Learning and Coordination: an Overview

Myriam Abramson
Naval Research Laboratory
myriam.abramson@nrl.navy.mil

Ranjeev Mittu
Naval Research Laboratory
ranjeev.mittu@nrl.navy.mil

ABSTRACT

Adaptive learning techniques can automate the large-scale coordination of multi-agent systems and enhance their robustness in dynamic environments. This paper surveys several learning approaches that have been developed to address three different aspects of coordination, namely, learning coordination behavior, team learning, and the integrated learning of trust and reputation in order to facilitate coordination in open systems including collaborative systems where artificial agents and humans interact. Although convergence in multi-agent learning is still an open research question, several applications have emerged using some of the learning techniques presented.

KEYWORDS: Multi-agent Learning, Coordination, Survey

1. INTRODUCTION

An agent in a multi-agent system (MAS) has to decide (1) what task to do next, (2) whether to accept a task from another agent, (3) whether to ask another agent to achieve a task on its behalf, and (4) what information to share with other agents in order to achieve its goals [1]. In cooperative, distributed MAS, the agents share the same global goal and the problem there is to coordinate on local goals which can eventually lead to the satisfaction of global goals. Open systems, where agents come and go, are non-cooperative because agents do not necessarily share the same global goals but can occasionally cooperate on local goals. It is convenient to model large dynamic multi-modal problems through an open MAS paradigm to reduce the problem complexity and to use indirect methods, such as negotiations and incentives, to induce coordination so that most goals can be achieved.

Learning is a process, described by the tuple *{T, P, E}* where *T* is a task, *P* is the performance metric, and *E* the experience, whereby a machine is said to learn to improve its performance *P* at task *T* given experience *E* [2]. Defining the performance metric is essential in measuring the amount of learning accomplished. In MAS, the performance of a task depends on the coordination quality which can be evaluated in two ways: (1) as a managerial cost (for example, communication costs) or (2) as an emergent property. When considered as a managerial control cost, some meta-level reasoning has to take place to balance the benefit of an action and its cost [3]. For example, asking another agent to achieve a task might require some delays for negotiations. The absence of interference (including role conflicts and collisions) reflects coordination and the time taken to resolve or prevent interference represents the coordination cost. Instead of explicit coordination mechanisms to reduce the coordination cost, coordination can emerge implicitly from the coherence of local interactions of autonomous behavior. For example, simple adaptation mechanisms of reactive behavior can induce self-organization [4, 5]. Consequently, a coordination quality metric describing the efficiency of a multi-agent algorithm must combine three fundamental aspects of coordination: performance, resources, and failures (Fig. 1) [6].

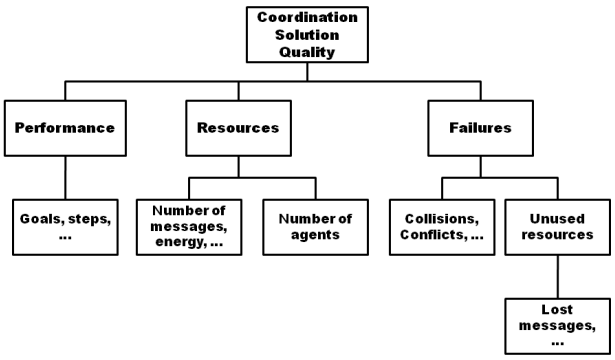

Figure 1. Taxonomy for coordination metrics

There are three basic approaches to machine learning: (1) supervised learning where the correct output is provided by a teacher during a training phase; (2) unsupervised learning where there is no knowledge of the correct output but where it is possible to distinguish equivalence classes in the input itself to guide the learning process; and (3) reward-based or reinforcement learning where only a feedback is provided by the environment on the utility of the performance. Using this feedback as a guide, a learner can in-

