4 and concludes in section 5.

2. LEARNING IN MULTI-AGENT SYSTEMS

Spears [6] raised the following questions needed to be addressed for learning agents: 1) Is learning applied to one or more agents?, 2) If multiple agents, are they competing or cooperating?, 3) What element(s) of the agent(s) are got adapted?, and 4) What algorithms(s) are used to adapt?. Answering these questions influenced on the majority of research directions in the domain during the last decade. Moreover, they yielded with a conceptual view of learning agent elements and different learning aspects to be considered while adopting or building a learning agent.

In this section, we start with the elements of learning agent and then we discuss how they should interact with each other. Then, we discuss different learning aspects. This characterize answers to the first three questions, while we list different learning techniques in section 3 as an answer to question number four.

2.1 Elements of Learning Agents

Russell and Norvig [7] have conceptually structured generic learning agent architecture into four elements: 1) a learning element, 2) a performance element, 3) a critic element, and 4) a problem generator. Figure 1 shows the four elements and their interaction patterns. The performance element takes input that the agent perceives at any given moment and decides on actions. The learning element uses feedback from the critic on how the agent is doing and determines how the performance element should be modified to do better in the future. The critic element tells the learning element how well the agent is doing with respect to the agent performance. The last element is the problem generator and it suggests actions that will lead to new informative experiences. Learning alters agent elements based on the learning goal, for example, it may change the choice of action to take in response to a stimulus (learning and performance elements). In addition, it could be initiated in response to success/failure (critic, learning and performance elements), or it could be prompted for general performance improvement (altered all learning elements).

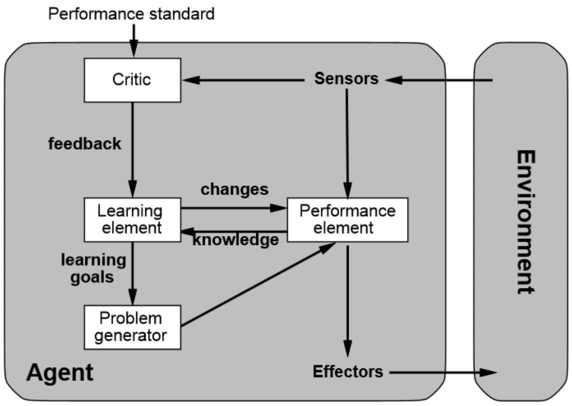

Figure 1. A general model of learning agents [7].

2.2 Multi-Agents Learning Aspects

Weiss [8] described six learning aspects for characterizing learning in multi-agent systems. In this subsection, we present and describe each of these learning aspects and then we will classify studied algorithms based on them. The learning aspects include: 1) The degree of decentralization: Is the learning centralized or distributed? In centralized systems, only one agent in the system learns for all of the agents in the system. In decentralized systems, all of the agents learn and adapt in a distributed and possibly simultaneous fashion. 2) Interaction-specific Aspects: What is the nature of the interactions among the agents in the system? Weiss [8] discussed the "level of interaction," the "pattern of interaction," and the variability of the interactions. Interactions can be based on observations, indirect effects from the environment, or explicit relationships. Additionally, interactions can change over time as agents move through an environment or modify their relationships with other agents. 3) Involvement-specific Aspects: How involved are each of the individual agents in the learning process? The learning of individual agents can be local or global. 4) Goal-specific Aspects: Are the goals of the agents selfish or collective? Multi-agent systems can be competitive, cooperative, and sometimes, both. In competitive systems, such as games, agents are selfish, attempting to maximize their individual reward regardless of the reward received by other agents. In cooperative systems, agents work toward common goals where the reward structure is shared, or common, among all of the agents. 5) The learning algorithm: How does learning take place for agents in the system? 6) The learning feedback: How do agents know if their behaviors are beneficial or detrimental?

