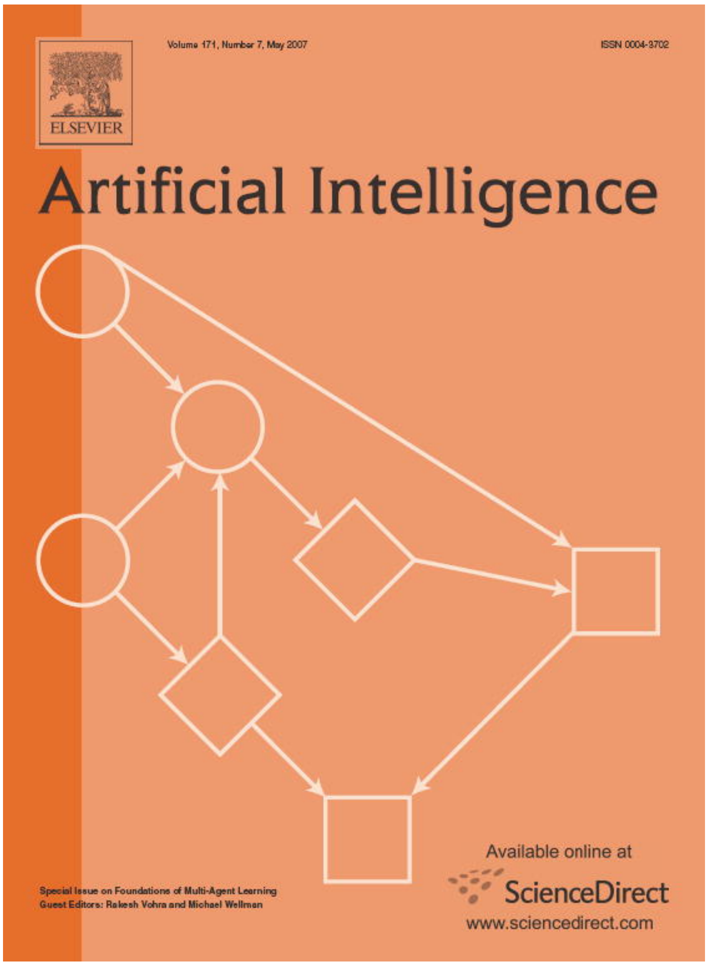
Volume 171, Number 7, May 2007
ISSN 0004-3702
ELSEVIER
Artificial Intelligence
Special Issue on Foundations of Multi-Agent Learning
Guest Editors: Rakesh Vohra and Michael Wellman
Available online at
ScienceDirect
www.sciencedirect.com

Available online at www.sciencedirect.com

ScienceDirect

Artificial Intelligence 171 (2007) 382–391

Artificial Intelligence

www.elsevier.com/locate/artint

Perspectives on multiagent learning ☆

Tuomas Sandholm

Carnegie Mellon University, Computer Science Department, Pittsburgh, PA 15213, USA

Received 18 May 2006; received in revised form 27 February 2007; accepted 27 February 2007

Available online 30 March 2007

Abstract

I lay out a slight refinement of Shoham et al.'s taxonomy of agendas that I consider sensible for multiagent learning (MAL) research. It is not intended to be rigid: senseless work can be done within these agendas and additional sensible agendas may arise. Within each agenda, I identify issues and suggest directions. In the *computational agenda*, direct algorithms are often more efficient, but MAL plays a role especially when the rules of the game are unknown or direct algorithms are not known for the class of games. In the *descriptive agenda*, more emphasis should be placed on establishing what classes of learning rules actually model learning by multiple humans or animals. Also, the agenda is, in a way, circular. This has a positive side too: it can be used to verify the learning models. In the *prescriptive agendas*, the desiderata need to be made clear and should guide the design of MAL algorithms. The algorithms need not mimic humans' or animals' learning. I discuss some worthy desiderata; some from the literature do not seem well motivated. The learning problem is interesting both in cooperative and noncooperative settings, but the concerns are quite different. For many, if not most, noncooperative settings, future work should increasingly consider the learning itself strategically.

Lower bounds cut across the agendas. They can be derived on the computational complexity and on the number of interactions needed.

Keywords: Multiagent learning; Learning in games; Reinforcement learning; Game theory

1. Introduction

Learning from experience is a key capability because one may not be able to devise a good strategy (a plan for all possible contingencies) in advance—even with the help of computers. In multiagent settings, learning may be needed because the opponents' strategies are unknown, because the rules of the game are unknown, or because it is computationally too complex to solve for a good strategy with other means. *Multiagent learning* (*MAL*) is complicated by the fact that the other agents may be learning as well (or changing their exploration behavior [61]), thus making the environment nonstationary for a learner. MAL has been studied with different objectives as well as with different restrictions on the game and on what the learners can observe.

☆ This work was supported by the National Science Foundation under ITR grants IIS-0121678 and IIS-0427858, and a Sloan Fellowship.

E-mail address: sandholm@cs.cmu.edu.

