

Multiagent Cooperation through Egocentric Modeling

Vincent Pei-wen Seah and Jeff S. Shamma
Department of Mechanical and Aerospace Engineering
University of California Los Angeles
{vpwseah, shamma}@ucla.edu

Abstract—We consider a scenario in which interacting agents cooperate through an iterative process of 1) forming empirical models of the behavior of other agents and 2) selfishly optimizing a local strategy based on these models. In each iteration, an agent revises its models of other agents. Selfish optimization according to these revised models alters the behavior of each agent. This, in turn, leads to a new round of revised models of other agents. The implication of convergence is a consistency condition. Namely, each agent’s behavior is consistent with how the agent is modeled by others. Furthermore, each agent’s local strategy is optimal with respect to how it models other agents. We consider a particular instance of this framework that is motivated by the “Roboflag drill” coordination scenario. This paper derives conditions for convergence, provides illustrative simulations, and establishes a connection to related work in evolutionary games.

I. INTRODUCTION

Many engineering systems can be modeled as a large-scale collection of interacting subsystems—each having access to local information, each making local decisions, and each seeking to optimize local objectives that may well be in conflict with other subsystems. Such models fall under the scope of research on “multiagent systems”. There is a vast literature on this subject, which includes collections such as [1] as well as related economic game theory monographs on learning in games [2] [3].

Designing autonomous or semi-autonomous systems is a driving motivation for the multiagent framework. Accordingly, much of the research is focused on *learning* or *adaptation*, e.g., [4]. Multiagent learning presents significant challenges [5] that distinguish it from conventional single agent [6] [7] learning. In single agent learning, there is a stationary environment, and an agent learns to operate within this environment through increased experience. Now consider a multiagent setting, as in Figure 1, and in particular, consider the “environment” from the perspective of *agent N*. It is composed of *other* agents, and since other agents are undergoing a learning process, the environment cannot be modeled as stationary. In other words, by the time *agent N* may have learned the environment from its own perspective, the environment has changed.

In this paper, we also consider a problem of multiple interacting agents. Our specific interest is the following “decomposition” approach. First, each agent is tasked with an individual performance objective that depends on its own strategy and the strategy of other agents. Through simulation and/or experience, agents revise their strategies in an iterative procedure as follows: 1) agents make a (very) simplified model of the behavior of other agents, 2) agents design “optimal” strategies that presume such behavior from other agents, 3)

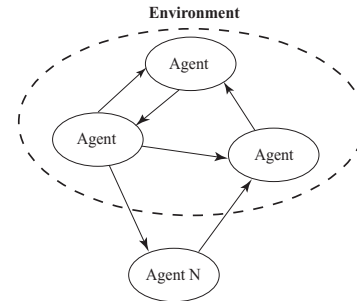


Fig. 1. Multiagent System Schematic Illustration

upon deployment of these strategies, agents observe other agents, revise their models, redesign their strategies, and so on.

The implication of convergence is a consistency condition. Namely, each agent’s behavior is consistent with how the agent is modeled by others. Furthermore, each agent’s local strategy is optimal with respect to how it models other agents.

Such an approach was taken in [8] [9] in the context of manufacturing systems. In that work, simulation studies illustrated the potential of the decomposition approach. Furthermore, simulation studies showed that the iterative procedure exhibits convergence, i.e., agents eventually settle on a set of models and optimized strategies that are consistent.

The particular context considered here is a variation of the “Roboflag drill”, introduced in [10]. In this problem, a team of defenders is to engage an oncoming team of attackers who appear randomly. Figure 2 illustrates the setup. The potential of the decomposition approach is to enable the decentralized design of a defense strategy that exhibits effective performance and yet avoids a centralized *a priori* design.

In this paper, we will present a simulation study of our decomposition approach on the Roboflag drill, compare the performance to a fully centralized and fully decentralized design, and use methods from stochastic approximation [11] [12] to analyze convergence.

The remainder of this paper is organized as follows. Section 2 contains the two attacker/two defender model illustrated in Figure 2 and presents both “fully centralized” and “fully decentralized” solution to the optimal defense. Section 3 presents the decomposition approach of evolutionary coordination. Section 4 presents an analytical discussion of convergence. Finally, Section 5 contains concluding remarks.

