1

Multi-Agent Coordination in Open Environments

Myriam Abramson and Ranjeev Mittu

Naval Research Laboratory
Washington, DC 20375

Summary. This paper proposes a new approach to multi-agent systems leveraging from recent advances in networking and reinforcement learning to scale up teamwork based on *joint intentions.* In this approach, teamwork is subsumed by the coordination of learning agents. The intuition behind this approach is that successful coordination at the global level generates opportunities for teamwork interactions at the local level and vice versa. This unique approach scales up model-based teamwork theory with an adaptive approach to coordination. Some preliminary results are reported using a novel coordination evaluation.

1.1 Introduction

Open environments such as Peer-to-Peer (P2P) and wireless or Mobile AdHoc Networks (MANET) provide new challenges to communication-based coordination algorithms such as *joint intentions* [13] as well as the opportunity to scale-up. Our framework is based on the proxy architecture of Machinetta [17] where *proxy* agents perform the domain-independent coordination task on behalf of *real*, domain-dependent agents. This framework is extended with a coordination mechanism of individual actions based on reinforcement learning. This adaptive proxy agent architecture is illustrated in Figure 1.1. In this approach, local teamwork outcomes provide the feedback for learning the coordination task on a larger scale. The teamwork theory of joint intentions and its associated problems in open environments are presented first and then our general approach, OpenMAS, is introduced with illustration from the prey/predator example [3]. An implementation addressing some of the issues is presented followed by conclusions for future work.

1.2 Joint intentions and Open Environments

Joint intentions [5, 13] form the cornerstone of teamwork theory of BDI (Belief, Desire, Intention) agents. It differentiates joint actions from individual actions by the

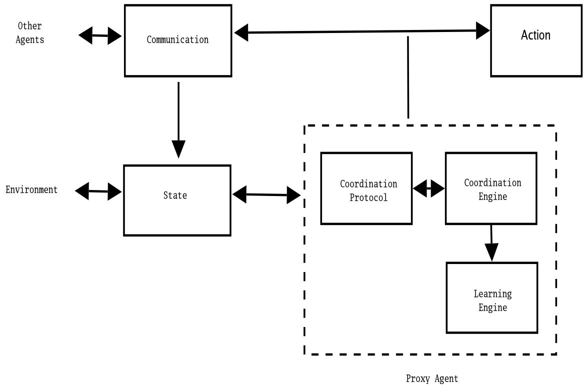

Fig. 1.1. Adaptive proxy agent architecture

presence of a common internal state (beliefs) and the joint commitment of achieving a goal. It is based on the communication of critical information among team members. However, the mutual responsiveness expected of team members at a local level is difficult to achieve on a larger scale. Open environments are characterized by their dynamic nature and the heterogeneity of the agents as well as asynchronous and unreliable communication on a large scale. The problems addressed can be categorized as follows: team formation, role allocation, synchronization of beliefs, communication trade-offs, and information sharing.

1. **Team Formation**. An open environment gives the opportunity to find teammates appropriate for a task instead of relying on a fixed group of agents. What is the best way to find teammates? When is the best time to find teammates? How to decide whether to join a team? In open environments, peers form "groups" by similarity of individual interests. Likewise, similarity of individual intentions is a necessary stepping stone for team formation in open environments. An intention is defined here [5] as the decision to do something in order to achieve a goal and can be construed as a partial plan.
2. **Role Allocation**. While direct point-to-point communication with any node can be expensive and uncertain, access to neighbors is readily available in open environments. P2P middleware, such as JXTA (Juxtapose) [1], provides the functionality needed to communicate reliably and cheaply with *neighbors*. In MANET, the possibility of disconnecting the network is another constraint in accepting a role requiring a change in location. Figure 1.2 describe the connection role that peers play in communication in MANET. In open environments, multiple teams are involved. How to adjust the connectivity role of the agents so that each team can accomplish its goals most effectively?
3. **Synchronization of Beliefs**. The theory of joint commitments is based on the ability to synchronize beliefs regarding "who is doing what". Teamwork breaks

down when roles do not match expected beliefs leading to *coordinated attack* dilemmas [14]. How to adjust gracefully to uncertainties in communication?