0004-3702/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.artint.2007.02.004

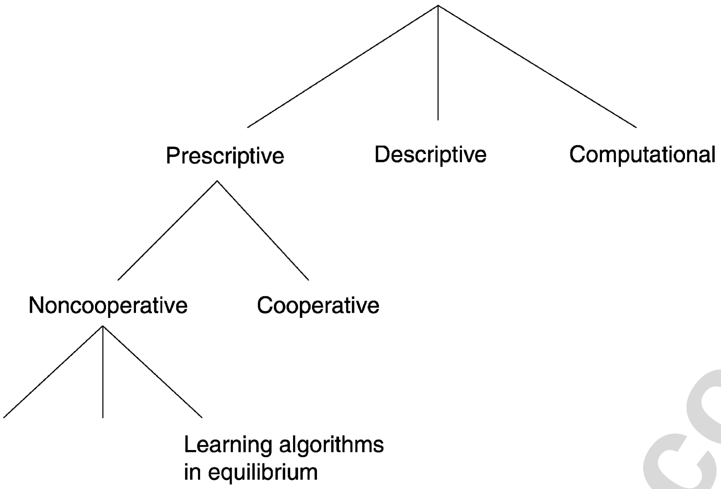

Fig. 1. Taxonomy of agendas for multiagent learning research.

2. On agendas for multiagent learning

In their paper "If multi-agent learning is the answer, then what is the question?", Shoham et al. make an important contribution by laying out five agendas of (and for) MAL. I present a refinement of that taxonomy in Fig. 1. It is consistent with their view, but adds hierarchy (and renames the "normative" agenda as "learning algorithms in equilibrium"). In the rest of this section, I will comment on the agendas using this taxonomy.

2.1. Computational agenda

While some MAL algorithms can, at least in principle, be used for equilibrium finding, they tend to be significantly less efficient than the best direct equilibrium-finding algorithms. For example, MAL algorithms have been able to solve only tiny poker games while direct techniques have been able to find the exact equilibrium of poker games up to, and including, the game of Rhode Island Hold'em, which has 3.1 billion nodes in the game tree [32].

If one's goal is equilibrium finding, there seems to be little reason to use MAL algorithms. There has been tremendous recent progress on faster algorithms for equilibrium finding in normal form games [58,62], graphical games [42, 71], sequential games of perfect information [64], and sequential games of imperfect information [31,32]. Why would one not want to take advantage of these efficient, direct techniques?

One would, but there are settings for which efficient direct techniques are not (yet) known. For example, the best computer programs for Backgammon have been developed using MAL, specifically Q-learning [69]. Also, MAL algorithms are in many cases simpler to program than the direct techniques, and are thus arguably preferable if code for the direct algorithm is not readily available and the scalability of MAL suffices for the setting. Finally, it is conceivable that for some equilibrium-finding problems, MAL-based algorithms will turn out to be the most efficient. For example, for finding a minimax solution to a two-player zero-sum game, MAL (specifically, no-regret learning) might be an efficient alternative to linear programming if the game has a huge number of (pure) strategies but the payoffs are small [2, page 73].

Most importantly, however, MAL algorithms are needed if the agents do not know the structure (e.g., payoffs) of the game, or if the goal is to exploit a weak opponent more than an equilibrium strategy would exploit that opponent.

2.2. Descriptive agenda

The descriptive agenda of MAL grew largely out of the concern that people might not have the required rationality (reasoning capability) to act according to game-theoretic equilibrium. This in turn calls into question the descriptive power of game theory. To mitigate this, economists have studied whether MAL techniques lead to equilibrium play.[1] If so, that is a justification of equilibrium because it is reached with learning, and thus does not require sophisticated game-theoretic reasoning and direct equilibrium-finding algorithms. This simplicity argument can also be used for equilibrium selection (easy-to-learn equilibria being more likely and more reasonable) and for justifying/selecting solution concepts (concepts that can be associated with convergent learning algorithms being more descriptive). Humans' and animals' learning other than learning that yields equilibrium is also in the scope of the descriptive MAL agenda.

There are, however, critical shortcomings in the descriptive MAL agenda, at least in many of the ways in which it has been pursued to date. The most important shortcoming is that it is ill-defined what counts as a MAL algorithm under this agenda, and the descriptive conclusions depend completely on what algorithms count. For example, elaborate work has been done on MAL algorithms that converge to equilibrium. However, even a trivial algorithm achieves this goal: the players first explore each cell of the payoff matrix in order, then each of them individually computes an equilibrium (each agent using the same algorithm so as to find the same equilibrium in case of multiple equilibria), and they play that equilibrium forever from then on. Researchers pursuing the descriptive agenda would not consider this straw man a MAL algorithm. Why not? Unfortunately, clear definitions of what algorithms count are usually not articulated. Instead, a variety of algorithms have been proposed, and their "naturalness" is argued based on intuition and taste on an algorithm-by-algorithm basis. Since the descriptive conclusions hinge on what algorithms count, I think there should be clear definitions of what MAL algorithms count. The definitions should preferably admit and exclude classes of algorithms—defined by some properties of the algorithmic steps—rather than individual algorithms.

What algorithms are admitted versus excluded should be guided at least by experiments on humans (or animals if that is the population about which descriptive conclusions are to be drawn). Unfortunately, this gives rise to a potential circularity in the agenda: in order to make descriptive conclusions about how humans behave, we need to start out with an understanding of how humans behave. (The circularity could also be used constructively to try to verify that the approach is working. One could observe behavior in one setting, select learning models accordingly, test the models in a different setting, and check whether the descriptive conclusions align with observed behavior in that setting.) Another downside is that experimental results are context dependent.