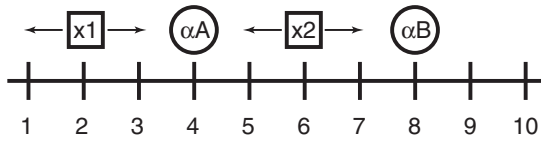


Fig. 2. Roboflag Drill.

II. CENTRALIZED AND DECENTRALIZED OPTIMIZATION

In this section, we specify the model of the Roboflag drill depicted in Figure 2 present two scenarios — a fully centralized optimization and a fully decentralized optimization — as benchmarks for the forthcoming “evolutionary” scenario.

A. Markov Model

We model a two attacker and two defender scenario as coupled Markov chains.

The state space, S , is all possible positions of the defenders and attackers. Specifically, $s \in S$ is of the form $s = (x_1, x_2, \alpha_A, \alpha_B)$, denoting the positions of Defender 1, Defender 2, Attacker A, and Attacker B, respectively. Each of these takes an integer value in $\{1, 2, \dots, 10\}$.

To avoid a technical ambiguity, we will impose the convention that if the defenders occupy the same position, then $x_1 < x_2$. In this regard, it may be convenient to view

$$\begin{aligned} x_1 &\in \{1^-, 2^-, \dots, 10^-\}, \\ x_2 &\in \{1^+, 2^+, \dots, 10^+\}, \end{aligned}$$

where

$$j - \varepsilon = j^- < j < j^+ < j + \varepsilon, \quad j = 1, 2, \dots, 10,$$

for some small $\varepsilon > 0$.

The combined control actions are $u = (u_1, u_2)$, where u_i is the *intended* movement to the i^{th} defender. Each u_i must satisfy $u_i(k) \in U(x_i)$, where

$$U(x_i) = \begin{cases} \{0, 1\}, & x_i = 1; \\ \{-1, 0\}, & x_i = 10; \\ \{-1, 0, 1\}, & \text{otherwise.} \end{cases}$$

In other words, u_i denotes move right, move left, or do not move, with movement restricted at the endpoints.

As stated earlier, the u_i represent the *intended* movement. In this model, the defenders are dynamically coupled in that the intended movement need not be the actual movement. Rather, there is a random “cohesiveness” that depends on the relative positions of the two defenders.

We can write the discrete-time dynamics of the defenders as

$$x_1(k+1) = x_1(k) + w_1(k) \quad (1a)$$

$$x_2(k+1) = x_2(k) + w_2(k). \quad (1b)$$

The *actual* moves are determined by the random variables, w_1 and w_2 , as opposed to the *intended* moves, u_1 and u_2 . The probabilities of w_1 and w_2 depend on 1) the relative positions, x_1 and x_2 , and 2) the intended movement, u_1 and u_2 , of the

two defenders, as follows. If a defender intends, through u_i , to move *towards* the other defender, then the move is realized, through w , with probability p . If a defender intends to move *away* from the other defender, then the move is realized with probability q . Finally, if a defender intends to not move, then the move is realized with probability one.

The values of p and q are chosen to be $1 > p > 1/2$ and $0 < q < 1/2$, although this is not essential for the forthcoming analysis. A interpretation is that an intended move to move *away* from the other defender has a higher probability of being “vetoed” than an intended move *towards* the other defender. In this way, the dynamics of the two defenders are coupled by their relative positions.

The probabilities of w_i are stated more precisely as follows:

- *Case 1:* $x_1 < x_2$ and u_1 or $u_2 \neq 0$:

$$\Pr[w_1 = 1 | u_1 = 1] = p \quad (2a)$$

$$\Pr[w_1 = 0 | u_1 = 1] = 1 - p \quad (2b)$$

$$\Pr[w_1 = -1 | u_1 = -1] = q \quad (2c)$$

$$\Pr[w_1 = 0 | u_1 = -1] = 1 - q \quad (2d)$$

$$\Pr[w_2 = 1 | u_2 = 1] = q \quad (2e)$$

$$\Pr[w_2 = 0 | u_2 = 1] = 1 - q \quad (2f)$$

$$\Pr[w_2 = -1 | u_2 = -1] = p \quad (2g)$$

$$\Pr[w_2 = 0 | u_2 = -1] = 1 - p \quad (2h)$$

- *Case 2:* $x_2 < x_1$ and u_1 or $u_2 \neq 0$:

$$\Pr[w_1 = 1 | u_1 = 1] = q \quad (3a)$$

$$\Pr[w_1 = 0 | u_1 = 1] = 1 - q \quad (3b)$$

$$\Pr[w_1 = -1 | u_1 = -1] = p \quad (3c)$$

$$\Pr[w_1 = 0 | u_1 = -1] = 1 - p \quad (3d)$$

$$\Pr[w_2 = 1 | u_2 = 1] = p \quad (3e)$$

$$\Pr[w_2 = 0 | u_2 = 1] = 1 - p \quad (3f)$$

$$\Pr[w_2 = -1 | u_2 = -1] = q \quad (3g)$$

$$\Pr[w_2 = 0 | u_2 = -1] = 1 - q \quad (3h)$$

- *Case 3:* u_1 or $u_2 = 0$:

$$\Pr[w_1 = 0 | u_1 = 0] = 1 \quad (4a)$$

$$\Pr[w_2 = 0 | u_2 = 0] = 1 \quad (4b)$$

Because of the assumed convention that $x_1 \in j^-$ and $x_2 \in j^+$, there is no ambiguity regarding moving “towards” or “away”.

The dynamics of the attackers are

$$\alpha_j(k+1) = \begin{cases} \alpha_j(k), & \text{if } x_i(k) \neq \alpha_j(k), i = 1, 2; \\ w_3(k), & \text{otherwise,} \end{cases} \quad (5)$$

for $j \in \{A, B\}$. The attacker stays at a location until it is intercepted by either defender. When intercepted, the attacker’s reappearance is a random variable, w_3 , based on an uniform probability distribution over $\{1, 2, \dots, 10\}$.

B. Fully Centralized Optimization

The objective is to intercept as many attackers as possible. This is reflected in a centralized discounted infinite horizon cost

$$\min E \left\{ \sum_{k=1}^{\infty} \rho^k g_c(s(k), u(k)) \right\}$$

with $\rho \in (0, 1)$. The minimization is over all stationary policies, μ , which are a function of the state, i.e.,

$$u(k) = \mu(s(k)).$$

The stage cost is

$$g_c(s, u) = \min \left\{ E \left\{ \left\| x(k+1) - \begin{pmatrix} \alpha_A \\ \alpha_B \end{pmatrix} \right\| \mid x(k), u(k) \right\}, E \left\{ \left\| x(k+1) - \begin{pmatrix} \alpha_B \\ \alpha_A \end{pmatrix} \right\| \mid x(k), u(k) \right\} \right\},$$

which reflects the expected normed distance of both defenders to both attackers. The minimization reflects an indifference to which defender engages which attacker.

$E \{x(k+1)|x(k), u(k)\}$ refers to the expectation of the location of the defenders at time $(k+1)$ given if $u(k)$ is chosen today.

This is a standard finite state Markov decision problem whose solution can be computed using dynamic programming [13]. Let $J_c^*(s)$ denote the optimal cost of state s , and let J_c^* denote the vector of optimal costs, i.e.,

$$J_c^* = \begin{pmatrix} J_c^*(s^1) \\ J_c^*(s^2) \\ \vdots \end{pmatrix}$$

The Bellman equation can be written as

$$J_c^*(s) = \min_{u \in U} g_c(s, u) + \rho T_c(s, u)^T J_c^*$$

where $T_c(s, u)$ is a vector of state and control dependent state-transition probabilities. More precisely,

$$T_c(s, u) = \begin{pmatrix} p_{ss^1}(u) \\ p_{ss^2}(u) \\ \vdots \end{pmatrix}, \quad (6)$$

where $p_{ss^k}(u)$ denotes the u -dependent transition probability from state s to state s^k .