4. **Communication Selectivity**. The tradeoffs involve the robustness that redundancy of messages can provide in open environments versus the costs of communication to the network. When reliable communication cannot be assumed, selective communication of critical information might be detrimental to the coordination task.
5. **Information Sharing.** Sharing information is critical to the formation of a common internal state. The redundancy of messages from different sources provides corroborative evidence to support the information transmitted while conflicts undermine certainty. However, a problem in open systems is the unnecessary replication of messages from the same source through the network leading to false corroborative evidence.

Synchronization of beliefs, communication selectivity and information sharing are areas that are complicated by open environments, while team formation as well as role allocation are the problems we are interested in addressing given these complicating factors.

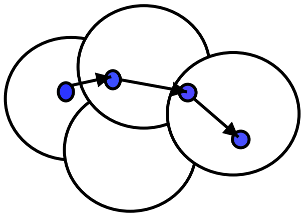

Fig. 1.2. Multi-hop routing in a MANET

1.3 OpenMAS Approach

Our proactive approach consists of leveraging from the belief framework of cognitive agents at the local level but endowing the agents with the adaptative capabilities of reinforcement learners as an additional coordination mechanism at the global level where communication is unsure and unreliable. The objective is to find a continuum between large-scale coordination and local teamwork. The overarching issues addressed in support of this objective are (1) how to integrate general models of cooperation with reinforcement learning in distributed, open environments (2) what are good evaluation measures for the propagation of beliefs to heterogeneous agents and (3) how to integrate multiple teams.

Methodology

Through the propagation of beliefs, the agents have some knowledge of the global situation, albeit imperfect and decaying with time. This capability relaxes the invalidation of the Markov property for multi-agent reinforcement learning systems. Instead of committing to a non-local role, the agents just commit to the next individual step. This is a least-commitment approach that addresses the problems outlined above of teamwork in open environments. Local environmental beliefs on the other hand trigger a role allocation mechanism among neighbors sharing the same beliefs. Role allocation of mutually exclusive tasks among agents can be modeled with distributed resource allocation algorithms based on constraint satisfaction [24]. The joint actions generated are preferred over the individual actions generated by the coordination learning mechanism. Similarities between joint actions and individual actions produce the terminal rewards needed for the learning algorithm. In this approach, there is a tight integration between the local level of teamwork and the global level of coordination. The overall approach is described in Algorithm 1. Figure 1.3 illustrates the approach in the prey/predator example.

Algorithm 1 Intention/Action loop

```
INPUT: intentions
OPENMAS-interpreter:
do
  <information, intention> ← receive-information()
  if similar-intentions(intention)
    accept-information()
    update-current-state()
  endif
  state-estimation()
  take-next-step()
  reinforce-learn()
  propagate <next-step, intentions>
forever
```

The information received includes information communicated from peers and/or perceived local information from the environment

The environment of agents acting under uncertainty can be conveniently modeled as a POMDP (Partially-observable Markov Decision Process). POMDP can be reformulated as continuous-space Markov decision processes (MDPs) representing belief states [10] and solved using an approximation technique. When propagating local environmental beliefs, the redundancy of messages reinforces current state beliefs through corroborative evidence while discrediting others. The most likely state of the global situation is then modeled as an MDP and the action to take determined by a stochastic policy approximated by a policy gradient method [19]. Through communication, the agents are able to construct a global, albeit imperfect, view of the world validating their assumption of the Markov property for independent autonomous decision-making based on trial and error. However, even assuming the same global