One approach is to admit in the definition algorithms that are simple (as opposed to complex to execute). While this is intuitively appealing, it suffers from difficulties. First, it is not clear what is simple because human reasoning may not coincide with the Turing machine model for which clear complexity measures have been devised. Second, some of the MAL algorithms admitted in the descriptive approach today involve numerous sophisticated calculations.

Another shortcoming in the descriptive agenda is that most of that work assumes that people use the learning rule even if they would do better for themselves by behaving differently during the learning process. In other words, people are reduced, by assumption, to following a certain learning algorithm which guarantees that, *if* each agent follows the algorithm, (eventually) an equilibrium will be reached. A related shortcoming is that each person is assumed to follow the learning rule even if he could, by behaving differently, drive the population to converge to a different equilibrium that is more beneficial for himself.

2.3. Prescriptive agendas

In the prescriptive agendas we are interested in how multiple agents *should* learn. It is thus largely irrelevant whether the learning algorithms mimic how humans (or other animals) learn. The goal is to develop learning algorithms that satisfy given desiderata, that is, do well against given performance measures. It is therefore paramount that the desiderata be important and that they be specified clearly upfront. This is in contrast to designing a MAL algorithm first, and then defining idiosyncratic desiderata to justify the algorithm.

The commonly used desideratum of convergence of the empirical distribution of play to Nash equilibrium does not seem meaningful: play might be far from optimal at every step, and the stability guarantees of equilibrium are not present (the agents might be far from equilibrium at every step). I also do not see the desideratum of guaranteeing

[1] For example, Kalai and Lehrer [41] show that if each agent knows its own payoff matrix in a repeated game, and rationally updates its beliefs about others' strategies, the agents converge to Nash equilibrium—if every measurable set of outcomes that has positive probability under their actual strategies has positive probability under each agent's prior belief. It has been argued that this last requirement is unreasonably restrictive [53, 54]; in some games such learning never comes close to equilibrium [25].

"satisficing" performance (utility within a bound of optimal) being highly important; why would an agent not want more?

In our experience, a fruitful approach within the prescriptive agendas is to use the desiderata to guide algorithm development. This tends to lead to algorithms that are not as intuitive as some more obvious choices, but which perform better. I will include examples of this in the next two subsections where I discuss two strands of the prescriptive agenda.

2.3.1. *Prescriptive, cooperative agenda*

The prescriptive, cooperative agenda is about team games: the agents' payoffs are identical, so their incentives are aligned. Why is this a multiagent setting? Why not consider all the agents to be one agent? Indeed, some team problems to which MAL has been applied would probably have been better and/or more easily solved using some centralized approach. For example, when designing a controller for a bank of elevators, why would one consider each elevator as a separate agent? On the other hand, some applications induce an inherent distribution of the problem. The data may be inherently distributed and cannot be centralized because there is too much of it or because it changes too quickly. Certain problems in computer networks constitute a typical class of settings (e.g., [14]). In such settings, modeling the problem as a multiagent problem may make sense even if the agents have identical payoffs.

If the game is known, the team setting is not really a learning setting. Each agent can, at least in principle, compute the optimal joint policy and use it. (If each agent uses the same deterministic algorithm, all agents will find the same optimal policy, so multiplicity of optima is not an issue.) However, computing the optimal joint policy can be complex in some settings—such as in the control of artificial predator teams [63,68], or robot soccer [67] even if the opponent team's strategy were known—so using some learning algorithm to come up with a joint policy may be a reasonable alternative. This establishes a connection to the computational MAL agenda. The team setting is also made simpler by the fact that, if the same designer gets to design all the agents, we need not worry that some of the agents might not follow the prescribed learning technique.

If the game is unknown, the problem is nevertheless highly nontrivial because it involves both of the following learning tasks:

- learning the game, and
- learning to play.

The agents have to (simultaneously) identify the game and maximize utility (e.g., discounted sum of rewards). (It may be possible to learn enough about the game to play well without building an explicit model of it.) Another complication is that even team games can have multiple equilibria, some suboptimal.

Research on this has an interesting history. In repeated games, learning automata [55,70] converge to an equilibrium which is a local optimum. In stochastic (aka Markov) games the problem is more difficult. Littman [47] introduced *Friend-or-Foe Q-learning*, which learns to play Nash equilibrium if the overall stochastic game has a global optimum (a strategy profile that maximizes payoff for each agent) or a saddle point (Nash equilibrium such that if any agent deviates, all others become better off). The algorithm needs to know which of these two settings it is operating in (in each, it suffices to learn one's own Q-function). The algorithm variants are called *Friend-Q* and *Foe-Q* (aka *minimax-Q*), respectively. The same two settings have been the scope of theoretical convergence results also for an earlier MAL algorithm, *Nash-Q* [38]. Those convergence guarantees require that the global optimality property or saddle point property is not only satisfied by the overall stochastic game but also by the internal representation of what has been learned so far (table of Q-values) [38,47]. This condition can be satisfied in practice if the algorithm is told whether it is in the global optimum setting or the saddle point setting, so it knows to use the appropriate updating rule. Without being told this, *Nash-Q* converges if the setting is so restricted that each stage game has a unique equilibrium [47] (or multiple equilibria that all have the same payoff vector, as in zero-sum games). Note that the saddle point setting is adversarial; in this sense these algorithms pertain to the prescriptive noncooperative agenda (discussed below) as well. Claus and Boutilier [16] developed the *joint action learner*, where the agents converge to a Nash equilibrium in team games, but that outcome may be a suboptimal equilibrium (even in repeated, rather than stochastic, team games). Learning to play an optimal Nash equilibrium in team Markov games was then posed as an important open problem [48]. Wang and Sandholm [72] developed *optimal adaptive learning*, which provably accomplishes this goal. That work exemplified the lesson that the desideratum—not idiosyncratic notions of "naturalness"—should guide algorithm design. The algorithm encompasses unintuitive features: *incomplete sampling of one's own memory*, and action selection that is *biased in a particular way* (apart from the stochasticity needed for exploration). Both of these are essential for that algorithm to meet the desideratum. Most recently, near-optimal polynomial-time algorithms for the problem were presented [10].