C. Fully Decentralized Optimization

In this setup, each defender independently solves an optimization. Furthermore, a defender is unaware of the location of the co-defender, even though the transition probabilities *still depend* on the relative location of the two defenders. In order to formulate an individual optimization problem, each defender makes a *simplified subjective model* of the other defender's behavior.

In the decentralized optimization, each defender maintains a separate state-space, S_1 , and S_2 , defined by

$$s_1 = (x_1, \alpha_A, \alpha_B) \in S_1, \quad s_2 = (x_2, \alpha_A, \alpha_B) \in S_2,$$

respectively for Defender 1 and Defender 2. As before, $u_i(k) \in U(x_i(k))$.

The simplified model each defender makes of the other defender is that the other defender is currently to its left or right with some probability and independently of other events. It is understood that this does not reflect the actual behavior. Nonetheless, this representation enables each agent to design an optimal controller independently, given its own subjective model.

Let the parameter θ represent the probability that $x_2(k) > x_1(k)$. The discrete time dynamics still take the form

$$x_1(k+1) = x_1(k) + w_1(k) \quad (7a)$$

$$x_2(k+1) = x_2(k) + w_2(k), \quad (7b)$$

but now the probabilities for w_i are

$$Pr [w_1 = 1 | u_1 = 1] = \theta p + (1 - \theta)q, \quad (8a)$$

$$Pr [w_1 = 0 | u_1 = 1] = 1 - Pr [w_1 = 1 | u_1 = 1], \quad (8b)$$

$$Pr [w_1 = 0 | u_1 = 0] = 1, \quad (8c)$$

$$Pr [w_1 = -1 | u_1 = -1] = \theta q + (1 - \theta)p, \quad (8d)$$

$$Pr [w_1 = 0 | u_1 = -1] = 1 - Pr [w_1 = -1 | u_1 = -1]. \quad (8e)$$

Likewise,

$$Pr [w_2 = 1 | u_2 = 1] = \theta q + (1 - \theta)p, \quad (9a)$$

$$Pr [w_2 = 0 | u_2 = 1] = 1 - Pr [w_2 = 1 | u_2 = 1], \quad (9b)$$

$$Pr [w_2 = 0 | u_2 = 0] = 1, \quad (9c)$$

$$Pr [w_2 = -1 | u_2 = -1] = \theta p + (1 - \theta)q, \quad (9d)$$

$$Pr [w_2 = 0 | u_2 = -1] = 1 - Pr [w_2 = -1 | u_2 = -1]. \quad (9e)$$

It is important to stress that these are *not* the probabilities for the w_i in the actual evolution of the system. Rather, these are the *presumed* (and incorrect) probabilities, as a function of θ , that each agent uses for the fully decentralized optimization.

The attackers' dynamics, for the sake of decentralized optimization, are the same as the previous setup. However, neither defender models the other defender intercepting an attacker, i.e.,

$$\alpha_j(k+1) = \begin{cases} \alpha_j(k), & \text{if } x_i(k) \neq \alpha_j(k); \\ w_3(k), & \text{otherwise,} \end{cases} \quad (10)$$

for *either* $i = 1$ or $i = 2$.

The stage cost at time k is now defined for each defender as

$$g_{dc}(s_i, u_i, \theta) = \min_{\alpha_j} E \left\{ |x_i(k+1) - \alpha_j| \mid u_i(k) \right\}. \quad (11)$$

The total cost for each defender is the discounted infinite horizon cost, i.e.,

$$\min E \left\{ \sum_{k=1}^{\infty} \rho^k g_{\text{dc}}(s_i(k), u_i(k), \theta) \right\}$$

As before, this optimization can be solved using standard dynamic programming, with the Bellman equation being

$$J_{\text{dc},\theta}^*(s_i) = \min_{u_i \in U(x_i)} g_{\text{dc}}(s_i, u_i, \theta) + \rho T_{\text{dc}}(s_i, u_i, \theta)^T J_{\text{dc},\theta}^* \quad (12)$$

Note that the resulting optimal cost-to-go depends on θ . As before, T_{dc} denotes a vector of transition probabilities,

$$T_{\text{dc}}(s_i, u_i, \theta) = \begin{pmatrix} p_{s_i s^1}(u, \theta) \\ p_{s_i s^2}(u, \theta) \\ \vdots \end{pmatrix} \quad (13)$$

where $p_{s_i s^k}(u, \theta)$ denotes the u and θ dependent transition probability from $s_i \in S_i$ to $s^k \in S_i$. Again, it is important to stress that the transition probabilities $T_{\text{dc}}(s_i, u_i, \theta)$ are according to a defender's θ -dependent *internal model* of the overall system (7)–(9) and attacker dynamics (10).

