GAME-THEORETIC MISSILE DEFLECTION IN NETWORK CENTRIC WARFARE

Michael Maskery
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, V6T 1Z4, Canada
Email: mikem@ece.ubc.ca

Vikram Krishnamurthy
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, V6T 1Z4, Canada
Email: vikramk@ece.ubc.ca

Christina O'Regan
Defence Research and Development, Canada
Ottawa, K1A 0Z4, Canada
Email: Christina.Oregan@drdc-rddc.gc.ca

***Abstract*— We construct a model for a distributed network of agents charged with defending a flotilla of ships and investigate how they can achieve their objective in a distributed manner. The dynamics of the model and the objectives are studied in detail, and a decentralized missile deflection algorithm based on solving a stochastic shortest path game is given. The main result is a complete formulation of the missile deflection problem as a stochastic shortest path game, and a distributed algorithm for computing the Nash equilibrium when the game is purely collaborative.**

I. INTRODUCTION

Network centric warfare is a military concept that allows forces to share information and synchronize operations to a degree never before attainable [1]. However, one must balance integration with autonomy; elements of a network-centric force must function independently if the network itself is attacked, and moreover, should conserve bandwidth by making some combat decisions locally, even in optimal operating conditions. Thus, one is led to investigate how to organize a network-centric force that allows decision makers to coordinate effectively while retaining their ability to act independently.

In this paper, we investigate a distributed decision making solution to the above problem, in the context of missile deflection. The theory of Markov decision processes, [2], and its extension to stochastic games [3], [4], addresses the problem of how single and multiple agents may act optimally in a dynamic environment, and hence provides an appropriate analytical vehicle for our investigation. Since missile attacks are transient events, we are able to formulate the problem as a stochastic shortest path (i.e. transient) game.

Most related work in game-theoretic missile deflection is focused on platform centric models, such the pursuit-evasion games between defensive and anti-ship missiles of [5] and [6].A network centric model, applicable to missile deflection and employing game theory, is presented in [7], but it assumes centralized control. Decentralized control outside of game theory has been extensively studied, see [8].

In contrast, we use game theory to model *allied* players, who interact in an environment that changes dynamically as the missile threat changes. This viewpoint allows us to investigate the possiblity that players may coordinate to achieve a common goal, without explicit communication, thus distributing the decision-making abilities of a network centric force. This type of cooperation is similar in nature to coordination games, collaborative games, and team games. Of course, we allow for both conflict and collaboration in our model, but it should be kept in mind that the ultimate goal is to discover solutions in which all players benefit as much as possible.

We consider the following scenario. A flotilla of ships, communicating over a common data network, faces an attack by a fixed number of missiles, which appear randomly on the perimeter of a fixed, discrete physical space over time, and move toward the (relatively stationary) ships under known targeting and guidance laws. Each ship carries electronic countermeasures (ECMs), which are either decoys or jamming signals used to deflect or confuse the missiles. A group of autonomous agents is assigned the task of defending the ships, each agent having control over a subset of ECMs and charged with defending a given subset of ships. Given the missiles' positions and dynamics at any time index, the task of each agent is to deploy a sequence of ECMs so as to defend their ships, without exact knowledge of the decisions being made by other agents.

Assuming the dynamics of the missile process are known and Markovian, and the system state is fully observed by each agent, this scenario can be formulated as a stochastic shortest path game. The assumption that the missiles evolve according to a (controlled) Markov process is standard. Agents associate a cost with each missile configuration and search for ECM policies that minimize the sum of expected future costs, in light of the fact that others are doing the same. By formulating the process as a game, a joint ECM policy that is in Nash equilibrium can be computed in a decentralized fashion, thereby making it suitable for network centric warfare.

