12th ICCRTS

"Adapting C2 to the 21st Century"

Paper #: 102

Title: "A Ghost of a Chance: Polyagent Simulation of Incremental Attack Planning"

Topics: Modeling and Simulation, Network-Centric Experimentation and Applications, Cognitive and Social Issues

Authors: Andrew Trice, Institute for Defense Analyses

Sven Brueckner, NewVectors LLC

Ted Belding, NewVectors LLC

Point of Contact: Andrew Trice

Name of Organization: Institute for Defense Analyses

Address: 4850 Mark Center Dr., Alexandria, VA 22311

Telephone: 703-933-6543

E-mail Address: atrice@ida.org

A Ghost of a Chance: Polyagent Simulation of Incremental Attack Planning[1]

Abstract

One technique for improving a C2 planning process is to explore as broad a range of potential scenarios as possible, while intelligently constraining the search space and managing the uncertainty of outcomes. From a modeling and simulation perspective, one novel way to do this is to employ a "polyagent" modeling construct to produce emergent planning behavior. A polyagent is a combination of a persistent agent (an "avatar") supported by a swarm of transient agents ("ghosts") that assist the avatar in generating and assessing alternative (probabilistic) futures. The ghosts in the model employ pheromone fields to signal, identify, and act on threats and opportunities relative to the goals, which are then reported back to the avatars for integration and decision-making.

The current work implemented a polyagent model of attack planning in a generic spatio-temporal space with Red/Blue forces and multiple targets pursued by Red. The results indicated that Red polyagents enjoy an asymmetrical advantage when force strength and planning behaviors, (specifically the number of steps in the future the ghosts simulate) are identical. However, simulating more than a few steps in the future has either no or negative impact on polyagent performance.

[1] The publication of this paper does not indicate endorsement by the Department of Defense, IDA, or NewVectors LLC, nor should the contents be construed as reflecting the official positions of those organizations.

1. INTRODUCTION

Agent-based models have been used to address a wide variety of C2 problems [e.g., 1, 2, 3, 4]. Adapting such models to C2 has many challenges. One is that as the number of agents and number of decision-making cycles in a C2 setting increases, the set of potential outcomes that could be explored also increases exponentially. Therefore, to be as scalable as possible while still providing reliable results, agent-based models must at some point provide aids to guiding the exploration of the planning space in a manner that both explores as many potential worthy alternatives as possible, yet is still computationally efficient.

Traditional agent-based models execute a single trajectory through the vast space of possible futures of the system that is spanned by the possible state changes of the agents and their shared environment. These state changes may occur probabilistically, especially when it comes to outcomes of interactions with the environment. Any such uncertain outcome in individual actions or local interactions in a non-linear model opens the possibility for the emergence of drastically different outcomes at the system level ("for want of a nail the battle was lost"). The analysis of the structure of this state space that may be full of complex attractors cannot be performed without exploring multiple alternative futures, which, in traditional agent-based modeling approaches requires the repeated execution of the model under varying initial conditions (e.g., different random seeds). In past research we have developed techniques for the automated generation and analysis of multiple runs of a multi-agent simulation model and the adaptive search for interesting features (e.g., phase transitions) in the emergent dynamics of such models [5]. The disadvantage of the "sweep" approach to the analysis of emergent multi-agent system dynamics is that it is an off-line analysis process. With the recently developed polyagent construct, such an analysis may be performed by the agents themselves on the fly, allowing them to select among all the upcoming attractors those with a desirable outcome.

In the case where the agents happen to be doing incremental or local rather than global planning and the agents need to self-organize, exploring a variety of potential interactions during each decision cycle is even more important. What we seek is an agent-based modeling construct that allows us to do so.

2. THE POLYAGENT CONSTRUCT

The polyagent modeling construct [6] has been proposed as a mechanism for addressing some of the shortcomings of traditional agent-based models noted above. An individual polyagent is composed of two key components, an "avatar" and its swarm of "ghosts". The "Avatar" is a persistent agent who takes action in the virtual world, and uses results suggested by the activities of its "ghosts" (see below) to decide its next action. The "ghost" is a transient actor in the virtual world that plays out alternative probabilistic scenarios over some "forecast horizon" of future timesteps[2] in the simulation by interacting through pheromone fields (in the present example, the opposing ghosts and the target).

[2] Note that in other applications (e.g., STRONG-ARM in DARPA RAID) we actually insert our ghosts in the recent past at an "insertion horizon" to observe their trajectory from the past to the present for evolutionary model tuning against real-world observations.