III. EVOLUTIONARY COOPERATION

We now look into how defenders with decentralized optimization can achieve cooperation through iterative learning.

The learning will evolve over intervals, $\{I_1, I_2, \dots\}$, where each interval consists of multiple stages. Over interval I_n , each defender employs an optimal decentralized policy (derived in Section II-C) based on a modeling parameter value $\theta(n)$. Then, a new $\theta(n+1)$ is computed based on 1) $\theta(n)$ and 2) observed date over interval I_n . In particular, define

$$\theta_{\text{obs}}(n) = \frac{\# \text{ times } x_1(k) < x_2(k) \text{ over interval } I_n}{\text{total number of stages in interval } I_n},$$

i.e., the percentage of times $x_1(k) < x_2(k)$ over interval I_n . The θ -update equation is now defined as

$$\theta(n+1) = \theta(n) + \frac{1}{n+1}(\theta(n) - \theta_{\text{obs}}(n)).$$

Table I tabulates the resulting performance of fully centralized, fully decentralized, and evolutionary cooperation. For this table, a “hit” denotes an attacker being intercepted within 10 steps. After 10 steps, the attacker moves to a randomly selected location and a “miss” is registered. As anticipated, the fully centralized setup exhibits the best performance, but the evolutionary cooperation setup outperforms the fully decentralized setup. The θ values converge to $\theta = 0.9113$ after approximately 50 iterations.

Table I also shows an additional evolutionary setup, which uses a modification of the decentralized stage cost defined in (11). The idea behind the modified stage cost is as follows. The original stage cost (11) reflects that each defender has a greedy policy of intercepting whichever attacker is nearer. The modified stage cost penalizes the defender should it attempt to intercept the attackers in the area where, according to

TABLE I
PERFORMANCE COMPARISONS.

Setup	Hits	Misses	% Hits	θ^*
Fully Centralized	976	24	97.6%	-
Fully Decentralized	700	90	88.6%	-
Evolutionary Cooperation	862	68	92.7%	0.9113
Evolutionary Cooperation (modified cost)	890	36	96.1%	0.9397

the modeled θ value, a co-defender has a high probability presence. Intuitively, this means that the defender respects the presence of the co-defender and its capability to intercept any attackers in that area *without* modeling the detailed behavior of the co-defender, and thereby maintaining a decentralized optimization.

The modified stage cost is defined as follows. First, define

$$\gamma_{ij} = E \left\{ |x_i(k+1) - \alpha_j| \mid u_i(k) \right\},$$

where the expectation for $x_i(k+1)$ is according to the (decentralized) probabilities of w_i in (8)–(9). The stage cost for Defender 1 is

$$g_{\text{mod}}(s_1, u_1, \theta) = \begin{cases} \min \{(1-\theta)\gamma_{1A}, (1-\theta)\gamma_{1B}\} & x_1 \leq \alpha_A, \alpha_B; \\ \min \{\theta\gamma_{1A}, (1-\theta)\gamma_{1B}\} & \alpha_A < x_1 \leq \alpha_B; \\ \min \{(1-\theta)\gamma_{1A}, \theta\gamma_{1B}\} & \alpha_B < x_1 \leq \alpha_A; \\ \min \{\theta\gamma_{1A}, \theta\gamma_{1B}\} & \alpha_A, \alpha_B < x_1. \end{cases} \quad (14)$$

A similar stage cost is defined for Defender 2. From Defender 1's perspective, the main idea is to give a higher weight to the attacker that has a higher probability of being closer.

The last row of Table I shows that the performance with the modified cost approaches that of the optimal centralized setup. The θ values converge to 0.9397 after approximately 60 iterations.