2.3.2. Prescriptive, noncooperative agenda

The prescriptive, noncooperative agenda is more widely applicable, but also significantly more challenging: the agents' nonidentical payoffs cause them to have different incentives. (The special case of constant-sum (i.e., purely adversarial) games [46] is easier than the general-sum case [61].) It is not even clear what the goal of MAL should be in this setting.

Much of the work has focused on learning to play equilibrium. While this is a sensible desideratum, it is not omnipotent. For example, if other agents are not playing equilibrium, my agent may be better off deviating from an equilibrium strategy. Bowling and Veloso [8] therefore suggested that a learning algorithm should have both of the following properties:

P1: convergence to Nash equilibrium in self-play.

Remarks. One could also postulate other notions of equilibrium as the target. For example, several papers have been written on MAL with correlated equilibrium as the target (e.g., [33]).

Furthermore, one could postulate as the target not an equilibrium of the stage game, but an equilibrium of the remaining repeated game. On one hand, such equilibria can yield better outcomes (such as in the Iterated Prisoner's Dilemma [61]). On the other hand, in many settings it is really stage game play that one is trying to learn. For example, when practicing soccer, chess, or Go with the same opponent over a course of a year or more, one's goal usually is to do as well as possible at the end. Also, what constitutes equilibrium in a repeated game depends on the agents' discounting, which might not be common knowledge.

P2: convergence to a best-response against any stationary opponent (or even any opponent that eventually becomes stationary).

Remarks. This is the agent learning to maximally exploit irrational (stationary) opponents. In more cooperative games this would be better worded as learning to do well despite the other agents' irrationality.

Alternatively, one might wish to learn to best-respond against larger classes of opponents [59,60].

One could view these properties as minimal desiderata in the sense that it would be desirable of any MAL algorithm to satisfy at least these. One might desire more, e.g., strengthen P1 to convergence to a Pareto efficient equilibrium.

The *WoLF-IGA* algorithm [8] (an improvement over an earlier algorithm [65]) uses a higher learning rate when losing and a lower one when winning. It satisfies P1 and P2 in 2-person 2-action repeated games, assuming that the agents can observe each others' mixed strategies and assuming that the gradient-following steps are infinitesimal.

The *AWESOME* (*Adapt When Everybody is Stationary, Otherwise Move to Equilibrium*) algorithm [22] satisfies P1 and P2 in n-person n-action repeated games, without assuming that mixed strategies are observable, and without using infinitesimal steps. It again showed that the desiderata should drive MAL algorithm design. The idea behind *AWESOME* is to adapt to the other agents' strategies when they appear stationary, but otherwise to retreat to a precomputed equilibrium strategy. At any point in time, *AWESOME* maintains either of two null hypotheses: that the others are playing the precomputed equilibrium, or that they are stationary. Whenever both of these hypotheses are rejected, *AWESOME* restarts. *AWESOME* may reject either hypothesis based on actions played in an *epoch*. Over time, the epoch length is carefully increased and the criterion for hypothesis rejection tightened to obtain the convergence guarantee. *AWESOME* is also self-aware: when it detects that its own actions signal nonstationarity to others, it restarts itself for synchronization. That is, an *AWESOME* player is mindful of what it teaches to others. The techniques used to prove the properties of AWESOME are fundamentally different from those used for prior algorithms, and may help in analyzing future MAL algorithms as well. *AWESOME* can also be viewed as a skeleton—that guarantees the satisfaction of P1 and P2—on top of which additional techniques may be used to guarantee additional desirable properties.

AWESOME still makes some of the assumptions made in the other theoretical work attempting to attain P1 and P2 [3,8,65]. First, it deals with repeated games (i.e., stochastic games with only one state). Second, it assumes the game is known (or has already been learned). This is assumed in much (but not all) of the game theory literature on learning (for a review, see [27]), but a significant amount of MAL research in computer science attempts to have the agents learn the game as well [4,9,11,16,17,19,33,38,46,47,49,57,72,73]. So far that research has not been able to make claims about satisfying P1 and P2. In fact, (for continuous-time dynamics) some knowledge of the other players' payoffs is necessary to converge to Nash equilibrium [37]. If the game is not known initially, but the agents can observe the realized payoffs of all agents, then, given that all the agents are using the same learning algorithm, they could conceivably collaboratively explore the game and learn the game, and then learn how to play. The third assumption is that the agents can compute a Nash equilibrium. (It is assumed that when there are multiple *AWESOME* players, they compute the same Nash equilibrium; e.g., they have identical algorithms.) It is unknown whether a Nash equilibrium can be found in worst-case polynomial time, but certain related questions are hard in the worst case [18, 30]. In practice Nash equilibria can be found for reasonably large games [44,58,62].