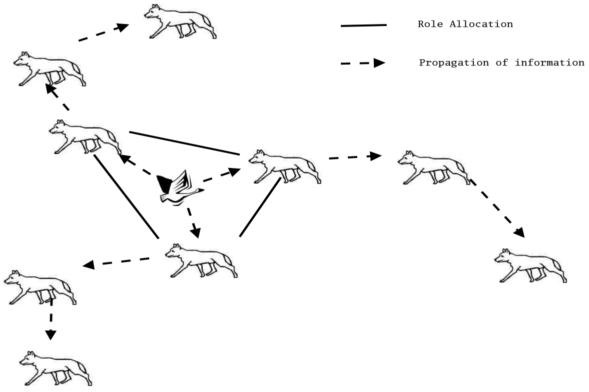

Fig. 1.3. Prey/predator example
The agents propagate changes of position and changes in the prey's status to their neighbors recursively according to a time-to-live (TTL) parameter. The TTL parameter ensures that a message does not bounce around needlessly when the destination cannot be found and can also be used to disseminate information within a certain range. Role allocation strategies resolve local conflicts.

knowledge of the world and optimization algorithm, coordination imposes the additional constraint that the agents choose the action leading to pareto optimality. For a deterministic policy, this constraint can be met through conventions or through the transmission of knowledge. Another way to meet this coordination constraint is to learn a stochastic policy that approximates a mixed strategy.

Role allocation endows the agents with a goal-driven behavior. In addition to accomplishing their roles and searching for possible role instantiations, agents in open environments have the additional implicit task of maintaining the connectivity of the network. It is necessary to balance those sometimes conflicting goals. The capability to assume multiple roles is a characteristic of intelligent and flexible behavior. Instead of modeling each goal separately in an MDP given the state of the environment, the goals themselves, as formulated by a role allocation strategy for each target, are part of the environment. This soft-subsumption architecture for multiple roles is illustrated in Figure 1.4.

1.4 Problem Modeling

The world is modeled as the problem space:

$$W = \{S, S', A, T, R\}$$

where

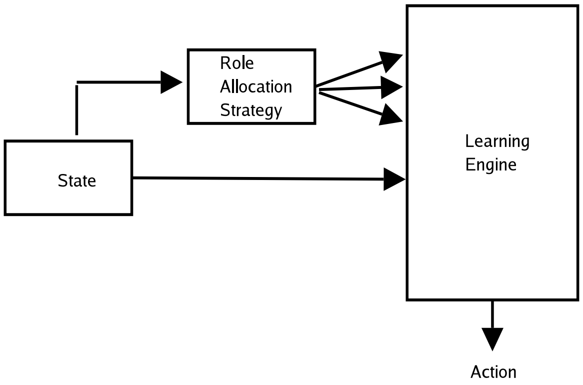

Fig. 1.4. Soft-subsumption architecture for multiple roles

- S is the believed perceived local state of the world.
- S' is the believed global state of the world through propagation of information.
- A is the set of actions.
- T is the set of transition probabilities

$$S \times A \times S \rightarrow [0, 1]$$

- R is the set of roles.

and

$$S \times R \rightarrow A_i$$

$$S' \times A_j \rightarrow \Re$$

where

- A_i is the action determined to achieve a role.
- A_j is the action determined by coordination in the believed state space S'.

A reward r is obtained if $A_i = A_j$.

The goal of each agent is to find a policy π maximizing the sum of expected rewards such that:

$$\forall s \in S',\, V^{\pi}(s) = r + \sum_{s' \in S'} T(s, \pi(s), s')\gamma V^{\pi}(s')$$

where s' is the next state following the action prescribed by $\pi(s)$, r is the reward in state s, and γ is the discount factor weighting the importance of future rewards.

1.5 An Example

An prototype evaluation of the OpenMAS general methodology has been conducted with some simplifying assumptions with the RePast simulation tool [6]. Further experiments are planned for a large-scale MANET simulation in ns-2 [2] using P2PS [22], a P2P agent discovery infrastructure designed to work in ns-2 simulations.

