Multi-agent learning for engineers ☆

Shie Mannor [a], Jeff S. Shamma [b],*

[a] *Department of Electrical & Computer Engineering, McGill University, Canada*
[b] *Department of Mechanical and Electrical Engineering, University of California, Los Angeles, USA*

Abstract

As suggested by the title of Shoham, Powers, and Grenager's position paper [Y. Shoham, R. Powers, T. Grenager, If multi-agent learning is the answer, what is the question? Artificial Intelligence 171 (7) (2007) 365–377, this issue], the ultimate lens through which the multi-agent learning framework should be assessed is "what is the question?". In this paper, we address this question by presenting challenges motivated by engineering applications and discussing the potential appeal of multi-agent learning to meet these challenges. Moreover, we highlight various differences in the underlying assumptions and issues of concern that generally distinguish engineering applications from models that are typically considered in the economic game theory literature.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Multi-agent systems; Cooperative control; Distributed control; Learning in games; Nash equilibrium

1. Introduction

In this paper we address the question "if multi-agent learning is the answer, what is the question?" posed in [34] by looking at the engineering agenda. As opposed to the descriptive agenda that tries to explain micro or macro economic phenomena using simple learning rules, or the predictive agenda that tries to forecast what could happen, the engineering agenda concerns designing systems that would satisfy certain pre-specified performance criteria. The purpose of multi-agent learning from an engineering perspective is therefore to assist in the design of a complex system that includes multiple agents.

From an engineering point of view, as we argue below, one of the main benefits of multi-agent learning is its potential applicability as a design methodology for distributed control, which is a branch of control theory that deals with design and analysis of multiple controllers that operate together to satisfy certain design requirements.

In this paper, we motivate the use of multi-agent learning in these domains in Section 2 as a means to simplify the design process of a distributed control system while reducing the complexity of each controller, taking uncertainty into account, and allowing for distributed interactions between the controllers. Issues that distinguish the engineering view of multi-agent learning from the descriptive and predictive views are discussed in Section 3. We finally provide an outlook to the engineering agenda and the challenges that it poses to multi-agent learning in Section 4.

2. Why multi-agent learning?

The problem of designing optimal (or even just reasonable) distributed control systems is notoriously difficult. Examples of complex engineering problems include scheduling in manufacturing systems [2,16,24,25], routing in data networks [3,4,26,30,31], and command and control of networked forces in adversarial environments [1,9,29]. These applications entail a collection of dispersed interacting components that seek to optimize a global collective objective through local decision making. The task is complicated by limited communication capabilities, local and dynamic information, faulty components, and an uncertain, if not hostile, environment. In general, it is not feasible to pass all information to a command center that could process this information and disseminate instructions. Furthermore, even if this were possible, the complexity of the overall system would make the problem of constructing a centralized optimal policy intractable.

It is both the complexity and distributed nature of these problems that motivate the use of game theoretic methods. A multi-agent viewpoint lets one look at an overall systems as a collection of simpler interacting components. Note that this means *choosing* to impose a multi-agent structure as a *design approach*. The result is that the decision making process for any single component is dictated by an optimization problem that is greatly simplified as compared to the centralized problem, but coupled to the decisions of other interconnected components. Accordingly, a Nash equilibrium reflects an optimality condition from the perspective of each individual component, but need not reflect optimal operation from as a collective. Nonetheless, the possibility of the system to self-organize into a suboptimal Nash equilibrium is less daunting than the prospect of constructing a centralized optimal policy.

3. Descriptive, predictive, or engineered?

It is safe to say that much of the research in multi-agent learning has its roots in the economic game theory literature. Accordingly, it is also safe to say that this literature did not intend to offer a design methodology for engineered systems. This does not mean that material, e.g., as in the many excellent monographs on learning in games [15,37, 38] or evolutionary games [32,35], cannot be a source of methods for engineered systems. Rather, it underscores the importance of recognizing "what is the problem?" when considering this material for a multi-agent approach for designing engineering systems.