Finally, we can add an additional aspect from [9], the On-line (or incremental) learning: Are agents able to accumulate its own knowledge and add new experience as soon as a new training example becomes available? Obviously, on-line algorithms are better suited for multi-agent systems where agents need to update their knowledge constantly.

3. LEANRING TECHNIQUES AND FUNCTIONAL PERSPECTIVES

The manner in which agent learning takes place and the output of the learning process depends on the underlying algorithms. These algorithms typically fall into one of several techniques, depending on the task, knowledge structure, and the output desired as a result matching learning aspects mentioned in section 2. These techniques can be classified according to a variety of perspectives. The most common classifying perspectives are machine learning perspective, and multi-agent functional perspective. In the following sub-sections, we will go through these two perspectives in details and we will list different algorithms underlying them.

3.1 Machine Learning Techniques

Machine learning perspective is distinguished by what kind of feedback the critic provides to the learner. Based on this criterion, the three main techniques to learn are: 1) Supervised Learning, 2) Unsupervised Learning, and 3) Reinforcement Learning. In supervised learning, the critic provides the correct output. In unsupervised learning, no feedback is provided at all. While in reinforcement learning, the critic provides a quality assessment (the "reward") of the learner's output. In all three cases, the learning feedback is assumed to be provided by the system environment or the agents themselves.

3.1.1 Supervised Learning

Because of the inherent complexity in the interactions of multiple agents, various supervised machine learning methods are not easily applied to the problem because they typically assume a critic that can provide the agents with the "correct" behavior for a given situation. However, there are some works considered as a notable exception supervised learners. Sniezynski [10] is using supervised rule learning method for the fish bank game. Garland and Alterman [11] are using learning coordinated procedures to enhance coordination in environment of heterogeneous agents. Williams [12] is using inductive learning methods to learn individual's ontologies in the field of semantic webs. Gehrke and Wojtusiak [13] are using rule induction methods for route planning. Airiau et al. [14] add learning capabilities into BDI model, in which decision tree learning is used to support plan applicability testing. Table 1 shows each one of the above supervised learners compared to the learning aspects mentioned in section 2.

Table 1. Supervised learners vs. learning aspects

	Centralization	Interaction	Learning	Goal
Sniezynski [10]	C	X	2 agents	S
Garland and Alterman [11]	D	√	All agents	Co
Williams [12]	D	√	All agents	Co
Gehrke and Wojtusiak [13]	D	√	All agents	Co
Airiau et al. [14]	D	√	All agents	Co
C: Centralized, D: Distributed, S: Selfish, Co: Cooperative				

3.1.2 Unsupervised Learning

In unsupervised learning no explicit target concepts are given. It is not well suited for the multi-agent learning too. The goal of multi-agent learning is to improve the overall performance of the system, whereas unsupervised learning is a kind of aimless learning. Thus, there is no significant research in this area. However, researchers tried to use unsupervised learning algorithms to solve subsidiary issues that help agent through its learning [15, 16].

3.1.3 Reinforcement Learning

Unlike supervised and unsupervised learning where the learner must be supplied with training data, the premise of reinforcement learning matches the agent paradigm exactly. Reinforcement learning allows an autonomous agent that has no knowledge of a task or an environment to learn its behavior by progressively improving its performance based on given rewards as the learning task is performed [17]. Thus, the very large majority of papers in this field have used reinforcement learning algorithms.

The reinforcement learning algorithms for multi-agent systems literature can be divided into two subsets: 1) Estimating value functions based methods; and 2) Stochastic search based methods in which agents directly learn behaviors without appealing to value functions and it concentrates on evolutionary computation.

For learning methods based on estimate value functions, we have single and multiple agents' algorithms. Many single-agent algorithms exist: 1) Methods based on dynamic programming such as Bertsekas [18] and Puterman [19], 2) Methods based on online estimation of value functions such as Watkins and Dayan [20] and Barto et al. [21], and 3) Methods that learn using world model-based techniques such as Sutton [22] and Moore and Atkeson [23]. While, most of the multi-agent algorithms are derived from Q-Learning algorithm and Temporal-Difference algorithm. Q-Learning based algorithm such as work of Bowling and Veloso [24], and Greenwald and Hall [25]. Temporal-Difference based algorithms such as work of Kononen [26] and Lagoudakis and Parr [27].