Table 1. Machine Learning Strategies

Machine Learning Strategy	Description
Case-based Learning	Instance-based method that adapts solutions to previous similar instances.
Decision-tree learning	Supervised method for instance classification that discovers the hierarchies of relevant features describing the search space.
Bayesian Learning	Incremental learning method that learns pattern approximations through the update of posterior distributions using Bayes' theorem.
Transfer Learning	Knowledge learned in one task can be reused to speed up learning in a different task.
Sequential Learning	Leverages from temporal contiguity in the data to formulate inferences.
Online Learning	Learns from one example at a time from multiple "experts." The goal is to learn to weight an expert's advice to minimize mistakes.
Active Learning	Learns by selecting the relevant training examples to maximize information gain.
Relational Learning	Learns inductively from relational features linking examples together.
Layered Learning	Learns at different degrees of abstraction.
Reinforcement Learning	Learns action utilities by trial and error from environmental feedback alone.
Evolutionary Learning	Population-based and policy-based reinforcement learning based on the biological metaphor of survival of the fittest.
Temporal Difference Learning	Reinforcement learning of a Markov decision process from two temporally successive estimates.
Collective learning	Distributed and emergent intelligence based on the metaphor of social insects.

crementally modify its performance. Table 1 summarizes some current machine learning strategies relevant to multi-agent learning (MAL). Learning coordination strategies is important in MAS to scale up to large state spaces using inductive techniques and for robust and flexible coordination behavior in dynamic systems. Learning by individual agents can, however, make a system more dynamic, but at the same time, may be problematic and prevent the system from reaching an equilibrium state. Convergence in MAL and its relevance in complex domains [7, 8] remains an open area of research that has stimulated countless workshops.

In MAL, it helps to distinguish between learning coordination behavior (Sec. 2), team learning of joint solutions (Sec. 3), and the integrated learning of indirect coordination mechanisms (Sec. 4). The merits of the different approaches depend on the problem context. In learning coordination behavior, the behavior of the other agents is implicit in maximizing individual performance. In team learning, anticipating the behavior of other agents is key to improving performance. In integrated learning, societal interaction issues between agents are considered. We conclude with some applications of machine learning techniques for coordination (Sec. 5) and a discussion of new perspectives (Sec. 6).

2. LEARNING COORDINATION BEHAVIOR

Coordination can be learned as a behavior itself. For example, learning conventions and learning when to communicate can improve the outcome due to less interference and conflicts. Learning coordination behavior can occur offline in a centralized manner or concurrently online by each agent.

2.1. Coordination Rules

One of the inherent limitations of multi-agent systems is the restricted knowledge and view of its agents of the global situation which can be compensated by information sharing. Monitoring traces from the context-specific situations of the agents can be collected offline, consolidated into a global picture, and divided into positive and negative problem-solving instances depending on the global outcome of the application of a task (or action). A supervised learning method, such as decision-tree learning, can then be used to determine the information necessary to execute a task. For example, in a disaster management scenario, the urgency of a refueling task for a firetruck depends on the knowledge of the intensity of nearby fires, the number of other rescuers in the area, the wind direction, the level of fuel, etc. Learning the relevant variables in the application of each task that drive the overall outcome can produce coordination rules that will determine the context-specific situation to establish in order to instantiate a task. The determination of this context will drive the need for information sharing in a multi-agent system.

Coordination rules can be learned using feedback from other agents and the environment in a collective way assuming homogeneity and cooperation. In a reinforcement learning approach, the action selection problem is modeled as a Markov decision process (MDP) represented by the tuple S,A,P,r where S represents the set of states, A represents the set of actions, P is the probability of going from state s_i to state s_j performing action a_i, and r is the reward obtained in state s_j. The goal is to learn a policy specifying the action to take in each state that will maximize the total expected reward. In temporal difference learning methods such as Q-learning, the expected reward at the next step, modulated by a decay factor, constitutes the credit assignment used to update the utility of performing action a_i in state s_i. In classifier systems where rules are evolved with a genetic algorithm [9], it is the final outcome that is propagated back (with a decay factor) as the credit assignment to each state-action rule in the sequence of actions leading to

it. Those two learning methods have been shown to be theoretically equivalent [10]. To avoid convergence problems due to the non-stationary environment of MAL, cooperative agents can bid on possible actions according to their action estimates [11] given their local context. The joint action set with the highest bid is executed and the sum of its bids propagated back to the actions of the previous joint action set to refine their estimates. Finding the maximum overall bid for an action set can be determined through distributed algorithms such as Adopt [12]. After a training phase, a policy can be learned by each agent that can be followed without additional communication overhead.