1.5.1 Prey/Predator Example

The prey/predator pursuit game is a canonical example in the teamwork literature [3,11] because one individual predator alone cannot accomplish the task of capturing a prey. Practical applications of the prey/predator pursuit game include, for example, unmanned ground/air vehicles target acquisition, distributed sensor networks for situation awareness, and rescue operations. The original problem can be extended to multiple teams by including more than one prey. Prey/predators can sense each other if they are in proximity but do not otherwise communicate. Predators communicate with other predators individually or can broadcast messages through their neighbors. Four predators are needed to capture a prey by filling out four different roles: surround the prey to the north, south, east and west. Those roles are independent of each other and can be started at any time obviating the need for scheduling. The only requirement is that they have to terminate at the same time either successfully when a capture occurs or unsuccessfully if no team can be formed. The predator agents are homogeneous and can assume any role but heterogeneity can be introduced by restricting the role(s) an agent can assume. The prey and predators move concurrently and possibly asynchronously at different time steps. In addition to the four orthogonal navigational steps, the agents can opt to stay in place. In case of collision, the agents are held back to their previous position. Several escape strategies are possible for the prey. A linear strategy, i.e. move away in the same random direction, has been shown to be an effective strategy while a greedy strategy, i.e. move furthest away from the closest predator, can lead to capture situations [9].

The preference or utility u_{ij} of predator agent i for a role j is inversely proportional to the Manhattan distance d required to achieve the role. Other factors such as fatigue, speed, resources, etc. can affect the preference for a role and are grouped under a capability assessment C [16].

$$u_{ij} = \frac{1}{d} \times C_{ij} \tag{1.1}$$

The predators move in the direction of their target when assigned a role or explore the space according to a pre-defined strategy. The decision space for the role allocation of P predators and p preys is $O(p^T)$ where T is the number of teams of size t that can be formed with P predators[1]. This problem belongs to the most difficult class of problems for constraint satisfaction in multi-agent systems [15] due to the dynamic nature of the environment and the mutually-exclusive property of role allocation.

1.5.2 Role Allocation Strategy

An optimization algorithm can be used in parallel fashion by each agent based on sensed and communicated information from the other agents in the group to autonomously determine which role to assume. It is assumed that the other agents reach the same conclusions because they use the same optimization algorithm [8]. The Hungarian algorithm [12] (see below) is used as the optimization method by each agent. Information necessary to determine the payoff of each role needs to be communicated. Therefore, it is the current local state within the perception range, or augmented with second hand information, that is communicated to the neighbors instead of the intended role in a trade-off between performance and privacy.

This algorithm, also known as the bipartite weighted matching algorithm, solves constraint optimization problems such as the job assignment problem in polynomial time. The implementation of this algorithm follows Munkres' assignment algorithm [4]. The algorithm is run over a utility matrix of $roles \times agents$. The maximization of utilities is transformed to a cost minimization problem:

$$cost = \arg\max \sum_{i,j} u_{ij}$$

$$Minimize \sum_{i,j} (cost - u_{ij})$$

The algorithm consists of transforming the matrix into equivalent matrices until the solution can be read off as independent elements of an auxiliary matrix. While additional rows and columns with maximum value can be added to square the matrix, the optimality is no longer guaranteed if the problem is over-constrained, i.e. there are more roles to be filled than agents. A simple example is illustrated in Table 1.1.

When multiple teams are involved, an agent chooses the role in the team that has the maximum sum of utilities rather than maximizing the sum of utilities across teams, thereby ensuring team formation.

1.5.3 Policy Search

In reinforcement learning, there are two ways to search the state space of a problem. We can search the policy space which is a mapping from current state to actions

[1] $\frac{P!}{(P-t)!}$

Table 1.1. A 4 × 4 assignment problem

The optimal assignment is (r_1, x_3), (r_2, x_2), (r_3, x_4), (r_4, x_1)

	x_1	x_2	x_3	x_4
r_1	0.79	0.28	1.00	0.89
r_2	0.29	0.51	0.83	0.38
r_3	0.33	0.03	0.47	0.91
r_4	0.92	0.14	0.82	0.80

or we can search the value space which is a mapping from possible states to their evaluation. Because there are only a limited number of actions that can be taken from a state, it is usually faster to search the policy space. Both methods however, should converge to the optimal greedy strategy whether by taking the best state-action value or the action that leads to the best valued state as the expected sum of rewards.