II. FORMULATION AS A STOCHASTIC SHORTEST PATH GAME

We begin by defining a general sum stochastic shortest path game for the missile deflection problem. This combines the features of stochastic games, as in [3] and [4], with the optimization of a transient Markov decision process, as in [2]. Fundamental to this game formulation are the following: a discrete state space $\mathbf{S}$, a set of defending agents L, a physical

0-7803-8812-7/05/$20.00

state space $\mathbf{S}_p$, a set of targets $T \subset \mathbf{S}_p$ defended by the agents, a set of countermeasures $U^l(i)$ available to agent l in state i, a real-valued cost function $g^l(i,u)$, a set of incoming missiles M, and a discrete-time Markov process describing the evolution of the missiles on $\mathbf{S}$, as influenced by the agents' ECM choices. States in $\mathbf{S}$ reflect aggregate information about the missile processes, including positions, status, and current targets. These details are assumed to be fully observed by the agents via sensor information distributed over a common data network.

At a minimum, $\mathbf{S}$ is comprised of $|M|$ copies of $\mathbf{S}_p$, since the positional information of all missiles must be incorporated into each state. To take full advantage of the correspondence between these two spaces, define $\gamma(m,i) : M \times \mathbf{S} \to \mathbf{S}_p$ to be the physical position of missile m in state i.

Let X_k denote the state of the process at time k and define $U(i) = \times_{l \in L} U^l(i)$ to be the set of available joint ECM actions in state i. Assume the aggregate missile process $\{X_k, k = 0,1,2,\dots\}$ evolves according to a Markov chain with transition probabilities

$$p_{ij}(u) = Pr(X_{k+1} = j | X_k = i, u_k = u), \tag{1}$$

where $u_k \in U(i)$ is the joint action at time k.

Some countermeasures, such as chaff and physical decoys, are limited to a single "launch" control. These may be incorporated into our model by expanding the state space so that the process reflects whether or not each decoy has been released, and the control set can be restricted so that only decisions can be made regarding remaining decoys. The optimal policy will then yield optimal launch times for each decoy. For details on how to formulate this problem with relatively low compuational overhead, we refer the reader to [9].

In order to minimize network dependence during the missile attack, each agent uses an independently executable stationary randomized ECM policy. Let $\Delta(Z)$ represent the set of probability distributions on a set Z, and let $\mathbf{U}^l$ (resp. $\mathbf{U}$) represent the vector of available individual (resp. joint) actions over all states i. Then we are led to the following definition.

Definition 2.1: An independently executable stationary randomized policy for agent l is a mapping μ^l from $\mathbf{S}$ to $\Delta(\mathbf{U}^l)$. The probability of employing ECM u in state i is denoted by $\mu^l(i,u)$.

Such a policy is essentially a lookup table prescribing a probability distribution over actions for each missile state.

In accordance with game-theoretic notation, define the stationary policy profile $\mu = \times_{l \in L}(\mu^l)$, and denote by μ^{-l} the policies all agents but l, so that $(\nu, \mu^{-l}) : \nu \in \Delta(\mathbf{U}^l)$ denotes a policy profile, with $\mu = (\mu^l, \mu^{-l})$. Note that, since agents independently randomize their actions, μ is restricted to a set of product distributions on $\Delta(\mathbf{U})$.

An important assumption, typical of stochastic shortest path problems (see [2]), is that the process $\{X_k, k = 0,1,2,\dots\}$ is transient. That is, there exists in $\mathbf{S}$ a cost-free absorbing state t associated with termination of the process (i.e. all missiles have reached their targets or have been neutralized), which is reached in finite time with probability one. Such an assumption has been in use in stochastic game theory since its introduction in [3]. Of course, since the missiles clearly have a finite flight time, this is a reasonable assumption, regardless of the ECM policy used. The appropriate technical condition here is that all policies are *proper* policies (see [2]), that is, stationary control policies such that

$$\max_{i \in \mathbf{S}} (Pr(X_{|\mathbf{S}|-1} \neq t | X_0 = i, \mu)) < 1, \tag{2}$$

A proper policy is one which, if followed, guarantees that a path can be followed with positive probability from any state to t, passing through every other state at most once, implying an almost surely finite termination time.

Each stationary policy profile μ induces a Markov chain, with associated transition probabilities

$$p_{ij}(\mu) = \sum_{u \in \mathbf{U}} p_{ij}(u)\mu(i,u). \tag{3}$$

Since the chain is transient, it will be convenient to define the transition matrix $P(\mu)$ as the matrix of values $p_{ij}(\mu)$ over all $i, j \neq t$. Hence $P(\mu)$ is substochastic.