For the learning methods based on stochastic search, we have learning using genetic algorithm such as McGlohon and Sen [28], and Qi and Sun [29]. Table 2 shows reinforcement learners compared to the learning aspects mentioned in section 2.

Table 2. Reinforcement learners vs. learning aspects

	Centralization	Interaction	Learning	Goal
Bertsekas [18]	C	X	All agents	S
Puterman [19]	C	X	All agents	S
Watkins and Dayan [20]	C	X	All agents	S
Barto et al. [21]	C	X	All agents	S
Sutton [22]	C	X	All agents	S
Moore and Atkeson [23]	C	X	All agents	S
Bowling and Veloso [24]	D	√	All agents	Co/Cm
Greenwald and Hall [25]	D	√	All agents	Cm
Kononen [26]	D	√	One agent	Co
Lagoudakis and Parr [27]	C	X	One agent	S
McGlohon and Sen [28]	D	√	All agents	Co
Qi and Sun [29]	D	√	All agents	Co
C: Centralized, D: Distributed, S: Selfish, Co: Cooperative, Cm: Competitive				

3.2 Multi-Agent Functional Perspective

The functional perspective shows the semantics associated with the agents functional roles that are motivated by the occurrence of events [30]. According to this, work in multi-agent learning can be divided into five agendas [31]: computational (compute equilibria), descriptive (describe human learners), normative (understand equilibria arising between learners with game theory tools), prescriptive cooperative (learn with communication, distributed problem solving), and prescriptive non cooperative (learn without communication). The first three agendas are related to the convergence of the learning algorithms to the optimal behavior, and modeling the world and different learners' goals respectively. They have a direct influence on the machine learning functions within the multi-agent system but they are indirectly influencing the function of the multi-agent system. Thus, we will elaborate on the last two perspectives: cooperative and non-cooperative multi-agent learning which have direct force on the nature of the multi-agent system function.

3.2.1 Cooperative Learning

This approach is to consider the problem as a whole, with the aim of finding optimal joint actions [32]. There are several advantages of the cooperative learning. It allows a system of individual agents with limited resources to scale up with the use of cooperation [33]. In addition, using multiple learners may also increase speed and efficiency for large and complex tasks. Finally, multiple learners allow the system to encapsulate specialized expertise in particular agents. Under the umbrella of cooperative learning, we can discuss the most significant learning methods: social learning, team learning, and concurrent learning methods. These methods are distinguished based on the way one agent might learn from the behavior of another.

Social learning is inspired by research of animals learning [34]. This involves a new agent that can benefit from the accumulated learning of the population of more experienced agents. Examples of social learning algorithms are Garland and Alterman [11], Williams [12], Bowling and Veloso [24], Greenwald and Hall [25], and McGlohon and Sen [28].

In team learning, a single learning agent is discovering behaviors for other agents. Team learning is an easy approach to multi-agent learning because it can use standard single-agent machine learning techniques. However, it lets the learning agent to process large state space for the learning process. Team learning methods are applied in Gehrke and Wojtusiak [13], and Qi and Sun [29].

Concurrent learning is the most common alternative to team learning in cooperative multi-agent systems, where multiple learning processes attempt to improve parts of the team. Typically, each agent has its own unique learning process to modify its behavior [35]. The central challenge for concurrent learning is that each learner is adapting its behaviors in the context of other co-adapting learners over which it has no control. Concurrent learning methods are applied in Airiau et al. [14].