2.2. Conventions

In social systems, external mechanisms induce coordinated behaviors. For example, certain holidays coordinate the behavior of producers and consumers in the market without need for explicit communication. Similarly, agents can agree on certain conventions, based on common knowledge, to minimize conflicts and maximize gains without the added expense of communication. To learn conventions, agents must be homogeneous and cooperative, that is, the set of possible joint actions and transition functions must be commonly known and the agents must desire a global payoff (social maximum in game-theoretical terms) instead of an individual payoff. Conventions can be learned through incremental methods such as Bayesian learning where the probability of a best response leading to a coordinated joint action gets reinforced through the exploration of randomized strategies [13] in repeated play. In this approach, what is learned are the beliefs about the actions of the other agent(s) from which to construct a best response. In addition, as noted in [13], agents must recognize when to stop learning for conventions to emerge where actions are selected without deliberation and random fluctuations.

3. TEAM LEARNING

It is possible to achieve coordination by learning a team model instead of learning individual reactive behaviors in two ways. First, less interference will result by learning a team composition of behaviors and discovering the internal hierarchical structure of the task. In this approach coordination occurs by "growing" the behaviors together. Second, when a task decomposition is given, the problem becomes to learn to coordinate within that hierarchical abstract space.

3.1. Composition of Behaviors

Evolutionary algorithms offer offline and centralized solutions to the problem of learning team models. Complementary agent behaviors can be learned through co-evolution for cooperative tasks. In cooperative co-evolution [14], populations of specialized agents are evolved separately. The "best" of each population is then drafted to perform a cooperative task in an elitist fashion. The success of this task will readjust the fitness value of the agents in their respective population. This approach learns the needed component behaviors of a task by evolving single solutions modulated by coordinated solutions.

In swarm intelligence [15], experience shapes behavior by lowering response thresholds to stimuli from the environment. In turn, behavior also shapes experience by affecting the demands of the environment. A response threshold T is an adaptive reaction likelihood to perform an action (or task) as a function of the stimulus intensity s and tendency θ of an agent to perform the task. Several response threshold functions are possible. For example,

$$T_\theta(s) = \frac{s^n}{s^n + \theta^n}$$

where n is a constant parameter determining the steepness of the threshold. The stimulus intensity is then adjusted according to the relative proportion of active agents N_{act} in a population of N agents and learning rate α as follows:

$$s(t+1) = s(t) + \alpha \frac{N_{act}}{N}$$

In other words, the more an agent performs a task, perhaps due to local stimulus conditions, the more specialized it will become in this task in contrast to other tasks by adjusting its response threshold. This gives rise to a dynamic division of labor of heterogeneous "specialists" within a team of stochastically homogeneous agents.[1]

3.2. Role Conflicts

Case-based learning of situations where team members didn't perform according to expectation can improve the performance of single agent learners [16]. Here, an existing coordination strategy is adapted to prevent future conflicts by storing exceptions to the strategy. Heuristics or optimization techniques are not enough sometimes to disambiguate certain situations. For example, in the predator-prey domain where predators concurrently and independently decide which location to select (north, south, east, or west) in order to encircle the prey, a situation might require moving away from the prey to avoid collisions with other predators but a greedy strategy prevents predators from considering alternatives and yielding their place. Figure 6.2 illustrates the type of conflict that can arise in the predator-prey domain [17] using constraint optimization agents [6] and Manhattan distance. Here, the predator immediately west of the prey is unable to move east of the prey without colliding with other predators to the north and south. A better alternative of equal cost would be for one of the predator located north or south of the prey to yield its place and to move east. To consider this alternative as an exception to the standard constraint optimization strategy, progression to the goal by the team needs to be monitored. In case of failure, a new case will be learned and a solution proposed for negotiation among the members of the team.

[1] Homogeneous agents that have different parameter values initially determined stochastically according to a specific distribution.

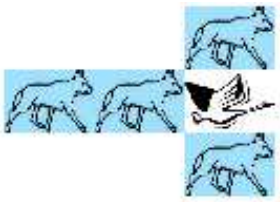

Figure 2. Conflict in the predator-prey domain: the predator located immediately west of the prey is unable to move east of the prey without colliding with other predators to the north and south.