The objective of agent l is to defend a subset of targets $T(l) \subset T$, where the exact nature of the defense is discussed in Section IV. One may assume that each ship is an agent that defends itself using its set of on-board countermeasures, but the model here is more general, allowing agents, targets, and countermeasures to be freely assignable throughout the flotilla. To motivate the defense, define functions ζ^l and ζ on $\mathbf{S}$, such that $\zeta^l(i) = 1$ whenever state i corresponds to at least one missile reaching $T(l)$, and $\zeta(i) = 1$ whenever state i corresponds to at least one missile reaching T. Formally, we may write

$$\zeta^l(i) = \delta(\{\gamma(m,i) : m \in M\}, T(l)), \tag{4}$$

where δ is the set indicator function; $\delta(A,B) = 1$ whenever $A \cap B \neq \emptyset$.

Each agent is assigned a cost function $g^l(i,u)$, the cost incurred by player l each time state i is visited and joint ECM u is used. This cost generally increases with proximity to the target set, and the goal of each agent is to minimize the sum of expected costs incurred over the length of the process. The expected instantaneous cost associated with a given policy is

$$g^l(i,\mu) = \sum g^l(i,u)\mu(i,u). \tag{5}$$

Define $\mathbf{g}^l(\mu)$ to be vector of such values over states $i \neq t$.

Assume now that the players have chosen a joint stationary policy μ. Since the Markov chain induced by $P(\mu)$ is transient, the mean time spent in state j given that the process starts in state i is given by the ij^{th} entry of $(I - P(\mu))^{-1}$ (see [10] for existence and derivation). Thus, the expected cost, starting from state i, is given by the i^{th} row of

$$(I - P(\mu))^{-1}\mathbf{g}^l(\mu). \tag{6}$$

Note that $P(\mu)$ is relatively sparse, since the missiles can only transition to neighbouring physical states. Hence the inverse above is not as difficult to compute as it appears; sparse matrix methods and/or Monte Carlo simulation can be applied.

The stochastic shortest path game involves each agent strategically choosing a stationary policy that is optimal given a belief about the policies of other agents, which are in turn expected to be optimal. The joint policy thus chosen should ideally guide the stochastic process to termination with a minimal cost incurred by each agent.

III. Dynamics of Incoming Missiles

The Markov process $\{X_k, k = 0, 1, 2, \ldots\}$ described above is an aggregate of $|M|$ independent missile flight processes, which are also Markovian. The dynamics of these individual processes are discussed below.

We allow a missile to be either active, potential, or terminated. An active missile is an immediate threat, taking on a physical position on S_p and moving toward a current target in T. A potential missile is an as-yet untracked missile, which may randomly appear on the perimeter of S_p and acquire a target at any time. A terminated missile is one that has reached a target, has otherwise terminated its flight, or has acquired a trajectory that ensures it is no longer a threat. The missiles are assumed to act independently and according to identical dynamics, so that each missile m moves according to a discrete-time homogeneous Markov decision process $\{Y_k, k = 0, 1, 2, \ldots\}$ on the state space $\tilde{\mathbf{S}} = \mathbf{S}_p \times T \cup \{r_1, t_1\}$. where r_1 corresponds to potential status, and t_1 corresponds to termination. The transition probabilities for an individual process are defined as

$$\tilde{p}_{ij}(u) = Pr(Y_{k+1} = j | Y_k = i, u_k = u), \tag{7}$$

where $u_k \in \mathbf{U}$ is the joint ECM action at time k. An active missile observes the apparent targets, chooses one, and moves toward it with random perturbations. We place the following conditions on the transition probabilities:

$$\tilde{p}_{it_1}(\cdot) = 1 \text{ whenever } \zeta(i) = 1, \tag{8}$$

$$\tilde{p}_{t_1t_1}(\cdot) = 1, \tag{9}$$

$$\tilde{p}_{r_1j}(\cdot) = \beta(j), \tag{10}$$

for some probability distribution β. (8) implies that the process terminates when the missile reaches the target set, (9) that the terminal state is absorbing, and (10) that the initial distribution is independent of ECM control.