The ghosts effectively act as surrogates for the avatar, which allows the avatar to “play-act” different courses of action and integrate the results to decide the next step. “Play-acting”, in this context, refers to another layer of modeling and simulation executed by the ghosts taking place within the higher-level space inhabited by the avatar. It is important to note that ghosts do not have a narrower scope of responsibility than the avatar as in a typical commander – subordinate relationship; rather, the ghost generally has the same objectives and action alternatives that its corresponding avatar has, only in a different modeling space, where each ghost reacts to the actions of other ghosts, starting from the basis of the last known actions of the avatars. The avatar has full access to the ghosts’ virtual experiences and outcomes, and can plan its avatar actions accordingly in avatar space. While the avatar will generally wait until its ghosts have completed a full virtual cycle before acting, it can cut the ghosts’ cycle short to get the current virtual data if a avatar decision in avatar space is required sooner than this.

A more significant difference between the avatar and its ghosts is the kind of reasoning technology used in the respective entity. We developed this polyagent construct to combine more complex, single-agent reasoning techniques as typically used in classical Artificial Intelligence based agent systems (e.g., Belief-Desire-Intention (BDI) Logic), with Swarm Intelligence reasoning techniques (i.e., many simple agents with emergent population properties) that derive from Artificial Life research. Thus, the avatar is the host for the complex reasoner while the ghosts are the simple agents with emergent population properties.

Polyagent models have been used in a variety of settings, including factory scheduling, robotic vehicle path planning, and characterizing the behavior of other agents [6]. The collaboration of the avatars and ghosts offer the opportunity for the modeler to explore a great variety of planning alternatives in a single run of the model.

We now discuss how we have adapted the polyagent construct in a specific command and control setting that employs incremental attack (and defense) planning.

3. POLYAGENT MODEL DESCRIPTION

Overview

The polyagent model for this simulation is designed to implement a relatively simple attack/defense scenario, in which a Red (offensive) Force, a Blue (defensive) Force, and single or multiple fixed targets are arranged in a grid configuration. The Red forces have the goal of reaching and destroying targets, while the Blue forces have the goal of eliminating Red forces. Figure 1 shows a snapshot of the model during a typical run.