A function approximator generalizes to large state space. For gradient methods, it was shown that a small change in the parameter space can lead to large changes in the output space when searching for the value function while policy search where the output are action probabilities is assured locally optimal convergence [19]. Learning a stochastic policy has some advantages in dynamic and uncertain environments especially in pursuit games where the opponent might learn to escape a deterministic adversary.

1.5.4 Coordination Evaluation

Because coordination is an emergent property of interactive systems, it can only be measured indirectly through the performance of the agents in accomplishing a task where a task is decomposed in a number of goals to achieve. The more complex the task, the higher the number of goals needed to be achieved. While performance is ultimately defined in domain-dependent terms, there are some common characteristics. Performance can be measured either as the number of steps taken to reach the goal, i.e. the time complexity of the task, or as the amount of resources required. An alternative evaluation for coordination is the absence of failures or negative interactions such as collisions, lost messages, or fragmentation of the network when no messages are received. Figure 1.5 illustrates a taxonomy of coordination solution metrics. To show the scalability of a solution, the evaluation must vary linearly with the complexity of the task [7].

A combined coordination quality measure is defined as the harmonic mean of goals achieved g, net resources expanded r and collisions c as follows:

$$g = \frac{\#Preys\,Captured}{\#Preys} \tag{1.2}$$

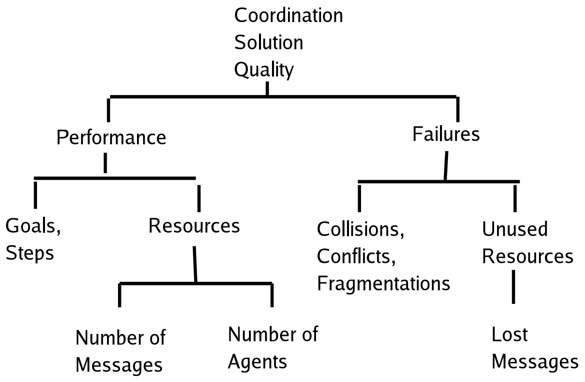

Fig. 1.5. Taxonomy of coordination solution quality for communicating agents

$$r = \frac{\#Predators}{log_2(\#Messages Received) + \#Predators} \tag{1.3}$$

$$c = \frac{\#Predators}{\#Collisions + \#Predators} \tag{1.4}$$

$$coordination = \frac{3grc}{gr + rc + cg} \tag{1.5}$$

Although the message size required by the different predator strategies was roughly equivalent, further work should measure the number of information bits per message [20].

1.5.5 Experimental Evaluation

In the prey/predator example, actions leading to collisions with other predators are negatively reinforced while actions leading to the capture of the prey are rewarded. Experiments were conducted on a 20x20 grid with 2 preys and a variable number of predators moving concurrently but synchronously at each time steps. The preys move to a random adjacent free cell 70% of the time except edge cells to avoid toroidal world ambiguities. The predators communicate their location and sensory information about the preys to their neighbors according to pre-defined communication and perception range. The probability of receiving a message vary according to a normal distribution based on (Euclidean) distance and the communication range of each agent. The current state is represented by the one-dimensional locations of the preys, the current location of the agent, and the location of the closest other three predators known. A feed-forward neural net was implemented with 54 binary input nodes, 7 hidden nodes and 5 output nodes to translate to the four possible orthogonal directions to move and an option to stay in place. Each output node o_i represents the probability that the direction will lead to success. The sigmoid transfer function was used for all internal and output nodes.