The stochastic nature of the transitions can be viewed as a consequence of randomness in the targeting mechanism, physical movement, and random errors in the sensor network's estimates of the missile position. Physically, a missile moves, with high probability, directly along an intercept course with its chosen target, only rarely deviating to the left or right. Deviations can be associated with random errors in the velocity estimates of the proportional guidance system, which is described in [11].

The targeting mechanism can be of several forms, although it is restricted here to be Markovian in nature. For example, the missile can retain its current target with some high probability p, and otherwise choose randomly from other nearby targets (e.g. weighted according to proximity). In the simple case that $p = 0$, the targeting system becomes memoryless, and hence the p_{ij} can be expressed as a convex combination of transition probabilities given the available target choices. In such a case there is no need for T to be a dimension of $\tilde{\mathbf{S}}$.

More generally, we can view the targeting system as a receiver operating in a noisy environment. At each time index, the missile receives information regarding potential true and false target positions over a discrete memoryless channel (see, for example, [12]), which randomly reports the potential target positions as s' when in fact it is s with probability $q(s'|s)$. The missile then chooses its target according to some law based on s', and moves according to transition probabilities $\tilde{p}_{ij}(s')$. In this case we may write the true transition probabilities as

$$\tilde{p}_{ij}(u) = \sum_{s' \in \mathbf{S}_p} q(s'|s)\tilde{p}_{ij}(s'(u)). \tag{11}$$

The distortion may vary with the missile state, in which case the function q in (11) can also be dependent in i.

Given the individual, i.i.d. missile dynamics, we can define the aggregate missile process $\{X_k, k = 0, 1, 2, \ldots\}$ as $|M|$ copies of the Y_k process as follows. The process evolves on $\mathbf{S} = (\tilde{\mathbf{S}})^{|M|}$, and has transition probabilities given by

$$p_{ij}(u) = \tilde{p}_{i_1j_1}(u)\tilde{p}_{i_2j_2}(u) \ldots \tilde{p}_{i_{|M|}j_{|M|}}(u) \tag{12}$$

under joint action u, where $i = (i_1, i_2, \ldots, i_{|M|})$. To reconcile this definition completely with the aggregate process of Section II, let $t \in \mathbf{S}$ correspond to the state for which all missile processes have reached their individual terminal states.

IV. ECM Metrics

The sole purpose of the objective function $g^l(i, u)$ is to ensure that agent l protects its target set. Protection is a broad term, and hence there are a wide range of appropriate objective functions. We consider only two here; one probability-based and one distance-based. It should be noted that two special categories of objectives are positional objectives, in which $g^l(i, u)$ is independent of u, and collaborative objectives, in which $g^l(i, u)$ is identical for all l. For simplicity, we consider only positional objectives, and later will present an algorithm designed for collaborative ones. Note that collaborative objectives are especially attractive, since agents will not face the temptation to guide the missiles each others' target sets in order to prematurely terminate the process and save themselves.

A. Minimizing the probability of kill

A reasonable defensive goal for agent l is to minimize the probability that the missiles will reach the target set $T(l)$ from any initial state. This is achieved by assigning cost function

$$g^l(i) = \zeta^l(i), \tag{13}$$

and using the following lemma, which is a consequence of the elementary properties of transient Markov chains.

Lemma 4.1: If g^l is as in (13), then the i^{th} entry of $(I - P(\mu))^{-1}\mathbf{g}^l$ can be interpreted as the probability of reaching $T(l)$ from state i under policy μ.

Proof: A general proof is given in [10]. By (8), states j such that $g^l(j) = 1$, are visited at most once, hence the expected number of visits to such states, given by the j^{th} columns of $(I - P(\mu))^{-1}$, is also the probability that they will ever be reached. Summing the probabilities, which correspond to independent events, over all such states gives the result. ■

The objective above can be extended if one can identify a set of "safe" exit states in $\mathbf{S}$ and coerce collaboration by providing incentive for agents to seek out these states. Assume there exists a set of guaranteed exit points $H \subset \mathbf{S}$ such that

$$\forall\ i \in H, \zeta(i) = 0, \tag{14}$$

$$\forall\ i \in H, p_{it}(u) = 1,\ \text{for some } u. \tag{15}$$

H can be interpreted as the set of states for which all targets become out of range of the missiles. That is, the missiles' trajectories are such that they can no longer recover to threaten the targets.