3.2.2 Non-Cooperative Learning

In non-cooperative learning, the overall behavior emerges from the interaction of the agents' behaviors. Since internal processing is avoided, these techniques allow the agent systems to respond to the changes in their environment in a timely fashion. As a side effect, agents do not have domain knowledge that is essential for making the right decision in complex, dynamic scenarios such as in work of Sniezynski [10], and Moore and Atkeson [23], and Lagoudakis and Parr [27].

4. DISCUSSIONS AND CHALLENGES

Multi-agent learning is a new field and thus it still poses many open research challenges. In this section, we elaborate major challenges and the gap observed recurring while surveying past literature. These challenges may eventually require new learning methods special to multiple agents, as opposed to the more conventional single-agent learning methods now common in the field.

4.1 Goal Setting

The multi-agent system goals typically formulate conditions for static games, in terms of strategies and rewards [35]. As a solution in dynamic games, some of the goals can be formulated in terms of stage strategies and expected return instead of strategies and rewards. The issue of a suitable learning goal requires additional work especially in the general stochastic learning. That is due to the agents' returns are correlated and cannot be maximized independently. This needs to find a solution towards the stability of the learning algorithm and its adaptation to behavior changes of other agents. In other words, converging to group equilibrium is required in such cases. At the end, learning goal should include bounds on the system performance.

4.2 Scalability

Most of the developed multi-agent learning algorithms are applied to small problems only, like static games and small grid-worlds [33]. Consequently, these algorithms are unlikely to scale up to real-life multi-agent problems, where the state and action spaces are large or even continuous. The dimensionality of the search space also grows rapidly with the number and complexity of agent behaviors, the number of agents involved, and the size of the network of interactions between them in the multi-agent systems. Since basic reinforcement learning algorithms estimate values for each possible discrete state or state-action pair, this growth leads directly to an exponential increase of their computational complexity. In addition, extra learning agents add their own variables to the joint state-action space. Possible solution may be by reducing the heterogeneity of the agents, or by reducing the complexity of the agent's behavior. Technique such as hybrid teams and partial restriction of world models provide promising solutions in this direction.

4.3 Communication Bandwidth

While it has been demonstrated that communicating information can be beneficial in multi-agent systems, in many domains, bandwidth considerations do not allow for constant, complete exchange of such information. In addition, if communications delayed, they may become obsolete before arriving at their intended destinations. In such cases, it may be possible for agents to learn what and when to communicate with other agents based on observed affects of invalid performance. There is a clear need to pre-define the communication language and protocol for use by the agents. However, an interesting alternative would be to allow the agents to learn for themselves what to communicate and how to interpret it.

4.4 Dynamic Systems

Multi-agent systems are typically dynamic environments, with multiple learning agents competing for resources and tasks. The question here is "How do agents learn in an environment where the goals are constantly and adaptively being moved?" Agents need to find optimal behaviors but it will be better to find sub-optimal behavior to reach global goal achievement.

Most of the reinforcement learning algorithms adopted in multi-agent systems are very time-consuming for learning a simulated control task and many of them work well, but only with the right parameter setting. Thus, searching for a good parameter setting is another time- consuming learning process.

Exploration-exploitation trade-off in dynamic environments is another face of the dynamic environment challenge. Learning algorithms need to strike a balance between the exploitation of the agent's current knowledge, and exploratory actions taken to improve that knowledge.

4.5 Domain Problem Decomposition

To simplify the dynamics of the domain problem, agents need to use the available domain knowledge. A smaller set of actions for the problem domain will increase the effectiveness of the multi-agent systems reaching their goals. Such decomposition can be implemented at various levels: team actions, each action can be divided into sub-actions. Then, each sub-action can be learned independently and iteratively. Which formal technique is suitable for such decomposition in dynamic environments is an open question. In addition, how and when the sub-actions can be paralleled in execution and how to handle the dependency between sub-actions among themselves and with the team actions are other open area.