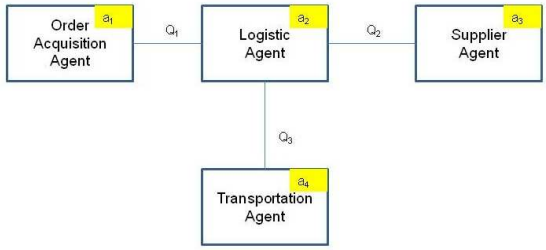

Figure 3. Coordination graph of supply-chain management agents. $Q = Q_1(a_1, a_2) + Q_2(a2, a3) + Q_3(a_2, a_4)$

3.3. Joint Actions

Learning in the joint action space can be computationally expensive in scaling up to a large number of agents. A joint action is a vector of single-agent actions. A coordination graph can decompose the joint action space into an agent dependency graph exploiting independence and isolating conflicts and constraints. Just like in the modeling of a Bayesian network, the dependencies have to be known in advance. Figure 3 illustrates dependencies in a partial supply-chain management workflow automated by intelligent agents where the optimality of each action depends on the joint optimality of "adjacent" actions. Factored MDPs [18] learn the global joint action value function Q as a linear approximation of several local, less complex, joint action value functions Q_i. Several centralized algorithms apply to solve Q which have been extended to decentralized algorithms suitable for autonomous agents alternating between single action learning and joint action learning[19].

3.4. Hierarchical Approaches

Teamwork occurs at different abstract levels requiring different learning approaches to seamlessly work together. In the RoboCup soccer domain [20], the individual, multi-agent and team abstract hierarchical levels motivate a layered learning approach [21]. What and how to learn at each layer needs to be specified a priori by the programming of the task. The behavior learned at one layer constrains the behavior learned at the upper layer. At the individual level, a skill not requiring the participation of other team members is learned. For example, in robotic soccer, intercepting the ball and dribbling are individual behaviors. At the multi-agent level, skills requiring the participation of other team members are learned (e.g., how to pass the ball to another player). Those multi-agent skills assume that individual complementary skills, such as ball interception, are learned. Learning at this layer feeds into the upper team layer learning where strategic behaviors are learned through policy evaluation methods such as reinforcement learning. For example, the decision of whether to pass the ball or to make a goal takes into consideration results from the mid-level layer on the evaluation of possible passes. A layered learning approach addresses the issue of non-stationarity in MAL by learning individual behaviors separately from multi-agent and team level behaviors. This approach requires a careful design of the *a priori* task decomposition.

Hierarchical reinforcement learning is a temporal difference learning method that offers a unified incremental learning approach for team learning at different levels of abstraction [22]. In the theory of semi-Markov decision processes (SMDPs), rewards for high-level abstract actions of strategic behavior, such as those found at the upper levels of a task decomposition, are a function of the mean reward accrued by their underlying primitive temporal actions weighted by the probability of reaching their goal in t time steps [23]. Given a task decomposition, the MAXQ hierarchical reinforcement learning algorithm [22] combines the utility value returned by a subtask with the predictive utility value of the task. This algorithm seamlessly and incrementally composes tasks at different levels of abstraction.

4. INTEGRATED LEARNING TECHNIQUES

Instead of directly learning coordination strategies, learning mechanisms for adjustable autonomy, incentives, negotiation, trust and reputation can help facilitate coordination and induce collaboration among heterogeneous agents.

4.1. Trust

Learning who to interact with is essential in open multi-agent systems. Trust can overcome the uncertainty associated with the outcome of an encounter and sway the decision of an agent to cooperate. Trust is a socio-cognitive belief that can be learned from experience but also acquired indirectly from the experience of other agents through recommendations. There are many aspects to trust and in the context of enhanced coordination, trust as a belief entails (1) that an agent will do as proclaimed (competence), (2) that an agent will do as predicted (commitment with no deception) and (3) that an agent will reciprocate. Reciprocation is a necessary component of trust to achieve a common social payoff through cooperation while non-reciprocating behavior provides short-term higher payoffs. Experiments

have shown the emergence of trust and reciprocating behaviors from interacting evolving agents in Prisoner's dilemma tournaments [24].