Figure 3 shows the evolution of the $\theta(n)$ for the modified cost setup. The figure shows two different initializations: $\theta(1) = 0.5$ (solid line) and $\theta(1) = 0.0$ (dashed line). Both appear to converge to the same limiting value of θ .

IV. ANALYSIS OF CONVERGENCE

The implications of convergence in θ are the following “consistency” conditions: 1) each defender is employing a policy that is optimal *with respect to* its model of the other defender and 2) each defender's behavior conforms with the model assumed by the co-defender.

We will analyze a idealized version of the evolutionary cooperation setup and establish convergence of the θ iterations. The discussion only will be for the stage cost of (11), but the analysis for the modified cost (14) is similar.

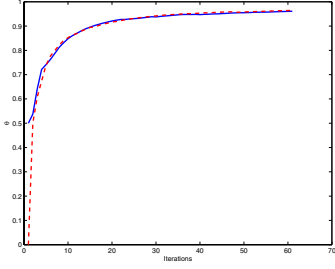


Fig. 3. Convergence of θ in evolutionary modified cost setup.

A. Idealized Iterations and Main Result

The first of two idealizations is that the policy implemented during an interval is a “smoothed” optimal policy as follows. Let $J_{dc, \theta(n)}$ denote the optimal decentralized cost vector from (12) for iteration interval $I(n)$. Accordingly, the optimal policy (for each defender) over iteration interval $I(n)$ at state s_i is

$$\begin{aligned} \mu_i(s_i) &= \arg \min_{u_i \in U(x_i)} g_{dc}(s_i, u_i, \theta(n)) + \rho T_{dc}(s_i, u_i, \theta(n))^T J_{dc, \theta(n)}^* \end{aligned}$$

We will replace this policy with a randomized “softmax” version. For an admissible control action, $a \in U(x_i)$, define

$$v(a; s_i) = g_{dc}(s_i, a, \theta(n)) + \rho T_{dc}(s_i, a, \theta(n))^T J_{dc, \theta(n)}^*$$

For example, if $U(x_i) = \{-1, 0\}$, then

$$\begin{aligned} v &= \begin{pmatrix} v(-1; s_i) \\ v(0; s_i) \end{pmatrix} \\ &= \begin{pmatrix} g(s_i, -1, \theta(n)) + \rho T(s_i, -1, \theta(n))^T J_{dc, \theta(n)}^* \\ g(s_i, 0, \theta(n)) + \rho T(s_i, 0, \theta(n))^T J_{dc, \theta(n)}^* \end{pmatrix}. \end{aligned}$$

The “smoothed” optimal policy is a randomized version of the optimal policy, where

$$Pr [u_i(k) = a \mid s_i(k)] = \frac{e^{-v(a; s_i(k))/\tau}}{\sum_{\bar{a}} e^{-v(\bar{a}; s_i(k))/\tau}}. \quad (15)$$

Such “softmax” smoothing is commonly introduced in learning algorithms (e.g., [6] [2]) to encourage exploration. The (temperature) parameter $\tau > 0$ regulates the degree of exploration. As $\tau \rightarrow 0$, the softmax chooses the maximizing (in our case, minimizing) action with probability increasingly close to one.

An important consequence of the smoothed optimal policy is the following theorem.

Proposition 4.1: The attacker dynamics (5) and defender dynamics (1) under w probabilities (2)–(4a) and smoothed policies (15) form an aperiodic irreducible Markov chain.

The second idealization involves $\theta_{\text{obs}}(n)$.

Suppose that each defender constructs an optimal policy according to (11)–(12) with model parameter θ and employs a smoothed optimal policy as in (15). Via standard methods in Markov chains (e.g., [14, Chapter 6]), Proposition 4.1 implies that there exists a unique stationary probability distribution, which we denote π_θ . Define

$$\Pi(\theta) = Pr [x_2 > x_1],$$

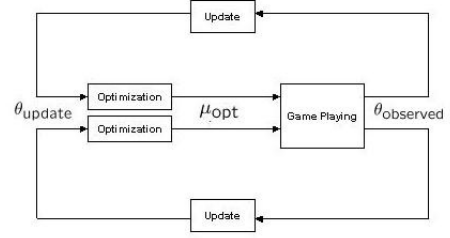


Fig. 4. Algorithm 4.1 flow diagram.