That stated, this section presents selected issues that can be distinguishing characteristics of multi-agent learning in engineered designs. In particular, we will see that various notions that play a descriptive or predictive role in the economics literature become design considerations when considering a multi-agent learning approach.

3.1. Defining the game

We have suggested that multi-agent learning may be an effective approach to the problems described earlier. And yet, we have yet to define a specific game in which to apply multi-agent learning methods. Recall that an important motivation for taking a multi-agent perspective was alleviating complexity. This means that the basic elements of players, strategy spaces, and player utilities are all design considerations when imposing a multi-agent framework.

The operations of agents in reactive environments that change with time is an important reality a design methodology must face. The type of game that is chosen to capture such dynamics can be either a one-shot game, a repeated game, or a stochastic game. Choosing a one-shot game implies that we ignore the dynamics of the problem. A repeated game offers a richer structure by reflecting the consequences of a strategy over multiple time steps. This suggests using regret minimizing techniques such as [7]. Stochastic games are a more natural model to model dynamics. Learning in this context is quite complex, as shown by [20], in the context of getting to a Nash equilibrium and by [28] in the context of regret minimization.

Finally, once players and strategy spaces have been specified, another design consideration is defining agent utility functions. As with most design specifications, there is considerable latitude in specifying suitable utility functions. Even in the ideal case of an agreeable centralized objective, there are important considerations in "distributing" this objective among the different players. Ref. [36] contains a general discussion, which is applied to vehicle target assignment in [6].

3.2. Nash equilibrium

There is a significant amount of work on multi-agent learning devoted to characterizing limiting behaviors of various algorithms (e.g., [17]), and in particular, determining whether behavior converges to a Nash equilibrium or to some other desirable target set. Given the historical role of Nash equilibrium as a *predictive* concept of social interactions, it should not be surprising that there are accompanying analytical[1] investigations into how a Nash equilibrium might emerge. Quoting Arrow [5],

"The attainment of equilibrium requires a disequilibrium process".

We believe that the correct context to view work such as [13,14,18,19,21,33] is in terms of understanding how a plausible disequilibrium process can converge to Nash equilibrium.

The importance of Nash equilibrium in engineered systems requires different justifications. The generality of the Nash equilibrium concept is such that a desirable solution can be expressed as a Nash equilibrium, but for *some* appropriately defined game. For various reasons, this "ideal" game is not the setting used in applying the multi-agent learning setup (cf., the preceding discussion on defining the game).

Another issue worth mentioning is the complexity of finding and describing a Nash equilibrium. If the computation of a Nash equilibrium is NP-hard, or even if the complexity is just high, any learning algorithm cannot really hope to get to a Nash equilibrium. For example, in [22] an equilibrium of a certain graphical game is computed via a message-passing algorithm. This algorithm is quite complex, so it seems that a learning scheme that converges to it will have to be at least as complex. This hampers the idea of mitigating complexity by using learning in situations where the complexity of learning a Nash equilibrium is high.

3.3. Performance, convergence, and averages

Much of the work on learning in games concerns characterizing the behavior of various learning mechanisms (cf., [17,38]) in terms of a limiting equilibrium set, e.g., Nash equilibria, correlated equilibria, Hannan consistency set, etc. This is reasonable, particularly in the absence of a specific application domain to assess the performance in terms of domain specific criteria.

If a Nash equilibrium does indeed reflect a desirable operating condition, then learning methods that steer behaviors to equilibrium are well motivated. An example for routing in networks is [10], which derives a routing algorithm that leads to the desirable Wardrop equilibrium, which, in an appropriate context, is a Nash equilibrium. But convergence of learning algorithms should not be a goal by itself from the engineering point of view. Indeed, it is possible that convergence can result in an efficiency *loss*, in terms of player utilities, versus the non-convergent scenario [27].