The lesson that desiderata should guide MAL algorithm design also emerged in developing efficient, provably near-optimal algorithms for learning to play Pareto-optimal strict Nash equilibria (when agents may prefer different equilibria) in repeated coordination games of nonidentical interest [73]. A key future direction is to develop MAL algorithms that converge to an optimal Nash equilibrium in general-sum repeated games, if possible.

In addition to properties about the end result of learning, one may desire properties of the entire learning process. For example, one might like to have

P3: low regret (e.g., compared to the best fixed strategy, averaged over time).

Several MAL algorithms have the property that their average regret tends to zero, regardless of the opponent(s), in repeated games (e.g., [2,7,26,28,34,36,39,45,74]), but this is unattainable in general stochastic games [50].

The *ReDVaLeR* algorithm [3] satisfies P1 and P2, but relies on agents observing others' mixed strategies, and on infinitesimal steps. For a different setting of a parameter, it achieves constant-bounded regret.

Stepping back: what are the right desiderata here? Consider Heads-Up poker (a 2-player zero-sum game) played repeatedly. If one wants to build an agent that is unbeatable (in expectation over draws of cards), one should compute a minimax strategy [31,32] and use it. If one is interested in accruing as much money as possible, one could do better, at least in principle, by learning to exploit the opponent's irrationality (e.g., [66]). That is risky because the opponent might teach the agent to play in a certain way, and later exploit the agent's learned way of playing. One reasonable approach would be to start by playing a minimax strategy, gradually modifying that strategy to take advantage of the opponent (as the agent learns more about the opponent). To avoid the *"get taught & exploited problem"* mentioned above, the agent might, for example, deviate from minimax play only to the extent that risks (e.g., in expectation against the worst possible opponent for that strategy) an amount equal to the agent's winnings so far. Or, the agent may want to risk more or less. (McCracken and Bowling [51] present a related approach. At each iteration of a game, the strategy is selected from among strategies that yield payoff no worse than the safety level (e.g., minimax payoff) minus a constant. If a certain opponent modeling algorithm is used for selecting within that set and for adjusting the constant, then in the limit the average payoff is no worse than the safety level.) In summary, in the type of setting where the payoffs are real (e.g., money) already during the learning process, one should also *view the learning process itself strategically*.[2]

Algorithms whose average regret tends to zero address this to an extent. In zero-sum games such an algorithm will play nearly optimally against a rational (minimax) opponent and nearly optimally against any stationary opponent (or stationary population of indistinguishable stationary opponents). However, such algorithms are not completely satisfactory. First, regret can grow without bound, even if average regret tends to zero. Second, average regret does not go to zero immediately, but rather approaches zero at certain rates. Third, zero average regret does not imply that the agents have learned to play Nash equilibrium. These learners are generally oblivious to other players' payoffs. Thus they cannot reason about others' incentives and steer the interaction toward a win-win, use threats, etc. (However,

[2] For settings where one player (in a zero-sum repeated game with potentially stochastic payoffs) does not know the game (payoffs) but has to learn them in order to play well, Conitzer and Sandholm [17] introduced a framework (BL-WoLF) for analyzing the inherent disadvantage to that player. The framework allows for probabilistic and approximate learning.

in certain classes of games low average regret does imply ϵ-Nash-equilibrium [6].) Fourth, even if each agent has zero average regret, the solution might be far from Pareto optimal. Fifth, on games with large numbers of stage-game pure strategies—such as poker (except tiny artificial variants)—it is not clear whether such algorithms are helpful (or even implementable from a computational perspective). If the algorithm learns at each repetition of the game what its expected payoff would have been for each pure strategy, ϵ average regret can be guaranteed while conducting a number of repetitions proportional to $\log n$ (and no faster in general), where n is the number of the pure strategies (e.g., [26]). In some applications, n is huge or infinite; in certain structured domains (such as adaptively choosing paths in a network), the strategy space can nevertheless be compactly represented and optimized over, yielding faster learning speed [40,74]. Unfortunately, in many applications the algorithm only observes its payoff for the strategy that it actually played (aka the bandit setting) [2]. In that setting, the number of repetitions needed by the best algorithms is $O(n \log n)$, and any algorithm will require $\Omega(n)$ iterations because it needs to play each strategy at least once. Again, in certain applications where n is huge or even infinite, the strategy space can nevertheless be compactly represented and optimized over, yielding faster learning [24,52]. This is an exciting future direction.

2.3.3. Agenda of learning algorithms in equilibrium

One conceptual difficulty, even with no-regret algorithms, is that a player can do better for himself by using some different repeated-game strategy. So, why would an agent use the given learning strategy? (Furthermore, even if the given strategies lead to an equilibrium, there might be multiple equilibria, and the agent might be able to drive the system to a better equilibrium for himself by playing a different strategy.) This motivates the idea that the learning algorithms should themselves be in equilibrium. I suggested this at a luncheon at the IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems. Independently, Brafman and Tennenholtz [12,13] have pursued such an agenda and made significant progress.