Introduce a "dummy" agent h, with a single null action in every state, such that $T(h) = H$, and consider the cost function

$$g^l(i) = \begin{cases} \zeta^l(i) - \frac{1}{|L|}\zeta^h(i),\ l \in L \\ \zeta^l(i) - \zeta(i),\ l = h \end{cases} \tag{16}$$

It can be shown that, for a given policy profile μ, the i^{th} entry of $(I - P(\mu))^{-1}\mathbf{g}^l$ equals

$$Pr(T(l) \text{ is reached}) - \frac{1}{|L|}Pr(T(h) \text{ is reached}), \tag{17}$$

for $l \in L$. The proof is similar to that of Lemma 4.1. Therefore, under cost (16), the agents seek a policy that minimizes their probability of kill while maximizing the probability that the missile process safely terminates through H. This explicit incentive for safe exists counteracts the tendency of agents to destroy each other in an attempt to prematurely terminate the missile threats.

B. Maximizing the minimum distance

An alternative objective is to maximize the minimum distance achieved between missiles and targets over the entire history of the engagement. As this is clearly not a Markov-type objective, we augment the state space to facilitate a problem reformulation.

For $i \in \mathbf{S}_p$ and $A \subset \mathbf{S}_p$, define a real-valued distance function $d(i, A)$, with $d(i, A) = 0$ whenever $i \in A$. The actual function can be chosen quite arbitrarily, but should represent some desired notion of distance. A particularly convenient notion is that of "worst-case vulnerability," defined as follows.

Define the set $D_0^l = T(l)$, choose $\varepsilon \geq 0$, and define recursively the sets

$$D_k^l = \left\{ i \in S_p \backslash \bigcup_{m=0}^{k-1} D_m^l : \tilde{p}_{ij}(u) > \varepsilon,\ \text{some } j \in D_{k-1}^l, u \right\}. \tag{18}$$

The collection $\{D_k^l : k = 0 \ldots\}$ partitions S_p, with state i in D_k if and only if the minimum number of steps from i to the target is k. Thus we define the distance function on S_p by

$$\begin{aligned} d(i) &= k : i \in D_k \\ &= \min_{\mu} \left(k : Pr(\zeta^l(X_k) = 1 | X_0 = i, \mu) > \varepsilon^k \right). \end{aligned} \tag{19}$$

By the proper policy assumption, $d(i, A) < |S_p|$, unless i leads directly to termination, in which case the distance is either zero ($i \in A$) or infinite. No change in policy occurs if we replace infinite values by zero ones, since control in and after such states is irrelevant. $d(i, A)$ thus measures the minimum number of steps it would take for a missile in position i to reach either the set A or the terminal state; it depends on the transition probabilities of the missile process, but not on the particular control used.

To efficiently formulate the problem, we group agents with the same objective together to form a reduced state space. That is, if l and l' have the same objective (i.e. $T(l) = T(l')$), then set $\sigma(l) = \sigma(l')$, and define $\bar{L}$ to be the image space of σ operating on L. If we denote the probabilities in (12) by $r_{ij}(u)$, and the corresponding state space by $\mathbf{S}_{old}$, then the transition probabilities on the new state space $\mathbf{S}$ reflecting the history of minimum distances are given by

$$p_{ij}(u) = r_{\tilde{i}\tilde{j}}(u) \prod_{m \in M, l \in \bar{L}} \chi_{ijml}, \tag{20}$$

where $\tilde{i}$ represents i restricted to $\mathbf{S}_{old}$, and

$$\chi_{ijml} = \delta(d(\gamma(m, j), T(l)) = \min(d(\gamma(m, i), T(l)), i_{ml})). \tag{21}$$

where i_{ml} represents i restricted to the ml^{th} copy of D. Note that the stochastic process so defined is non-increasing in each new dimension.