A related concern is characterizing performance in terms of averages. Concepts such as correlated equilibrium, no regret, and calibration, all reflect asymptotic and averaged measures. In settings that reflect "steady state" operations, then average performance criteria are appropriate.

There are, however, situations in which average performance criteria are not adequate. An obvious example in military operations. In the absence of very large teams, then worst case, or at least risk averse, performance measures are more appropriate. A similar concern for routing in networks is the presence of emergency (911) and other critical network traffic. It is certainly possible to use simulated multi-agent learning off-line as virtual experiments to tune a decision making policy. But even with this usage, there is still a need for a multi-agent learning framework that provides guarantees on something stronger than averaged performance.

3.4. Limited information and costly communication

An important consideration in multi-agent systems and multi-agent learning in particular is the information available to each agent. An example is whether agents have access to the actions of other agents or only their own individual payoffs, thereby distinguishing action-based from payoff-based learning algorithms [38]. Another, somewhat more

---

[1] As well as experimental, e.g., [11].

subtle, issue is whether agents have access to the utility functions of other agents. The lack of such utility information results in so-called "uncoupled" learning and can have important consequences on the resulting limiting behaviors of certain classes of learning dynamics [18]. We should comment that a functional expression of utility is often unavailable for both engineered and economic systems, making on-line payoff-based learning a necessity.

For engineered systems, the information available to each agent is a design consideration that is influenced by the specific domain. This is in stark contrast to a descriptive social model, where the information flow is dictated by the particular modeled setting. A good example is knowledge of the utility function of other agents. In an engineering application, where agents are programmed components, this knowledge can simply be communicated, albeit at some cost, among agents. In social modeling contexts, where utility reflects an agent's underlying intent, such communication of utility cannot be assumed.

3.5. Learning dynamics and bounded rationality

There is a substantial (and growing) number of different adaptation mechanisms for multi-agent learning. For descriptive social models, there is an interest in understanding how agents (humans) learn in games (e.g., [11]). This may explain why most dynamics in the economic literature require very little real-time processing capability. In contrast, in engineered systems agents are programmed components, and so the specification of learning dynamics becomes yet another design choice. Indeed, if one views feedback control systems (e.g., [8]) as a form of sequential decision making, asking whether a controller plausibly reflects human decision making is almost never a consideration.

Along these lines, let us take the notion of "bounded rationality" simply to mean limitations on the processing capacity of individual agents. In this context, the specific application domain dictates the boundaries of bounded rationality, and hence, the set of feasible learning dynamics. Depending on the real time rate of decision making, the result is a wide spectrum of possibilities, ranging from computationally intensive processing to simple decision rules.

Our list of engineering design considerations in multi-agent learning is far from comprehensive, and will continue to grow as long as economic methods continue to be explored for engineering problems. One example is the growing interest, particularly in the networking literature, in applying concepts from mechanism design. Starting from [23] the idea of providing incentives via pricing has become extremely attractive in the theoretical networking community. Recent results (e.g., [31]) present a simple mechanism leading to bounded efficiency loss with respect to social optimality. Still, engineering such a mechanism seems like a formidable task.

4. Concluding remarks: The engineering agenda

As we argued above, the engineering agenda in multi-agent systems is rather different than the economic game theory agenda. In this section, we describe two additional aspects that are particularly important for engineered systems: *robustness* and *domain knowledge*.

Robustness, in the classical sense of control theory, means the resiliency of a system to variation in the parameters of nature or the expected input. In order for multi-agent learning to be used in real engineering systems it must possess some robustness properties. There are several aspects of robustness that are important for engineered systems. First, robustness implies that there are some performance guarantees that would hold even under the worst circumstances. This can be done, perhaps, using the online learning scheme [12] where the regret is minimized and some performance guarantees are provided. Second, robustness implies that one can guarantee performance with respect to some set of *specifications*. This means that if we restrict the set of opponents and the actual environment to belong to a certain set of possible opponents and environments, we can guarantee some performance levels. Finally, robustness is perhaps most important in the dynamical setup, where there might be transient disturbances. In this case, a design goal may be to guarantee that the performance as measured per time period never deteriorates below a certain level.