5. CONCLUSION

Intelligence implies a certain degree of autonomy, which in turn, requires the ability to make independent decisions. Intelligent agents have to be provided with the appropriate tools to make such decisions. In most dynamic domains, a designer cannot possibly foresee all situations that an agent might encounter, and therefore, the agent needs the ability to learn from and adapt to new environments. This is especially valid for multi-agent systems, where complexity increases with the number of agents acting in the environment. For these reasons, machine learning is an important technology to be considered by designers of intelligent agents and multi-agent systems.

We reviewed different learning agent elements and aspects considered while adopting or building the learning agent. Furthermore, we reviewed several techniques and algorithms in the multi-agent learning literature. We based our review on machine learning and multi-agent functional perspectives. Machine learning perspective includes supervised, unsupervised and reinforcement learning, while multi-agent functional perspective includes cooperative and non-cooperative learning.

Throughout the paper, machine learning algorithms are emphasized. Although each domain requires a different approach, from a research perspective the ideal domain embodies as many techniques as possible. We have found that reinforcement algorithms are the most common algorithms for agents learning because they match the agent paradigm exactly. While, different techniques can be adopted for domains based on each domain requirements. Within our review, we compared different algorithms based on their matching of the learning aspects.

After summarizing a wide range of such existing work, we presented future directions that we believe that they will be useful. We discussed the gap and challenges of multi-agent learning, and some methods to address them. Two particular challenges are the formal statement of the multi-agent learning goal, and applying the learning algorithms to real and large problems. We believe that there is still a lot of work in the multi-agent learning. Moreover, the question of which techniques and algorithms are suitable for specific domain is still a matter of debate.

6. REFERENCES

[1] Shoham, Y., and Leyton-Brown, K.. 2009. *Multiagent Systems Algorithmic, Game-Theoritic, and Logical Foundations*. Cambride University Press.

[2] Martinez-Gil, F., Lozano, M., and Fernández, F. 2014. Emergent Collective Behaviors in a Multi-agent Reinforcement Learning Pedestrian Simulation: A Case Study. In *Proceedings of Workshop on Multi-Agent Systems and Agent-Based Simulation,* 228-238.

[3] Parde, N., and Nielsen, R. D. 2014. Design Challenges and Recommendations for Multi-Agent Learning Systems Featuring Teachable Agents. In *Proceedings of the 2nd Annual GIFT Users Symposium (GIFTSym2),* 12-13.

[4] Yang, Z., and Shi, X. 2014. An agent-based immune evolutionary learning algorithm and its application. In *Proceedings of the Intelligent Control and Automation (WCICA)*. 5008-5013.

[5] Qu, S., Jian, R., Chu, T., Wang, J., and Tan, T. 2014. Comuptional Reasoning and Learning for Smart Manufacturing Under Realistic Conditions. In *Proceedings of the Behavior, Economic and Social Computing (BESC) Conferencece*, 1-8.

[6] Spears, D. F. 2006. Assuring the Behavior of Adaptive Agents. In *Agent Technology from a Formal Perspective*, NASA Monographs in Systems and Software Engineering, 227-257.

[7] Russell S., and Norvig, P. 2003. *Artificial Intelligence: A modern Approach*, 2nd Ed., New Jersey: Prentice Hall.

[8] Weiss, G. 1999. *Multiagent Systems A Modern Approach to Distributed Modern Approach to Artificial Intelligence*, 1st ed., MIT Press, 259-298.

[9] Alonso, E., d'Inverno, M., Kudenko, D., Luck, D., and Noble, J. 2001. Learning in Multi-Agent Systems. *The Knowledge Engineering Review* 16, 277-284.

[10] Sniezynski, B. 2009. Supervised Rule Learning and Reinforcement Learning in A Multi-Agent System for the Fish Banks Game. *Theory and Novel Applications of Machine Learning*.

[11] Garland, A., and Alterman, A. 2004. Autonomous agents that learn to better coordinate. *Autonomous Agents and Multi-Agent Systems* 8, 267-301.