The agents learn to coordinate through trial and error in simulation using temporal difference learning between the value of the current direction in the output vector and the value of the direction in the last output vector. They train following the optimized role allocation strategy (see 1.5.2) when available as their behavior policy. The direction to take is then *conventionally* derived according to the differences between the destination of the role assignment and the current location of the agent along the x-axis first and then the y-axis. When no role allocation is found, a softmax policy is followed where the direction i is selected stochastically according to the probability $P(i) = \frac{e^{o_i}}{\sum_i e^{o_i}}$. A reward of 1.0 is received when a goal is reached or when a role allocation was found and a penalty of 1E-6 is received when colliding.

In the performance phase, the neural net from the most successful agent is selected. Table 1.2 summarizes the different parameters used. Figure 1.6 shows coordination quality results averaged over 1000 runs comparing different policies followed when restricting the optimized role allocation strategy above a certain utility threshold comprising about 5% of the interactions. There is a significant difference between the results obtained following the greedy policy learned through reinforcement learning and a random policy (t-test p-values were 2E-5, 0.0001, 0.004 for 7, 8 and 9 predators respectively). The memory-based approach consists of moving to an adjacent cell that was not visited in the last 7 steps. Interestingly, although memory-based exploration performs better than random walk for a single reinforcement learner agent [21], they rate worse for multi-agent coordination. Those experiments have shown that learning from past experiences can produce a viable behavioral policy on a larger scale that is conducive to teamwork on a local scale and that can produce domain-dependent coordination rules. Further application of state estimation techniques should enhance this approach.

Table 1.2. Parameters

Input nodes	54
Hidden nodes	7
Output nodes	5
Learning rate α*	0.3
Penalty	1E-6
Reward	1.0
Role utility threshold	3.0
Communication range	7
Perception range	2
Cycles	1000
Tmax	3000

*decreasing with time t at the rate $\frac{\alpha}{1+\frac{t}{Tmax}}$

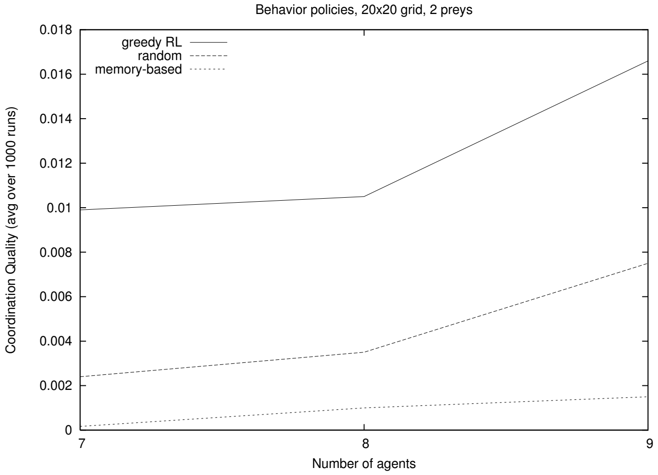

Fig. 1.6. Behavior policies

1.6 Related Work

The dissemination of information enables agents to obtain some global, though imperfect, knowledge of the world. This capability is taken into account in scaling up teamwork approaches based on communication and our approach also takes this capability into account to enhance multi-agent learning. Our approach is different from the large-scale coordination of Machinetta proxies [18] because (1) individual actions lead to joint actions through on-line adaptation and (2) the uncertainty and ambiguity of information is taken into account through state estimation. Our least-commitment approach is however similar to a token-based approach to teamwork [16].

The importance of communication in solving decentralized Markov decision processes was noted in [23] where the goal was to develop a communication policy in addition to the navigation policy. For agents in open environments, those policies overlaps since the location of the agent determines its communication range.