The costs $g^l(i, u)$ are defined so that an agent is penalized whenever the missile reaches a historic new minimum distance. That is,

$$g^l(i) = \begin{cases} \max(0, i_{ml} - d(\gamma(m, i), T(l))),\ i \in S\backslash\{t\}, \\ 0, \quad i = t. \end{cases} \tag{22}$$

An agent will then compute a policy which minimizes these penalties, and hence maximizes the minimum distance.

Since the number of states in $\mathbf{S}$ increases exponentially with both the number of target sets and the number of missiles, the objective of maximizing the minimum distance is most useful in a purely collaborative game setting, with only a small number of missiles.

We remark that the construction above may yield useful information about a chosen policy. For example, the many new absorbing states convey information about the number of targets hit by the end of the missile attack. Note also that since the extra dimensions are artificial, the distance function can be coarsened, refined, or changed without losing information on the missile dynamics, although the optimal policies will in general be changed.

V. DECENTRALIZED ECM ALGORITHMS

In this section we consider the decentralized computation and execution of a joint ECM policy. Agents' policies are an optimal response to each other, and hence in equilibrium. Moreover, the policies are independently executable, and hence require no communication once the missile attack has been launched.

A. Optimal solution for a single agent

If μ^{-l} is held fixed, agent l can compute an optimal stationary policy by viewing the problem as a stochastic shortest path Markov decision process. Such problems have been extensively studied, for example in [2]. We focus on the linear programming solution, but note that many other methods, such as value iteration, policy iteration, Monte Carlo methods and approximation can also be used to obtain suitable policies. Let α be an initial probability distribution on $\mathbf{S}$. The optimal policy is found by first solving the following linear program for π^l (see [2] for complete details).

$$\begin{aligned} &\text{minimize} \sum_{i\in\mathbf{S}\setminus\{t\},u\in U^l(j)} g^l(i,(u,\mu^{-l}))\pi^l(i,u) \\ &\text{subject to } (\forall\ j\in\mathbf{S}) \\ &\sum_{u\in U^l(j)} \pi^l(j,u) = \sum_{i\in\mathbf{S}\setminus\{t\},u\in U^l(j)} p_{ij}((u,\mu^{-l}))\pi^l(i,u) \\ &\qquad\qquad + \alpha(j), \\ &\pi^l(j,u)\geq 0\ \forall\ u\in U^l(j). \end{aligned} \tag{23}$$

The optimal policy μ^l is a mapping from $\mathbf{S}\rightarrow\Delta(\mathbf{U}^l)$, derived from π^l by:

$$\mu^l(i,u_r) = \frac{\pi^l(i,u_r)}{\sum_{u\in U^l(i)}\pi^l(i,u)}. \tag{24}$$

This policy minimizes the expected cost, and hence minimizes $(I-P((\nu,\mu^{-l}))^{-1}\mathbf{g}^l((\nu,\mu^{-l}))$ over all possible ν.

If a single agent has control of all countermeasures, then the above program yields an optimal centralized policy. This corresponds the case where all agents yield their autonomy to a central authority, which dictates actions in real time. Note that such a solution is expensive to implement, and leaves the flotilla vulnerable in case of communication breakdown. Hence the need for distributed, game-theoretic solutions.

B. Nash equilibrium solution

The computation above relies on the assumption that μ^{-l} is known and held fixed, but when all agents are simultaneously attempting to perform optimally, the situation is much more complex. In this case a reasonable option is to find and execute stationary policies which are in Nash equilibrium.

Definition 5.1: A Nash equilibrium in stationary policies is a joint policy μ such that for each player l and all $\nu\in\Delta(\mathbf{U}^l)$,

$$(I-P(\mu))^{-1}\mathbf{g}^l(\mu)\leq(I-P((\nu,\mu^{-l})))^{-1}\mathbf{g}^l((\nu,\mu^{-l})). \tag{25}$$

Such Nash equilibrium policies always exist (see [4]), and are locally optimal, but generally inferior to centralized policies since they suffer from coordination issues due to their restriction to the space of product distributions. However, this restriction is a benefit insofar as it requires no communication during the execution of the policy, and effectively distributes computation among all agents. Nash policies are based on the notion that each agent's actions are a "best response" to the others' actions, so computation is naturally viewed as an iterative process, with candidate policies being improved upon by one or more players at each iteration until a fixed point is reached. Indeed, iteration is easily seen to lead to equilibrium in the purely collaborative case, as the following lemma states.