On one hand, this is a holy grail of the prescriptive, noncooperative agenda. On the other hand, it is still frail. If at least one agent fails to play the prescribed learning strategy, others may be better off playing differently themselves. Second, the work assumes that agents' ways of discounting are known. Third, there can be multiple equilibria of the learning algorithms; which should be picked?

Stepping back: why is this learning rather than computing an equilibrium of a repeated game? One possibility is that payoff matrices are unknown. Brafman and Tennenholtz [12,13] proposed *efficient learning equilibrium* for this: the learning algorithms must be in equilibrium, deviations must become irrational after a polynomial number of steps, and payoffs must approach those of a Nash equilibrium after a polynomial number of steps if everyone sticks to the learning algorithm. Such equilibria exist for a surprisingly broad class of settings.

Future work should strive for MAL algorithms that achieve some desirable properties related to viewing the learning process itself strategically—perhaps along the lines discussed above—and (versions of) P1, P2, and/or P3.

3. Research that cuts across agendas: Lower bounds

There is at least one research direction that cuts across the agendas: lower bounds. They can be derived on the computational complexity of finding a solution (satisfying a given solution concept). These bounds apply to the computational complexity of MAL algorithms as well, because those algorithms can be used to find a solution. The complexity of finding a Nash equilibrium is unknown, but it is the same (polynomially equivalent) for n-person games with rational numbers as payoffs and for 2-person games with 0/1 payoffs [1,15]. Finding good Nash equilibria is hard, as are many related targets, even in 2-player games [18,30]. Good correlated equilibria are easy to find in 2-player games [30] and many concisely represented n-person games [56]. The complexity of several other solution concepts has also been determined [5,18,20,21,23,29,43]. One caveat is that the computational lower bounds are usually derived under the Turing machine model of computation, and it is conceivable that humans and animals have a more powerful computational apparatus, in which case these lower bounds might not apply to the descriptive agenda.

Another important complexity measure of MAL is the number of interactions (rounds), or cost, needed to find a solution. Conitzer and Sandholm [19] proposed a methodology based on communication complexity for deriving lower bounds for this. They derived bounds (as a function of the number of actions available in the game) for several solution concepts. Lower bounds derived with that methodology apply to *all* MAL algorithms (when opponents' payoffs are unknown), and thus establish *inherent* limitations of MAL. The bounds do not rely on any particular model of computation, so they apply to the descriptive agenda as well. Recently, this methodology was used also for n-agent games [35].

4. Conclusions

I laid out a slight refinement of Shoham et al.'s taxonomy of agendas that I consider sensible for multiagent learning (MAL). My goal is to spur discussion and research, and perhaps ultimately provide the field more clarity. The taxonomy is not intended to be rigid: senseless work can be done within these agendas, and new sensible agendas may emerge. Within each agenda, I briefly identified some issues and suggested some directions that future research should take.

In the computational agenda, direct algorithms are often more efficient than MAL, but MAL plays a role especially when the rules of the game are unknown or efficient direct algorithms are not known for the class of games.

In the descriptive agenda, more emphasis should be placed on establishing what classes of learning rules actually model learning by multiple humans or animals. Also, the agenda has a potential circularity. This has a positive side too: it can be used to verify the learning models.

In the prescriptive agendas, desiderata need to be clear and should guide the MAL algorithm design. The algorithms need not mimic humans' or animals' learning. I discussed some worthy desiderata, and pointed out that some used in the literature do not seem to be well motivated. The learning problem is interesting both in cooperative and noncooperative settings, but the issues and concerns are quite different. For many, if not most, noncooperative settings, future work should increasingly consider the learning itself strategically.

Lower bounds cut across the agendas. Lower bounds can be derived on the computational complexity of learning a solution (assuming, for example, the Turing machine model). Lower bounds can also be derived on the number of interactions (or cost) needed to learn a solution, using tools from communication complexity. These latter bounds are independent of the computational model, so they apply to human and animal learning as well.

Acknowledgements

I thank Rakesh Vohra and Michael Wellman for organizing this important special issue. I thank them and an anonymous reviewer for helpful suggestions. I thank Avrim Blum for interesting communications about no regret learning, and Michael Littman for an illuminating discussion on Nash-Q and Friend-or-Foe-Q.

References

[1] T. Abbott, D. Kane, P. Valiant, On the complexity of two-player win-lose games, in: Symposium on Foundations of Computer Science, 2005.

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

[3] B. Banerjee, J. Peng, Performance bounded reinforcement learning in strategic interactions, in: National Conf. on Artificial Intelligence, 2004.

[4] B. Banerjee, S. Sen, J. Peng, Fast concurrent reinforcement learners, in: Internat. Joint Conf. on Artificial Intelligence, 2001.

[5] M. Benisch, G. Davis, T. Sandholm, Algorithms for rationalizability and CURB sets, in: National Conf. on Artificial Intelligence, 2006.

[6] A. Blum, E. Even-Dar, K. Ligett, Routing without regret: On convergence to Nash equilibria of regret-minimizing algorithms in routing games, in: ACM Symposium on Principles of Distributed Computing, 2006.