where the probability is with respect to the stationary distribution π_θ . We will assume that

$$\theta_{\text{obs}}(n) = \Pi(\theta(n)) + \delta(n) \quad (16)$$

where the $\delta(n)$ form a uniformly bounded random sequence with

$$E \{ \delta(n) | \theta(n) \} = 0.$$

In the simulations of the previous section, $\theta_{\text{obs}}(n)$ reflected the percentage of times $x_2(k) > x_1(k)$ over stages $k \in I(n)$. The idealization (16) assumes that $\theta_{\text{obs}}(n)$ is a noisy measurement of the exact steady state probability $\Pi(\theta(n))$.

For clarity of exposition, we now state the complete idealized iterative algorithm:

Algorithm 4.1:

1) *Initialization:*

- a) $k = 0$.
- b) $\theta(0) = \theta_0$.

2) *Iteration n :*

- a) Based on $\theta(k)$, each defender designs an optimal policy based on the decentralized dynamics, probabilities, and cost function (7)–(12).
- b) Over interval $I(k)$, each defender employs a smoothed optimal policy (15), for the dynamics defined by equations (1), (10), and (2)–(4a).
- c) At the end of interval $I(n)$, each defender measures $\theta_{\text{obs}}(n)$ according to (16).

3) *Update:*

- a) $\theta(n+1) = \theta(n) + \frac{1}{n+1}(\theta_{\text{obs}}(n) - \theta(n))$.
- b) $n = n + 1$.

The algorithm is illustrated in Figure 4.

We are now in a position to state the main result:

Theorem 4.1: In the framework of Algorithm 4.1,

$$\lim_{n \rightarrow \infty} \theta(n) = \theta^*,$$

for some θ^* , almost surely.

The remainder of this section is devoted to the proof of Theorem 4.1.

B. Proof of Theorem 4.1

The key element of the proof is to show that the function

$$\theta \mapsto \Pi(\theta)$$

is continuous. In the framework of Algorithm 4.1, the θ iterations

$$\theta(n+1) = \theta(n) + \frac{1}{n+1}(\Pi(\theta(n)) - \theta(n) + \delta(n)) \quad (17)$$

satisfy the stochastic approximation assumptions (e.g., [11] [12]) so that the limit set of the iterations (17) is determined by the continuous differential equation

$$\dot{\theta} = -\theta + \Pi(\theta). \quad (18)$$

In our case, the flow induced by (18) evolves over $[0, 1]$. As long as $\Pi(\cdot)$ is continuous, the result of [12, Theorem 1.2] implies that the limit set of (17) is an equilibrium point

$$\theta^* = \Pi(\theta^*),$$

almost surely.

We will prove that $\Pi(\theta)$ is continuous through a series of claims, which are stated without proof for the sake of brevity.

Claim 4.1: For any $s_i \in S_i$ and $u_i \in U(x_i)$, the decentralized stage cost (11), $g_{dc}(s_i, u_i, \theta)$ and decentralized transitions probabilities (13) $T_{dc}(s_i, u_i, \theta)$ are both continuous functions of θ .

Claim 4.2: For any $s_i \in S_i$, the optimal decentralized cost $J_{dc}^*(s_i)$ defined by (12) is a continuous function of θ .

Claim 4.3: Let $T_c(s; \theta)$ denote the overall (centralized) state transition probabilities (as in (6)) induced by smoothed decentralized θ -dependent optimal policies as in (15). For any $s \in S$, $T(s; \theta)$ is a continuous function of θ .

Claim 4.4: The function $\Pi(\theta)$ is continuous.

Proof: In the context of Claim 4.3, let π_θ denote the stationary probability vector for $T_c(s; \theta)$. It is easy to see that

$$\Pi(\theta) = \sum_{s \in S: x_1 < x_2} \pi_\theta(s).$$

Reference [15] shows that the stationary probability vector is a continuous function of the transition matrix. From Claim 4.3, we have that $\Pi(\theta)$ is continuous. \square

V. CONCLUDING REMARKS

We have developed a framework for cooperation via iterative egocentric modeling. We present both a simulation study and an analytical proof of convergence for an idealized scenario.