Domain knowledge is a vital element of systems engineering specification. Currently available multi-agent learning algorithms are typically extremely simple and they do not take domain knowledge into account. While this might be appropriate for the prescriptive agenda, some engineering systems allow the agents to use considerable resources. Efficiently using domain knowledge in the design process of multi-agent learning algorithms is critical to making these algorithms effective and, in particular, to scaling them up beyond illustrative academic problems. Efficient incorporation of domain knowledge in planning and in artificial intelligence in general is a long standing challenge. Due to the complexity of the interaction between multiple decision makers, developing rigorous methods to describe

the specifications of multi-agent systems is crucial to understanding, analyzing, and simulating multi-agent systems. Possible specification technologies can range from symbolic languages to mathematical models. The specification of domain knowledge should represent a tradeoff between the amount of details needed to describe the model and interactions reliably and the complexity of the description.

We close with the comment that an article, such as this one, that dwells on caveats and challenges can easily come across as overly pessimistic. Quite to the contrary, we do believe that the multi-agent learning framework, stemming from the economic game theory literature, offers considerable promise for important engineering problems, and we look forward to seeing this direction continuing to blossom in the years to come.

References

[1] R.K. Ahuja, A. Kumar, K. Jha, J.B. Orlin, Exact and heuristic methods for the weapon-target assignment problem, Technical Report #4464-03, MIT, Sloan School of Management Working Papers, 2003.

[2] R. Akella, P.R. Kumar, Optimal control of production rate in a failure-prone manufacturing systems, IEEE Transactions on Automatic Control 31 (2) (1986) 116–126.

[3] E. Altman, T. Boulogne, R. El Azouzi, T. Jiménez, L. Wynter, A survey on networking games in telecommunications, Computers and Operations Research 33 (2) (2005) 286–311.

[4] E. Altman, N. Shimkin, Individual equilibrium and learning in processor sharing systems, Operations Research 46 (1998) 776–784.

[5] K.J. Arrow, Rationality of self and others in an economic system, The Journal of Business 59 (4) (1986) S385–S399.

[6] G. Arslan, J.R. Marden, J.S. Shamma, Autonomous vehicle-target assignment: A game theoretical formulation, http://www.seas.ucla.edu/~shamma, 2006.

[7] P. Auer, N. Cesa-Bianchi, Y. Freund, R.E. Schapire, The nonstochastic multiarmed bandit problem, SIAM Journal of Computing 32 (1) (2002) 48–77.

[8] T. Basar (Ed.), Control Theory: Twenty-Five Seminal Papers (1932–1981), Wiley, New York, 2000.

[9] R.W. Beard, T.W. McLain, M.A. Goodrich, E.P. Anderson, Coordinated target assignment and intercept for unmanned air vehicles, IEEE Transactions on Robotics and Automation 18 (6) (2002) 911–922.

[10] V.S. Borkar, P.R. Kumar, Dynamic Cesaro–Wardrop equilibration in networks, IEEE Transactions on Automatic Control 48 (3) (2003) 382–396.

[11] C.F. Camerer, Behavioral Game Theory: Experiments in Strategic Interaction, Princeton University Press, Princeton, NJ, 2003.

[12] N. Cesa-Bianchi, G. Lugosi, Prediction, Learning, and Games, Cambridge University Press, New York, 2006.

[13] D.P. Foster, R. Vohra, Calibrated learning and correlated equilibrium, Games and Economic Behavior 21 (1997) 40–55.

[14] D.P. Foster, H.P. Young, Learning, hypothesis testing, and Nash equilibrium, Games and Economic Behavior 45 (2003) 73–96.