[7] M. Bowling, Convergence and no-regret in multiagent learning, in: Conf. on Neural Information Processing Systems, 2005.

[8] M. Bowling, M. Veloso, Multiagent learning using a variable learning rate, Artificial Intelligence 136 (2002) 215–250.

[9] R. Brafman, M. Tennenholtz, A near-optimal polynomial time algorithm for learning in certain classes of stochastic games, Artificial Intelligence 121 (2000) 31–47.

[10] R. Brafman, M. Tennenholtz, Learning to coordinate efficiently: A model-based approach, Journal of Artificial Intelligence Research 19 (2003) 11–23.

[11] R. Brafman, M. Tennenholtz, R-max—a general polynomial time algorithm for near-optimal reinforcement learning, Journal of Machine Learning Research 3 (2003) 213–231.

[12] R. Brafman, M. Tennenholtz, Efficient learning equilibrium, Artificial Intelligence 159 (2004) 27–47. Earlier version in NIPS-02.

[13] R. Brafman, M. Tennenholtz, Optimal efficient learning equilibrium: Imperfect monitoring in symmetric games, in: National Conf. on Artificial Intelligence, 2005.

[14] Y.-H. Chang, T. Ho, L. Kaelbling, Mobilized ad-hoc networks: A reinforcement learning approach, in: Internat. Conf. on Autonomic Computing, 2004.

[15] X. Chen, X. Deng, Settling the complexity of 2-player Nash equilibrium, in: Electronic Colloquium on Computational Complexity, Report No. 150, 2005.

[16] C. Claus, C. Boutilier, The dynamics of reinforcement learning in cooperative multiagent systems, in: National Conf. on Artificial Intelligence, 1998.

[17] V. Conitzer, T. Sandholm, BL-WoLF: A framework for loss-bounded learnability in zero-sum games, in: Internat. Conf. on Machine Learning, 2003.

[18] V. Conitzer, T. Sandholm, Complexity results about Nash equilibria, in: Internat. Joint Conf. on Artificial Intelligence, 2003.

[19] V. Conitzer, T. Sandholm, Communication complexity as a lower bound for learning in games, in: Internat. Conf. on Machine Learning, 2004.

[20] V. Conitzer, T. Sandholm, Complexity of (iterated) dominance, in: ACM Conf. on Electronic Commerce, 2005.

[21] V. Conitzer, T. Sandholm, A generalized strategy eliminability criterion and computational methods for applying it, in: National Conf. on Artificial Intelligence, 2005.

[22] V. Conitzer, T. Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents, in: Special issue on Learning and Computational Game Theory, Machine Learning 67 (2007) 23–43. Short version in ICML-03.

[23] V. Conitzer, T. Sandholm, Computing the optimal strategy to commit to, in: ACM Conf. on Electronic Commerce, 2006.

[24] A. Flaxman, A. Kalai, B. McMahan, Online convex optimization in the bandit setting: Gradient descent without a gradient, in: ACM-SIAM Symposium on Discrete Algorithms, 2005.

[25] D.P. Foster, H.P. Young, On the impossibility of predicting the behavior of rational agents, Proceedings of the National Academy of Sciences 98 (2001) 12848–12853.

[26] Y. Freund, R. Schapire, Adaptive game playing using multiplicative weights, Games and Economic Behavior 29 (1999) 79–103.

[27] D. Fudenberg, D. Levine, The Theory of Learning in Games, MIT Press, 1998.

[28] D. Fudenberg, D.K. Levine, Consistency and cautious fictitious play, Journal of Economic Dynamics and Control 19 (1995) 1065–1089.

[29] I. Gilboa, E. Kalai, E. Zemel, The complexity of eliminating dominated strategies, Mathematics of Operation Research 18 (1993) 553–565.

[30] I. Gilboa, E. Zemel, Nash and correlated equilibria: Some complexity considerations, Games and Economic Behavior 1 (1989) 80–93.

[31] A. Gilpin, S. Hoda, J. Peña, T. Sandholm, Gradient-based algorithms for finding Nash equilibria in extensive form games, Mimeo, 2007.

[32] A. Gilpin, T. Sandholm, Finding equilibria in large sequential games of imperfect information, in: ACM Conf. on Electronic Commerce, 2006.

[33] A. Greenwald, K. Hall, Correlated Q-learning, in: Internat. Conf. on Machine Learning, 2003.

[34] A. Greenwald, A. Jafari, A general class of no-regret learning algorithms and game-theoretic equilibria, in: Conf. on Learning Theory, 2003.

[35] S. Hart, Y. Mansour, The communication complexity of uncoupled Nash equilibrium procedures, 2006, Draft.

[36] S. Hart, A. Mas-Colell, A simple adaptive procedure leading to correlated equilibrium, Econometrica 68 (2000) 1127–1150.

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

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

[39] A. Jafari, A. Greenwald, D. Gondek, G. Ercal, On no-regret learning, fictitious play, and Nash equilibrium, in: Internat. Conf. on Machine Learning, 2001.

[40] A. Kalai, S. Vempala, Efficient algorithms for online decision problems, Journal of Computer and System Sciences 71 (2005) 291–307.