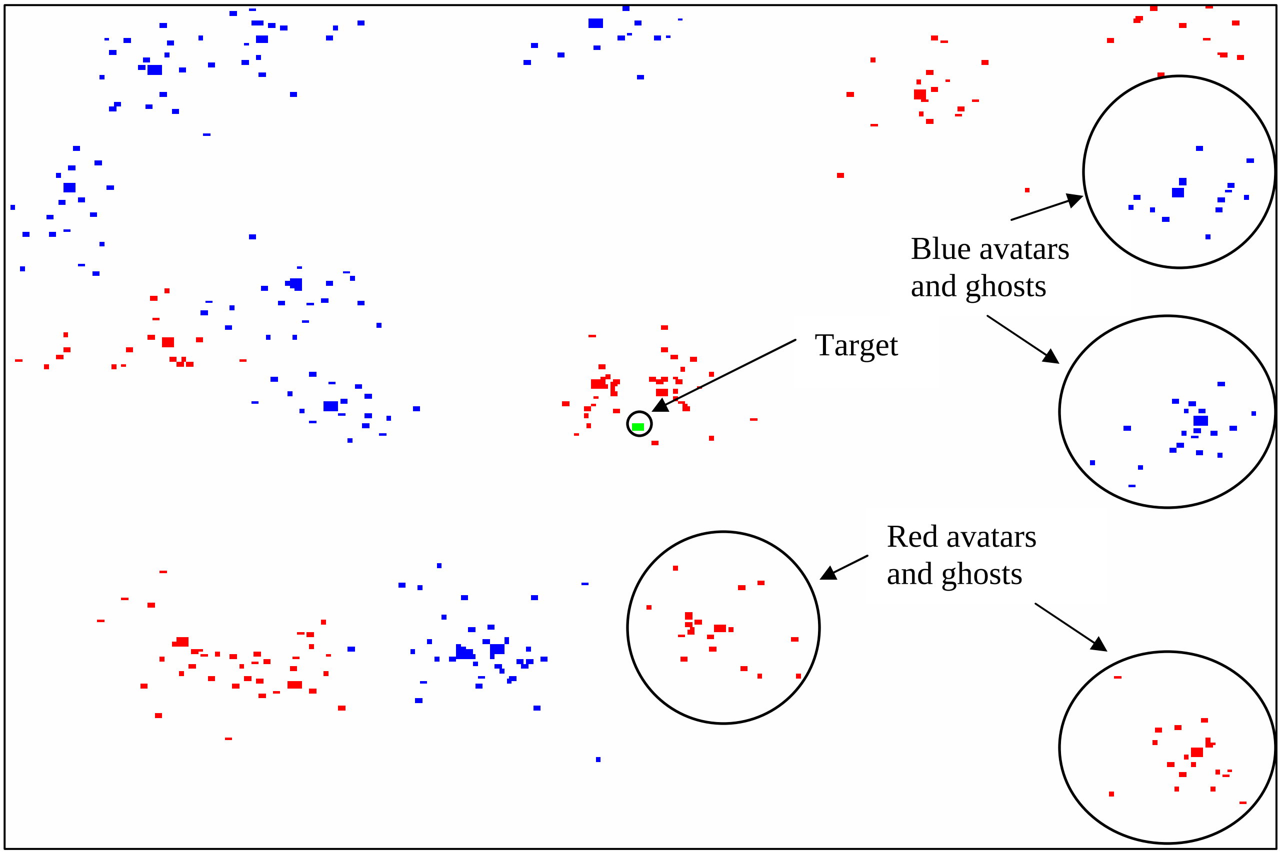

Figure 1: Snapshot of Polyagent Model

In Figure 1 above, the large blue dots represent Blue force avatars, with the small blue dot clusters around each one representing that avatar's ghosts. The corresponding relationship holds for the red dots and the Red forces. Green dots (in the center in the above figure) represent targets pursued by Red forces. Although the number of ghosts associated with each avatar is the same at the beginning of the ghost's forecast horizon within an avatar timestep, as ghosts "die" over the course of the forecast horizon, they disappear from the display as expected.

The modeling environment used to run the model and visualize simulations is a Java-based application. Configuration points in the application (described below) that modify agent behavior and modeling environment variables are set in editable XML files.

Initial Conditions

As a simulation run begins, a specified (configurable) number of targets and Red/Blue forces are placed on a grid. In our simulations, we explored both symmetrical (equal Red/Blue forces) and asymmetrical (greater Blue forces) scenarios. In addition, the polyagents and targets can be placed at specified locations on the grid, or at random locations. Our primary focus in the experiments we ran was to explore random initial placement of targets and forces. Random placement was chosen because we were more interested in "unpredictable" scenarios where the location of the enemy and the identity of the targets are not initially known, a situation closer to asymmetric conflicts and terrorism /counterterrorism scenarios than, say, traditional land combat.

Also, these scenarios present settings in which incremental or local planning is more heavily emphasized.

Polyagent behaviors and interactions

The goals of polyagents are relatively straightforward. As stated above, the Red polyagents both seek out the target(s), while avoiding Blue forces. Conversely, Blue polyagents are exclusively focused on seeking out and destroying Red forces. All polyagent behaviors are derived from the above motivations. Also note that the same objectives apply to both the avatars and their corresponding ghosts (though they have different decision-making procedures in pursuit of these goals, as outlined below).

The polyagents interact with opposing forces and the target through *pheromone* fields. Red forces, Blue forces, and the target all emit different "pheromone flavors" or types [7] that can be detected by other players, as follows:

- Green – Emitted by the target at a consistent rate, detectable by Red ghosts
- Blue – Emitted by Blue forces, indicating threat to Red forces
- Red – Emitted by Red forces, indicating threat to Blue forces

Note that all pheromones propagate and spread, while also evaporating over time to provide an overall "pattern of relevance" to the polyagents detecting them. The detailed functions for the pheromone pattern behavior, for both propagation and decay, can be found in [8].

The key decision that the polyagents must make during each cycle is determining which square in the grid to which to move. Each execution of the "move algorithm" (based on the polyagent's goals) results in a vector that determines which adjacent square the polyagent (avatar or ghost) transitions to. This calculation is designed to align with the polyagent's goals, as follows:

- Ghosts
    - Red ghost next move vector:
      Highest green pheromone concentration square – highest blue pheromone concentration square + weighted random factor[3]
    - Blue ghost next move vector:
      Highest red pheromone square + weighted random factor
- Avatars
    - Red avatar next move vector:
      Highest green pheromone square that any ghost the current avatar timestep encountered during its lifetime[4] – a vector is created by summing the components of ghost death locations + a weighted random factor

[3] Random X and Y factors are generated over the uniform interval [0,1] and then scaled so that their magnitudes are each at most equal to 10% of the non-random X and Y components.