In addition to evolving trust implicitly through the evolution of coordination strategies such as tit-for-tat [24], trust can be learned socially as a trust function [25] for interacting with "trustworthy" agents. This trustworthiness attribute can be expressed through different values or labels l in applying the trust function. A trust function for a certain label l can be updated directly as a function of the interaction payoff P and learning rate $\alpha \in (0,1)$ as follows:

$$trust(l)[t] = (1-\alpha)trust(l)[t-1] + \alpha P\frac{N_l}{N}$$

where N_l is the number of agents with label l and N is the total number of agents. In conjunction, agents learn when to use different behaviors corresponding to a specific label to gain the trust of other agents and maximize the global social payoff. Labels do not have any intrinsic meaning but become signals for selective learning. For example, outer appearance and non-verbal messages often constitute labels around which trust is formed. Trust is learned and groups are formed through interaction and the exchange of labels. As noted in [25], because agents can learn to use labels deceptively, there is no convergent solution in this approach.

4.2. Adjustable Autonomy

Learning when to transfer control of certain decisions, or adjustable autonomy, is a key issue in teams of heterogeneous agents that may include humans. Transfer of control can leverage from the unique expertise of the different agents but can incur coordination costs in delaying decisions. Similarly, mixed-initiative interaction provides a flexible way to harness the cognitive capabilities of the human-in-the-loop. The key control of transfer decisions involve knowing *when* to task for help, *when* to ask for more information, and *when* to inform the user of a decision. Learning mechanisms include the reinforcement learning of Markov decision processes maximizing the utility of control transfer decisions based on the overall coordination performance as well as expectations and preferences [26]. For example, a scheduling agent could autonomously decide to cancel a meeting if it thinks that somebody will be late rather than incurring coordination costs in waiting for a confirmation while people sit idle. However, if the meeting is important, it is not expected or desired that it be canceled. Because reinforcement learning takes into account delayed rewards, it can avoid taking the wrong action minimizing short-term costs.

Coordination proxy agents are personal agents that take on the coordination role on behalf of a human user (Fig. 4) [27, 28]. While coordination proxies might prescribe optimal actions, switching tasks involves preferences such as loyalty, boredom and persistence thresholds. A learning

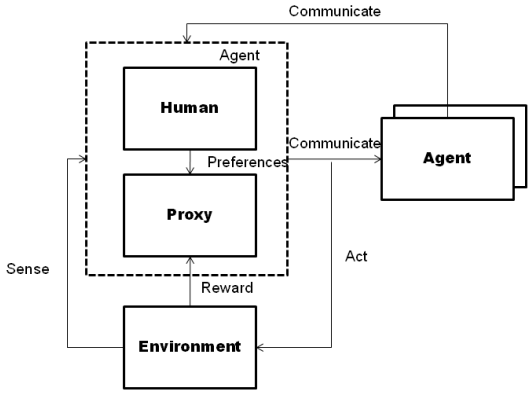

Figure 4. Coordination proxy architecture with interactions from the environments and with other agents

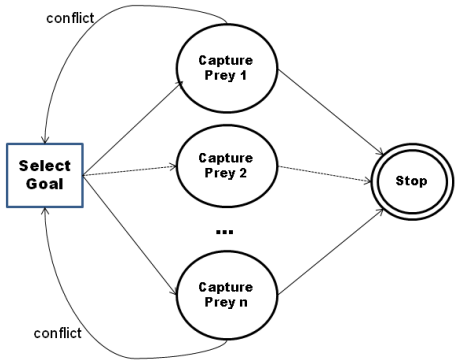

Figure 5. Prey/Predator HAM

approach for training coordination proxies in making decisions include the reinforcement learning of hierarchical abstract actions. Hierarchical abstract actions are high-level decisions, for example a planning decision, that are implemented by several primitive actions but are temporally abstract or "offline." Based on the theory of SMDPs, hierarchical abstract machines (HAMs) [29] addresses the issue of constraining a non-deterministic finite state machine of primitive actions by specifying valid transitions through high-level decisions or choice points. Machine states superimpose to environmental states to identify behavioral states and choice points. In the reinforcement learning of user preferences [28], coordination proxies learn whether to interrupt their users by reducing the ambiguity of a goal selection at a choice point. The intermediate reward r_c is obtained by similarity to a user decision at a choice point while the discounted reward $\gamma_c r$ is obtained from the temporal MDP upon reaching the goal selected. Figure 5 illustrates the application of HAMs for coordination proxies in the prey/predator domain [28].