[15] D. Fudenberg, D.K. Levine, The Theory of Learning in Games, MIT Press, Cambridge, MA, 1998.

[16] S.B. Gershwin, Manufacturing Systems Engineering, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[17] S. Hart, Adaptive heuristics, Econometrica 73 (5) (2005) 1401–1430.

[18] S. Hart, A. Mas-Colell, Uncoupled dynamics do not lead to Nash equilibrium, American Economic Review 93 (5) (2003) 1830–1836.

[19] S. Hart, A. Mas-Colell, Stochastic uncoupled dynamics and Nash equilibrium, Preprint, http://www.ma.huji.ac.il/~hart/abs/uncoupl-st.html, 2004.

[20] J. Hu, M. Wellman, Nash Q-learning for general-sum stochastic games, Journal of Machine Learning Research 4 (2003) 1039–1069.

[21] S.M. Kakade, D.P. Foster, Deterministic calibration and Nash equilibrium, in: J. Shawe-Taylor, Y. Singer (Eds.), Proceedings of the 17th Annual Conference on Learning Theory, 2004, pp. 33–48.

[22] M.J. Kearns, M.L. Littman, S.P. Singh, Graphical models for game theory, in: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, 2001, pp. 253–260.

[23] F.P. Kelly, Charging and rate control for elastic traffic, European Transactions on Telecommunications 8 (1997) 33–37.

[24] J. Kimemia, S.B. Gershwin, An algorithm for the computer control of a flexible manufacturing system, IIE Transactions 15 (4) (1983) 353–362.

[25] P.R. Kumar, Re-entrant lines, Queueing Systems: Theory and Applications 13 (1993) 87–110.

[26] R. La, V. Anantharam, Optimal routing control: Repeated game approach, IEEE Transactions on Automatic Control 47 (3) (2002) 437–450.

[27] S. Mannor, J.S. Shamma, G. Arslan, Online calibrated forecasts: Memory efficiency vs universality for learning in games, Machine Learning Journal, special issue on "Learning and Computational Game Theory", published online September 2006.

[28] S. Mannor, N. Shimkin, The empirical Bayes envelope and regret minimization in competitive Markov decision processes, Mathematics of Operations Research 28 (2) (2003) 327–345.

[29] R.A. Murphey, Target-based weapon target assignment problems, in: P.M. Pardalos, L.S. Pitsoulis (Eds.), Nonlinear Assignment Problems: Algorithms and Applications, Kluwer Academic Publishers, Dordrecht, 1999, pp. 39–53.

[30] A. Orda, R. Rom, N. Shimkin, Competitive routing in multi-user communication networks, IEEE/ACM Trans. Networking 1 (5) (1993) 510–521.

[31] T. Roughgarden, Selfish Routing and the Price of Anarchy, MIT Press, Cambridge, MA, 2005.

[32] L. Samuelson, Evolutionary Games and Equilibrium Selection, MIT Press, Cambridge, MA, 1997.

[33] J.S. Shamma, G. Arslan, Dynamic fictitious play, dynamic gradient play, and distributed convergence to Nash equilibria, IEEE Transactions on Automatic Control 50 (3) (2005) 312–327.

[34] Y. Shoham, R. Powers, T. Grenager, If multi-agent learning is the answer, what is the question? Artificial Intelligence 171 (7) (2007) 365–377, this issue.
[35] J.W. Weibull, Evolutionary Game Theory, MIT Press, Cambridge, MA, 1995.
[36] D. Wolpert, K. Tumer, An overview of collective intelligence, in: J.M. Bradshaw (Ed.), Handbook of Agent Technology, AAAI Press/MIT Press, 1999.
[37] H.P. Young, Individual Strategy and Social Structure, Princeton University Press, Princeton, NJ, 1998.
[38] H.P. Young, Strategic Learning and its Limits, Oxford University Press, Oxford, 2006.