[4] Ghosts and avatars "die" with a certain probability when they are co-located on a square, as described in more detail below.

- Blue avatars next move vector:
  A vector created by summing the components of ghost death locations during the current avatar timestep + a weighted random factor

The decision method implemented in this model is inherently stochastic. This represents the noise and randomness of the real world, and has the beneficial side effect of preventing agents from getting stuck in a corner or in a local pheromone optimum.

Finally, there is the question of what occurs when the polyagents directly encounter opposing forces or the target (i.e., they are co-located on a particular square in the grid). The outcomes in each of these cases are straightforward, and apply to both avatars and ghosts, as follows:

- Red and Blue on same square: Outcome is governed by a “kill probability” parameter that Red agent dies when encountering Blue, or vice-versa
- Red and Target on same square: Red destroys the target and “dies”; and a new target in a new location may then appear
- Blue and Target on same square: no change in state for Blue or Target
- If two or more polyagents (ghosts or avatars) of the same type (Blue or Red) are in the same square, this merely has the effect of increasing the amount of that flavor of pheromone in the square

A run of the simulation can continue running until all targets are eliminated or until all Red forces have been eliminated.

Summary

Informally, an interpretation of the structure and behavior of the two sides are as follows. Red’s behavior pattern utilizes relatively independently operating “cells” (since there is no direct interaction between avatars) that avoid the enemy, but perform a “suicide attack” when reaching the target. Blue is primarily interested in “taking the fight to the enemy”, without directly knowing exactly what the enemy is targeting. These conditions and behavior are similar to those found in many asymmetric warfare and terrorism / counterterrorism scenarios, in which the location of targets can be highly uncertain and the precise location and intentions of the adversary are primarily inferred indirectly and probabilistically.

Also note that the model employs incremental attack planning because the polyagents continually adjust to local conditions “on the ground”, never looking more than one step ahead at a time, with no central command and control globally guiding their behavior.

4. SIMULATION AND EXPERIMENTAL RESULTS

The purpose of this initial round of simulations was exploratory rather than confirmatory; we sought to demonstrate that the polyagent modeling environment could be applied to attack planning and gain some understanding of the critical variables driving the results. Subsequent experiments will test specific hypotheses about polyagent behavior.

Key parameters

Following are the key parameters of the polyagent model simulations in the present study (for a listing of all configurable polyagent behavior parameters, see Appendix X):

Parameter	Description	Range	Default
NA	Number of Avatars per side	Any non-negative number	<none>
NG	Number of Ghosts per Avatar	Any non-negative number	5 (for both Red and Blue[5])
KP	Kill Probability when encountering opposition (avatar or ghost)	Decimal between 0 and 1	0.9
FH	Forecast Horizon (number of cycles the ghosts "play ahead" before reporting back to their avatar	Integer between 0 and 25	<none>
WR	Weight of random factor in determining next move of polyagent	Decimal between 0 and 1	0.1
DG	Dimension of grid	Array	25 x 25
TR	Target Regeneration	Target reappears in the Same location, or in a New (randomly chosen) location	<none>

Table 1: Key Parameters in Polyagent Simulation

Metrics of Success

We measured success in a simulation run for the polyagents in terms of relatively simple objectives relative to the goals stated above. For Red, the objectives were to maximize the number of targets found and destroyed, and to maximize the number of its surviving avatars over time. For Blue, the objectives were to minimize the number of targets destroyed and to eliminate all Red avatars.

Experimental Results

We executed a variety of exploratory runs of the model, focusing on two key variables, the relative strength of force for the Red / Blue polyagents, and the forecast horizon, which can be viewed as the number of steps into the future that the avatar's ghosts "play ahead" to assess the likely outcome of different actions. For example, a forecast horizon of 10 means that the ghosts run up to 10 virtual time steps into the future. Relative strength of forces is an obvious variable

[5] Most behavioral parameters can be set differentially for Red and Blue, but during the initial round of simulations the parameters were set to be equal for both sides unless otherwise specified.

to focus on, whereas the forecast horizon variable is particularly interesting from the standpoint of exercising the core capability and potential of the ghost component of the polyagent.

To explore the relative force strength variable, we began by executing a variety of runs with equal numbers of various quantities (NA = 5, 10, and 15) and a fixed target. For all of these scenarios, we found that the Red forces easily hit the target, in most cases multiple times, before the Red forces are eliminated. This general result was consistent across changes in NA, NG, and FH.