5. APPLICATIONS OF MACHINE LEARNING TECHNIQUES FOR COORDINATION

Adaptive learning techniques have emerged in several application domains characterized by dynamic and uncertain environments.

5.1. Adaptive Routing

Routing algorithms for communication networks rely on information provided at each node by a routing table for transmitting packets to their destination. A routing table typically specifies the next node to select along with its cost in terms of transmission delay. Routing tables should be consistent among all nodes to avoid loops and should be dynamically updated as new nodes are brought up and old nodes fail or are congested. In distributed routing, local routing table information is shared among neighboring nodes until all nodes contain information on how to send packets to every other node.

In AntNet [30], a swarm intelligence approach based on the artificial ant colony paradigm [15] is used to update the routing tables in a distributed fashion. In this approach, mobile agents are dispatched from each network node to a random destination node in the network. Those agents traverse the network with a shortest path algorithm and added exploration so that alternative routes can be evaluated. In addition, they record the cost of each link and update the routing tables of the nodes traversed by returning to the source node along the same path. Those updates on the return trip are accumulated and aggregated much like the stigmergy process found in ant colonies whereby signals are embedded in the environment. With the help of those mobile agents, the network nodes collectively learn to maximize throughput. What is learned specifically are the probabilities to reach a destination node going through an intermediary node as a function of the associated trip time. In addition, probabilistic routing avoids congestion by using all probable good paths.

5.2. Robotics

As the deployment of unmanned systems is predicted to increase, the control of teams of robots has become a key issue. Self-organization through effective and robust coordination mechanisms to prevent collisions and interference between tasks reduces the amount of control necessary.

Adaptive coordination methods (without communication) in response to the environment were found to perform better than static methods [31]. Characteristics of the environment include group size, scenario type, uncertainty, etc. Machine learning techniques can be used for automatically tuning the parameters of heuristic coordination algorithms [32] or to automatically learn when to switch between heuristics. For example, one such coordination heuristic in robotics would be to move away from a teammate for a certain period of time. The size of the group was found to be a good indicator of the optimal amount of time required [31] for such a strategy. Evolutionary algorithms can search and optimize the parameter space of coordination algorithms by simulating their effect on relevant environmental conditions.

The ALLIANCE architecture [4] is a behavioral-based system for heterogeneous mobile robots applying multi-robot learning to control parameters. In this architecture, robots decide which task to perform based not only on the goals of the mission and environmental conditions, but also on their motivations for performing the task. Two such motivations are modeled: (1) impatience, whereby a robot takes up a task (possibly from another robot) and (2) acquiescence, whereby a robot gives up a task (possibly to another robot). The rate of impatience and acquiescence for a task characterizes different dynamic task allocation and reallocation strategies. Here, a scale factor by which to update those rates is learned by evaluating performance time first in a training phase and then in an adaptive phase. In the training phase, robots are experimenting with different behaviors and are maximally patient and minimally acquiescent. In an adaptive learning phase leading to lifelong learning, the rates of impatience and acquiescence guide the task allocation of robots and are continuously updated. The robots collectively learn to adjust to each other in the performance of tasks in a dynamic environment.

5.3. Intelligence, Surveillance, and Reconnaissance

The use of unmanned aerial vehicles (UAVs) to support intelligence, surveillance, and reconnaissance (ISR) has become a key enabler of network-centric operations to maintain situation awareness [33]. One key issue in teams of UAVs is the coordination of path planning to maximize surveillance.