There is a strong relationship between the method discussed here and the area of learning in games (e.g., [2] [16] [3] as well as [17] [18]). In these methods, agents also make a model of opposing agent strategies using empirical data. Unlike the method discussed here, each iteration requires agents to employ an optimal *response* to the empirically characterized strategies of other agents. In our case, we never presumed a policy for the other agent in order to derive an optimal response. Rather, each agent employs a best response to a highly simplified behavioral model of the other agent, which is more in line with a bounded rationality model of adaptation. Accordingly, the implication of convergence in learning in

games is a Nash equilibrium, whereas convergence in our case implies a modeling and optimization consistency condition.

In this work, the evolving parameter, θ , was a scalar. An obvious next step in this work is to consider situations involving multiple parameters, e.g., to allow more sophisticated models of agent interactions. Although such an approach has been demonstrated in [8] and [9], proof of convergence remains elusive. The obstacle is that one must assess the asymptotic behavior of a multivariate differential equation, e.g., to apply methods from stochastic approximation. In the scalar case, the analysis is straightforward without explicit knowledge of $\Pi(\theta)$ other than continuity and bounded range. In the multivariate case, definite statements regarding asymptotic behavior would be elusive in the absence of additional structural knowledge.

ACKNOWLEDGEMENTS

Research supported by NSF grant #ECS-0501394, AFOSR/MURI grant #F49620-01-1-0361, and ARO grant #W911NF-04-1-0316.

REFERENCES

- [1] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 2000.
- [2] D. Fudenberg and D. Levine, *The Theory of Learning in Games*. Cambridge, MA: MIT Press, 1998.
- [3] H. Young, *Strategic Learning and its Limits*. Oxford University Press, 2006.
- [4] D. Wolpert, K. Wheeler, and K. Tumer, "General principles of learning-based multi-agent systems," in *Third International Conference on Autonomous Agents*, 1999, pp. 77–83.
- [5] Y. Shoham, "Multiagent reinforcement learning: A critical survey," 2003, preprint, <http://robotics.stanford.edu/~shoham/YoavPublications.htm>.
- [6] R. Sutton and A. Barto, *Reinforcement Learning*. Cambridge, MA: MIT Press, 1998.
- [7] D. Bertsekas and J. Tsitsiklis, *Neuro-dynamic programming*. Belmont, MA: Athena Scientific, 1996.
- [8] C.-H. Hsu and J. Shamma, "A decomposition approach to scheduling of failure prone transfer lines," in *System Theory: Modeling, Analysis, and Control*, T. Djaferis and I. Schick, Eds. Kluwer Academic Publishers, 1999.
- [9] V.-W. Seah, "Decomposition based control of failure prone production lines," Master's thesis, UCLA, Department of Mechanical and Aerospace Engineering, 2002.
- [10] M. Earl and R. D'Andrea, "A study in cooperative control: The Roboflag drill," in *Proceedings of the American Control Conference*, Anchorage, Alaska, 2002.
- [11] H. Kushner and G. Yin, *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, 1997.
- [12] M. Benaïm, "A dynamical system approach to stochastic approximations," *SIAM Journal of Control and Optimization*, vol. 34, no. 2, pp. 437–472, 1996.
- [13] D. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995.
- [14] D. Bertsekas and J. Tsitsiklis, *Introduction to Probability*. Athena Scientific, 2002.
- [15] C. Meyer, "The condition of a finite markov chain and perturbation bounds for limiting probabilities," *SIAM Journal on Algebraic Discrete Mathematics*, vol. 1, pp. 273–283, 1980.
- [16] H. P. Young, *Individual Strategy and Social Structure*. Princeton, NJ: Princeton University Press, 1998.
- [17] J. Shamma and G. Arslan, "Dynamic fictitious play, dynamic gradient play, and distributed convergence to Nash equilibria," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 312–327, 2005.
- [18] G. Arslan and J. Shamma, "Distributed convergence to Nash equilibria with local utility measurements," in *43rd IEEE Conference on Decision and Control*, 2004, pp. 1538–1543.