Lemma 5.1: Iterated sequential best reply leads to Nash equilibria in collaborative games.

Proof: Since the expected costs coincide for each agent, and since each iteration necessarily reduces these costs (which are bounded below by zero), Monotone Convergence immediately yields the result. ■

Lemma 5.1 suggests the following algorithm for calculating a Nash equilibrium in a purely collaborative game.

Algorithm 5.1: Agents adopt initial policies μ_0^l and i is initialized to zero. Repeat the following:

1) For $l=1,\ldots,|L|$ repeat:
2) Agent l computes a stationary policy μ_{i+1}^l via Program (23), assuming the others use policies
$$(\mu_{i+1}^1,\ldots,\mu_{i+1}^{l-1},\mu_i^{l+1},\ldots,\mu_i^{|L|}).$$
3) If $l=|L|$, check whether the resultant joint policy is a Nash equilibrium. If it is not, reset l, increment i, and continue.

Unfortunately, this algorithm is quite inefficient, since players compute improvements sequentially instead of in parallel. A more efficient method might be to compute an initial policy that is good, but not in equilibrium, so that Algorithm 5.1 may be initialized close to equilibrium, with fewer iterations required.

In the non-collaborative case, the best reply dynamic is not guaranteed to converge to equilibrium, but may still yield acceptable results. Other methods, such as Fictitious Play (see [13]) may yield better results.

VI. NUMERICAL RESULTS

In this section we investigate the following simple missile deflection game. Two ships are threatened by a single missile moving on an 11×11 grid, with x representing the east-west axis, and y representing the north-south axis. Ship one is located at $(0,0)$, the center of the grid, while Ship two is located two units south of center. Two agents wish to execute a policy with the common objective of maximizing the minimum distance achieved between the missile and both ships. Agent one controls a decoy that can be placed to the north, east, south, or west of ship one, while Agent two possesses a jamming device that may confuse the missile, causing it to believe that the decoy is one unit north, south, east, or west of its actual position (the choice being up to agent two).

At each time index, the missile randomly targets either Ship one or the decoy with probabilities 0.2 and 0.8 respectively. Ship two is never targeted directly, but will be destroyed by

3.48	3.49	3.45	3.38	3.33	3.32	3.33	3.38	3.45	3.49	3.48
3.49	2.60	2.59	2.52	2.43	2.41	2.43	2.52	2.59	2.60	3.49
3.45	2.59	1.69	1.65	1.52	1.46	1.52	1.65	1.69	2.59	3.45
3.38	2.52	1.65	0.77	0.65	0.46	0.65	0.77	1.65	2.52	3.38
3.33	2.43	1.52	0.65	0.24	0.23	0.24	0.65	1.52	2.43	3.33
3.33	2.41	1.47	0.46	0.24	0 S1	0.24	0.46	1.47	2.41	3.33
3.33	2.43	1.53	0.66	0.25	0.28	0.25	0.66	1.53	2.43	3.33
3.39	2.53	1.67	0.78	0.21	0 S2	0.21	0.78	1.67	2.53	3.39
3.47	2.62	1.73	0.83	0.28	0.60	0.28	0.83	1.73	2.62	3.47
3.53	2.65	1.73	0.88	1.11	1.40	1.11	0.88	1.73	2.65	3.53
3.53	2.61	1.76	1.86	2.00	2.19	2.00	1.86	1.76	2.61	3.53

Fig. 1. Expected costs for the Nash equilibrium policy.