Several metrics affect the optimization of coordinated paths such as track continuity, area coverage, idleness, cost, etc. Machine learning methods for the offline coordination of path planning include multi-objective, combinatorial optimization techniques such as evolutionary algorithms that can effectively search waypoint permutations and take into account constraints such as proximity to cell tower for communication purposes or safe distance from targets to escape detection [34]. Differential evolution [35] is an efficient technique for the evolution of waypoints as continuous values such as latitude-longitude or degree coordinates that replaces the traditional crossover technique of evolutionary algorithms. Differential evolution climbs the search space of possible solutions by exploiting differences in the population while exploring new solutions.

Environmental factors affect the traverse time of UAVs and therefore any offline route optimization has to incorporate a reactive component for replanning in mission-level tasks. ADAPTIV is a pheromone, swarm-based approach that provides a dynamic approach for the coordination of UAVs in

the battlespace environment [36]. Its approach is to embed a swarm of heterogeneous interacting agents on sensor platforms. Digital pheromones, deposited by *place agents* on unattended ground sensors (UGSs), act as potential fields for guiding UAVs toward their targets while avoiding collisions and threats. Place agents exchange information between themselves while UAVs, *walker agents*, dynamically interact with UGSs to plan a path in real time.

6. DISCUSSION

New perspectives in coordination include the use of economic concepts such as social choice to maximize agent preferences in a fair way. Combinatorial auctions provide a theoretical framework for resource allocation that applies to a variety of coordination problems such as bandwidth allocation, airport arrival and departure slots, and the load balancing of parallel processes. Beyond costly optimization techniques, learning how to bid or to select winning bids is a new, promising area of research for the coordination of MAS. As the capabilities and adaptability of MAS increase, the cognitive and ethical challenges for the human in the loop will become harder to ignore.

REFERENCES

[1] T. Sugawara and V. Lesser, "Learning coordination plans in distributed problem-solving environments," tech. rep., In Twelfth International Workshop on Distributed Artificial Intelligence, 1993.

[2] T. Mitchell, "The discipline of machine learning," Tech. Rep. CMU-ML-06-108, Carnegie Mellon University, 2006.

[3] A. Raja and V. Lesser, "Reasoning about Coordination Costs in Resource-Bounded Multi-Agent Systems," *Proceedings of AAAI 2004 Spring Symposium on Bridging the multiagent and multirobotic research gap*, pp. 35–40, March 2004.

[4] L. E. Parker, "L-ALLIANCE: Task-oriented multi-robot learning in behaviour-based systems," in *Advanced Robotics, Special Issue on Selected Papers from IROS'96*, pp. 305–322, 1997.

[5] T. H. Labella, M. Dorigo, and J. louis Deneubourg, "Efficiency and task allocation in prey retrieval," in *Proceedings of the First International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT2004), Lecture Notes in Computer Science*, pp. 32–47, Springer Verlag, 2004.

[6] M. Abramson, W. Chao, and R. Mittu, "Design and evaluation of distributed role allocation algorithms in open environments," in *International Conference on Artificial Intelligence*, (Las Vegas, NV), 2005.

[7] Y. Shoham, R. Powers, and T. Grenager, "If multi-agent learning is the answer, what is the question?," *Artificial Intelligence*, vol. 171, no. 7, pp. 365–377, 2007.

[8] P. Stone, "Multiagent learning is not the answer. it is the question," *Artificial Intelligence*, vol. 171, no. 7, pp. 402–405, 2007.

[9] I. Sen and M. Sekaran, "Multiagent coordination with learning classifier systems," in *Proceedings of the Workshop on Adaption and Learning in Multi-Agent Systems at AAMAS*, pp. 218–233, Springer Verlag, 1995.

[10] M. Dorigo and H. Bersini, "A comparison of q-learning and classifier systems," in *In Proceedings of From Animals to Animats, Third International Conference on Simulation of Adaptive Behavior*, pp. 248–255, MIT Press, 1994.

[11] G. Weiss, *Readings in Agents*, ch. Learning to coordinate actions in multi-agent systems, pp. 481–486. Morgan Kaufmann Publishers Inc., 1997.

[12] P. J. Modi, "An asynchronous complete method for distributed constraint satisfaction," in *Autonomous Agents and Multiagent Systems*, 2001.

[13] C. Boutilier, "Learning conventions in multiagent stochastic domains using likelihood estimates," in *In Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pp. 106–114, 1996.