[41] E. Kalai, E. Lehrer, Rational learning leads to Nash equilibrium, Econometrica 61 (5) (1993) 1019–1045.

[42] M. Kearns, M. Littman, S. Singh, Graphical models for game theory, in: Conf. on Uncertainty in Artificial Intelligence, 2001.

[43] D.E. Knuth, C.H. Papadimitriou, J.N. Tsitsiklis, A note on strategy elimination in bimatrix games, Operations Research Letters 7 (1988) 103–107.

[44] C. Lemke, J. Howson, Equilibrium points of bimatrix games, Journal of the Society of Industrial and Applied Mathematics 12 (1964) 413–423.

[45] N. Littlestone, M.K. Warmuth, The weighted majority algorithm, Information and Computation 108 (2) (1994) 212–261.

[46] M. Littman, Markov games as a framework for multi-agent reinforcement learning, in: Internat. Conf. on Machine Learning, 1994.

[47] M. Littman, Friend or foe Q-learning in general-sum Markov games, in: Internat. Conf. on Machine Learning, 2001.

[48] M. Littman, Value-function reinforcement learning in Markov games, Journal of Cognitive Systems Research 2 (2001) 55–66.

[49] M. Littman, C. Szepesvári, A generalized reinforcement-learning model: Convergence and applications, in: Internat. Conf. on Machine Learning, 1996.

[50] 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.

[51] P. McCracken, M. Bowling, Safe strategies for agent modelling in games, in: AAAI Fall Symposium on Artificial Multi-agent Learning, 2004.

[52] B. McMahan, A. Blum, Online geometric optimization in the bandit setting against an adaptive adversary, in: Conf. on Learning Theory, 2004.

[53] J. Nachbar, Prediction, optimization, and learning in games, Econometrica 65 (1997) 275–309.

[54] J. Nachbar, Bayesian learning in repeated games of incomplete information, Social Choice and Welfare 18 (2001) 303–326.

[55] K.S. Narendra, M.A.L. Thathachar, Learning Automata: An Introduction, Prentice Hall, 1989.

[56] C. Papadimitriou, T. Roughgarden, Computing equilibria in multi-player games, in: Symposium on Discrete Algorithms, 2005.

[57] K. Pivazyan, Y. Shoham, Polynomial-time reinforcement learning of near-optimal policies, in: National Conf. on Artificial Intelligence, 2002.

[58] R. Porter, E. Nudelman, Y. Shoham, Simple search methods for finding a Nash equilibrium, in: National Conf. on Artificial Intelligence, 2004.

[59] R. Powers, Y. Shoham, Learning against opponents with bounded memory, in: Internat. Joint Conf. on Artificial Intelligence, 2005.

[60] R. Powers, Y. Shoham, New criteria and a new algorithm for learning in multi-agent systems, in: Conf. on Neural Information Processing Systems, 2005.

[61] T. Sandholm, R. Crites, Multiagent reinforcement learning in the iterated prisoner's dilemma, Biosystems 37 (1996) 147–166, special issue on the Prisoner's Dilemma. Early version in IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems.

[62] T. Sandholm, A. Gilpin, V. Conitzer, Mixed-integer programming methods for finding Nash equilibria, in: National Conf. on Artificial Intelligence, 2005.

[63] T. Sandholm, M.V. Nagendra Prasad, Learning pursuit strategies, Project for CmpSci 698 Machine Learning, Computer Science Department, University of Massachusetts at Amherst, Spring, 1993.

[64] J. Schaeffer, Y. Björnsson, N. Burch, A. Kishimoto, M. Müller, R. Lake, P. Lu, S. Sutphen, Solving checkers, in: Internat. Joint Conf. on Artificial Intelligence, 2005.
[65] S. Singh, M. Kearns, Y. Mansour, Nash convergence of gradient dynamics in general-sum games, in: Conf. on Uncertainty in Artificial Intelligence, 2000.
[66] F. Southey, M. Bowling, B. Larson, C. Piccione, N. Burch, D. Billings, C. Rayner, Bayes' bluff: Opponent modelling in poker, in: Conf. on Uncertainty in Artificial Intelligence, 2005.
[67] P. Stone, M. Veloso, Towards collaborative and adversarial learning: A case study in robotic soccer, International Journal of Human Computer Studies 48 (1998).
[68] M. Tan, Multi-agent reinforcement learning: Independent vs. cooperative agents, in: Internat. Conf. on Machine Learning, 1993.
[69] G. Tesauro, Temporal difference learning and TD-gammon, Communications of the ACM 38 (3) (1995).
[70] M.L. Tsetlin, Automaton Theory and the Modelling of Biological Systems, Academic Press, 1973.
[71] D. Vickrey, D. Koller, Multi-agent algorithms for solving graphical games, in: National Conf. on Artificial Intelligence, 2002.
[72] X. Wang, T. Sandholm, Reinforcement learning to play an optimal Nash equilibrium in team Markov games, in: Conf. on Neural Information Processing Systems, 2002.
[73] X. Wang, T. Sandholm, Learning near-Pareto-optimal conventions in polynomial time, in: Conf. on Neural Information Processing Systems, 2003.
[74] M. Zinkevich, Online convex programming and generalized infinitesimal gradient ascent, in: Internat. Conf. on Machine Learning, 2003.