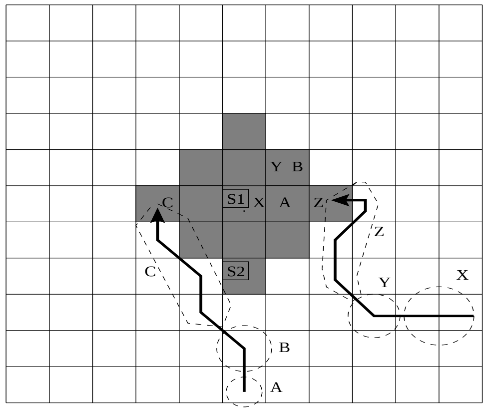

Fig. 2. Sample missile and ECM paths under the Nash policy.

the missile if reached. Similarly, Ship one is destroyed with probability one if the missile reaches it. Once the target has been chosen, the missile correctly determines the most direct path to it with probability 0.7, and moves to the next adjacent grid position j along this path. Otherwise, the missile moves indirectly towards the target, one square to its left or right of this correct grid position j, or stays in its current position. If the missile reaches the perceived decoy position, it detonates with probability 0.5. It terminates with probability 0.02 at any other position. At time zero, the missile appears randomly on the edge of the grid, then moves according to the above dynamics, as influenced by the ECM policy.

Based on the transition probabilities derived from the above dynamics, Algorithm 5.1 was used to compute a stationary Nash equilibrium policy. This policy is characterized as follows. For states with $|x| \leq 2$ and $|y| \leq 2$ (the area of active control), the agents attempt to cooperate to place the perceived target at the nearest state such that $|x| = 2$ or $|y| = 2$. For example, if the missile is at $(1, 0)$, then the decoy is placed there by Agent one, while Agent two emits a signal to project its position to $(2, 0)$. If there is more than one such adjacent state, (e.g. at $(1, 1)$,) the agents must choose one randomly and independently, and are hence vulnerable to the coordination problems. For states outside the active control area, the agents randomly choose ECMs that cause the missile to move toward the nearest state such that $|x| = 2$ or $|y| = 2$. For example, if the missile was at $(5, 0)$, then any joint ECM that causes the missile to believe the target was at $(x, 0), |x| \leq 2$ is used.

This policy is quite reasonable. As a check, the optimal policy was also computed by assuming a single agent had control over all ECM. Except for the coordination problems associated with independently randomizing decisions, the Nash equilibrium policy was almost identical to the optimal one.

In Figure 1, the shaded area indicates the possible false target positions and the marked square indicates the position of the ships. The numbers are the expected cost associated with the process when the missile is in each position, i.e. the difference between the present distance and the expected minimum future distance.

Figure 2 shows two sample missile paths and the associated positions of the false targets in each case. False Targets are placed at positions A, B, C and X, Y, Z during the indicated portions of the two missile flight paths.

VII. SUMMARY

We have presented a complete model of a multiple missile deflection game, including a discussion of the missile dynamics and the development of defensive objectives. Solution via linear programming and iterated best response was proposed and shown to be effective when player objectives coincide. Future work will include introduction of limited communication, more detailed modeling, and conflicting objectives.

VIII. ACKNOWLEDGEMENT

This research was funded by a contract with Department of National Defence, Canada.

REFERENCES

[1] D. Alberts, J. Garstka, and F. Stein, *Network Centric Warfare: Developing and Leveraging Information Superiority*, CCRP Publications, second edition, 1999.

[2] D. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1 and 2, Athena Scientifi c, second edition, 2001.

[3] L. Shapley, "Stochastic games," *Proceedings of the National Acadamy of Sciences USA*, vol. 39, no. 10, 1953.

[4] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*, Springer-Verlag, 1997.

[5] S. Gutman, "On optimal guidance for homing missiles," *Journal of Guidance and Control*, 1979.

[6] Y. Lipman and J. Shinar, "Mixed-strategy guidance in future ship defense," *Journal of Guidance, Control, and Dynamics*, 1996.

[7] J. Cruz and M. Simann et. al., "Game-theoretic modeling and control of a military air operation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 4, 2001.

[8] N. Sandell, P. Varaiya, M. Athans, and M. Safonov, "Survey of decentralized control methods for large scale systems," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, 1978.

[9] M. Maskery and V. Krishnamurthy, "Decentralized missile defection algorithms for network centric warfare," *In Preparation*, 2004.

[10] S. Ross, *Introduction to Probability Models*, Academic Press, sixth edition, 1997.

[11] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.

[12] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley, 1991.

[13] J. Robinson, "An iterative method of solving a game," *Annals of Mathematics*, vol. 54, 1951.