[14] M. A. Potter and K. A. D. Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evolutionary Computation*, vol. 8, pp. 1–29, 2000.

[15] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, USA, 1999.

[16] T. Haynes, K. Lau, and I. Sen, "Learning cases to compliment rules for conflict resolution in multiagent systems," in *Working Notes for the AAAI Symposium on Adaptation, Co-evolution and Learning in Multiagent Systems*, pp. 51–56, AAAI Press. AAAI, 1996.

[17] M. Benda, V. Jagannathan, and R. Dodhiawalla, "On optimal cooperation of knowledge sources," Tech. Rep. BCS-G2010-28, Boeing AI Center, Boeing Computer Services, 1985.

[18] C. Guestrin, D. Koller, and R. Parr, "Multiagent planning with factored MDPs," in *Advances in Neural Information Processing Systems NIPS*, 2001.

[19] J. R. Kok and N. Vlassis, "Sparse cooperative q-learning," in *Proceedings of the 21st International Conference on Machine learning*, 2004.

[20] H. Kitano, M. Tambe, P. Stone, M. Veloso, S. Coradeschi, E. Osawa, H. Matsubara, I. Noda, and M. Asada, "The RoboCup synthetic agent challenge 97," in *Fifteenth International Join Conference on Artificial Intelligence*, (San Francisco, CA), pp. 24–29, Morgan Kaufmann, 1997.

[21] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, pp. 345–383, 2000.

[22] T. G. Dietterich, "The maxq method for hierarchical reinforcement learning," in *In Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 118–126, Morgan Kaufmann, 1998.

[23] M. L. Putterman, *Markov Decision Processes*. Wiley-Interscience, 2nd ed., 2005.

[24] R. Axelrod, *The Evolution of Cooperation*. Basic Books, 1984.

[25] A. Birk, "Boosting cooperation by evolving trust," *Applied Artificial Intelligence*, vol. 14, pp. 769–784, 2000.

[26] M. Tambe, P. Scerri, and D. V. Pynadath, "Adjustable autonomy for the real world," in *In Proceedings of AAAI Spring Symposium on Safe Learning Agents*, pp. 43–53, Press, 2002.

[27] P. Scerri, D. Pynadath, N. Schurr, A. Farinelli, S. Gandhe, and M. Tambe, "Team oriented programming and proxy agents: The next generation," in *Workshop on Programming MultiAgent Systems, AAMAS 2004*, 2004.

[28] M. Abramson, "Training coordination proxy agents using reinforcement learning," tech. rep., American Association of Artificial Intelligence, Arlington, VA, 2006. Fall Symposium.

[29] R. Parr and S. Russell, "Reinforcement learning with hierarchies of machines," in *Neural Information Processing Systems*, 1998.

[30] G. D. Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, no. 9, pp. 317–365, 1998.

[31] A. Rosenfeld, G. Kaminka, and S. Kraus, "Adaptive robot coordination using interference metrics," in *Proceedings of The Sixteenth European Conference on Artificial Intelligence*, 2004.

[32] J. Polvichai, P. Scerri, and M. Lewis, "An approach to online optimization of heuristic coordination algorithms," in *Proceedings of the 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pp. 623–630, May 2008.

[33] M. Abramson, R. Mittu, and J. Berger, "Coordination challenges and issues in stability, security, transition and reconstruction (sstr) and cooperative unmanned aerial vehicles," in *Int. Conf. on Integration of Knowledge Intensive Multi-Agent Systems (KIMAS 2007)*, pp. 428–433, 2007.

[34] I. K. Nikolos and A. N. Brintaki, "Coordinated UAV path planning using differential evolution," in *Proc. of the 13th Mediterranean Conference on Control and Automation*, 2005.

[35] R. Storn and K. Price, "Differential evolution- a simple and efficient adaptive scheme for global optimization over continuous spaces," Tech. Rep. TR-95-012, Berkeley, 1995.

[36] H. D. Parunak, M. Purcell, and R. O'Connell, "Digital pheromones for autonomous coordination of swarming uav's," in *First Technical Conference and Workshop on Unmanned Aerospace Vehicles, Systems, and Operations*, American Institute of Aeronautics and Astronautics, 2002.