1.7 Conclusions and Future Work

Open environments such as P2P and MANET forces a reexamination of teamwork in large scale systems relying more on adaptive coordination than explicit cooperation requiring synchronization points. The capability to acquire global, albeit imperfect, knowledge through the propagation of information makes it possible to use independent reinforcement learners for coordination tasks in multi-agent systems. Similarity of intentions can help relieve the burden placed on the network by selectively propagating information while state estimation based on evidence reasoning calibrates incoming information. A local teamwork model drives the rewards of the overall coordination task. This proactive approach scales well to any dimensions and its precision can be modulated by the TTL parameter. Future experiments are planned for large MANET network simulations and P2P agent discovery of heterogeneous agents.

Acknowledgement

The authors want to acknowledge useful discussions with Joe Macker, William Chao, and Joe Collins.

References

1. Project jxta. http://www.jxta.org.
2. The network simulator. Retrieved from http://www.isi.edu/nsnam/ns/, 2003.
3. M. Benda, V. Jagannathan, and R. Dodhiawalla. On optimal cooperation of knowledge sources. Technical Report BCS-G2010-28, Boeing AI Center, Boeing Computer Services, 1985.
4. F. Burgeios and J. C. Lassalle. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Communication of the ACM*, 14:802–806, 1971.
5. Philip R. Cohen and Hector J. Levesque. Teamwork. *Nous*, 25(4):487–512, 1991.
6. Nick Collier. Repast: the recursive porous agent simulation toolkit. Retrieved from http://repast.sourceforge.net, 2001.
7. E. H. Durfee. Scaling up agent coordination strategies. *IEEE Computer*, 2001.
8. Brian P. Gerkey and Maja J. Mataric. *RobotCup 2003*, volume 3020, chapter On Role Allocation in RobotCup. Springer-Verlag Heidelberg, 2004.
9. Thomas Haynes and Sandip Sen. Evolving behavioral strategies in predator and prey. In *IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems*, 1995.
10. L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, (101):99–134, 1998.
11. Richard E. Korf. A simple solution to pursuit games. In *Working Papers of the 11th International Workshop on Distributed Artificial Intelligence*, pages 183–194, 1992.
12. H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(83), 1955.
13. H. J. Levesque, P. R. Cohen, and J. Nunes. On acting together. In *Proceedings of the National Conference on Artificial Intelligence*, 1990.
14. Nancy Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.
15. P. J. Modi, H. Jung, M. Tambe, W. Shen, and S. Kulkarni. Dynamic distributed resource allocation: A distributed constraint satisfaction approach. In *Proceedings of the 7th International Conference on Principles and Practice of Constraint Programming*, 2001.
16. P. Scerri, A. Farinelli, S. Okamoto, and M. Tambe. Token approach for role allocation in extreme teams: Analysis and experimental evaluation. In *Proceedings of 2nd IEEE International Workshop on Theory and Practice of Open Computational Systems*, 2004.

17. P. Scerri, D. Pynadath, N. Schurr, A. Farinelli, S. Gandhe, and M. Tambe. Team oriented programming and proxy agents: The next generation. Workshop on Programming MultiAgent Systems, AAMAS 2003.
18. Paul Scerri, Elizabeth Liao, Justin Lai, and Katia Sycara. *Cooperative Control*, chapter Coordinating Very Large Groups of Wide Area Search Munitions. Kluwer Publishing, 2004.
19. R. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, pages 1057–1063. MIT Press, 2000.
20. Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative learning. In Michael N. Huhns and Munindar P. Singh, editors, *Readings in Agents*, pages 487–494. Morgan Kaufmann, San Francisco, CA, USA, 1997.
21. S. B. Thrun. Efficient exploration in reinforcement learning. Technical Report CMU-CS-92-102, Pittsburgh, Pennsylvania, 1992.
22. Ian Wang and Ian Taylor. P2ps, peer-to-peer simplified. Retrieved from http://srss.pf.itd.nrl.navy.mil/, 2003.
23. Ping Xuan and Victor Lesser. Multi-agent policies: From centralized ones to decentralized ones. In *Autonomous Agents and Multi-Agent Systems*, 2002.
24. M. Yokoo. *Distributed Constraint Satisfaction*. Springer-Verlag, 1998.