[12] Williams, A. 2004. Learning to share meaning in a multi-agent system. *Autonomous Agents and Multi-Agent Systems* 8, 165-193.

[13] Gehrke, J. D., and Wojtusiak, J. 2008. Traffic Prediction for Agent Route Planning. In *Proceedings of the International Conference on Computational Science*, 692-701.

[14] Airiau, S., Padham, L., Sardina, S., and Sen, S. 2008. Incorporating Learning in BDI Agents. In *Adaptive Learning Agents and Multi-Agent Systems Workshop (ALAMAS+ALAg-08).*

[15] Mirmomeni, M., and Yazdanpanah, M. J. 2008. An unsupervised Learning method for an attacker agent in robot soccer competitions based on the Kohonen Neural Network. *International Journal of Engineering, Transactions A: Basics* 21, 255-268.

[16] Kiselev, A. 2008. A self-organizing multi-agent system for online unsupervised learning in complex dynamic environments. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 1808-1809.

[17] Sadeghlou, M., Akbarzadeh-T, M. R., and Naghibi-S, M. B. 2014 Dynamic agent-based reward shaping for multi-agent systems. In *Proceedings of the Iraniance Conference on Intelligent Systems (ICIS)*, 1-6.

[18] Bertsekas, D. P. 2001. *Dynamic Programming and Optimal Control*, 2nd Ed., Athena Scientific.

[19] Puterman, M. L. 2008. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st Ed., Wiley.

[20] Watkins, C. J. C. H., and Dayan, P. 1992. Q-learning. *Machine Learning* 8, 279-292.

[21] Barto, A. G., Sutton, R. S., and Anderson, C. W. 1983. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics* 5, 843-846.

[22] Sutton, R. S. 1990. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning (ICML-90)*, Austin, US, 216-224.

[23] Moore, A. W., and Atkeson, C. G. 1993. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning* 13, 103-130.

[24] Bowling, M., and Veloso, M. 2002. Multiagent learning using a variable learning rate. *Artificial Intelligence* 136, 215-250.

[25] Greenwald, A., and Hall, K. 2003. Correlated-Q learning. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-03)*, Washington, US, 242-249.

[26] Kononen, V. 2005. Gradient descent for symmetric and asymmetric multiagent reinforcement learning. *Web Intelligence and Agent Systems* 3, 17-30.

[27] Lagoudakis, M. G., and Parr, R. 2003. Least-squares policy iteration. *Machine Learning Research* 4, 1107-1149.

[28] McGlohon, M., and Sen, S. 2004. Learning to Cooperate in Multi-Agent Systems by Combining Q-Learning and Evolutionary Strategy. *In Proceedings of the World Conference on Lateral Computing*.

[29] Qi, D., and Sun, R. 2003. A multi-agent system integrating reinforcement learning, bidding and genetic algorithms. *Web Intelligence and Agent Systems* 1, 187-202.

[30] Rodríguez, L., Insfran, E., and Cernuzzi, L. 2011. Requirements Modeling for Multi-Agent Systems. In *Multi-Agent Systems*, InTech Press, 3-22.

[31] Shoham, Y., Powers, R., and Grenager, T. 2007. If multi-agent learning is the answer, what is the question?. *Artificial Intelligence* 171, 365–377.

[32] Panait, L., and Luke, S. 2005. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems* 11, 387–434.

[33] Weiß, G. 1996. Adaptation and Learning in Multi-Agent Systems: Some Remarks and a Bibliography. In *Proceedings of the International Joint Conference on Artificial Intelligence - Workshop on Adaption and Learning in Multi-Agent Systems*, 1–21.

[34] Noble, J., and Franks, D. W. 2003. Social Learning in a Multi-Agent System. *Computers and Artificial Intelligence* 22, 561-574.

[35] Miconi, T. When evolving populations is better than coevolving individuals: The blind mice problem. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, 647-652.