Figure 2 shows the results of a typical sequence of ten runs with NA=10, NG=5, FH=5, and TR=Same. For the ten runs performed with these parameters, the average number of Red avatars reaching the target was 7.9 (out of 10 possible), with a standard deviation of 1.66. The total number of cycles needed to eliminate all Red avatars (either through reaching the target or being killed by Blue) was 399, with a standard deviation of 177.

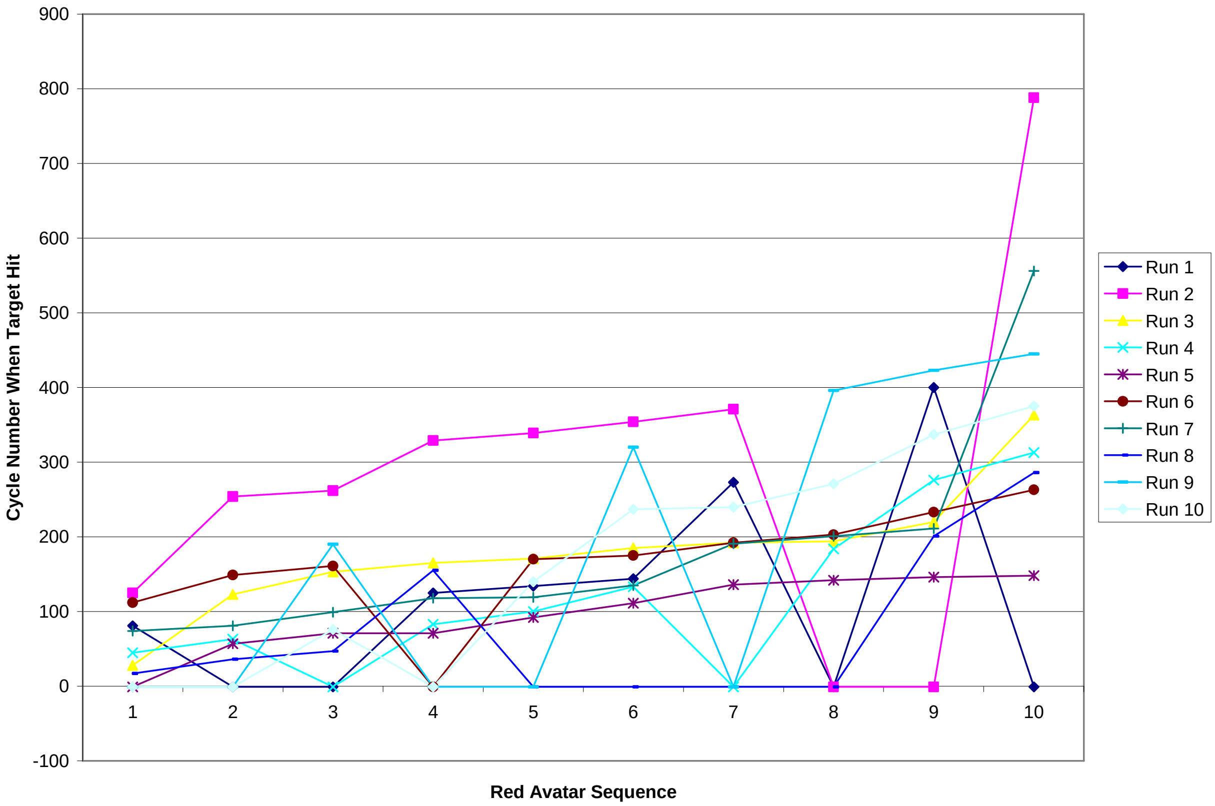

Figure 2: Sample Equal Force Polyagent Run

Each line in the above display represents the results of the Red force avatars over the life a particular run. Note that a point dropping to the zero line represents a Red avatar that was killed before reaching the target. So to take a specific example, in Run 2, the results of the avatars were as follows (Table 2):

Avatar	Result
1	Reached target (cycle 125)
2	Reached target (cycle 254)
3	Reached target (cycle 262)
4	Reached target (cycle 329)
5	Reached target (cycle 339)
6	Reached target (cycle 354)
7	Reached target (cycle 371)
8	Killed by Blue (cycle 434[6])
9	Killed by Blue (cycle 568)
10	Reached target (cycle 788)

Table 2: Run 2 Results

Our intuition about Red's high degree of success is that it arises primarily from the information asymmetry about the target: simply put, Red receives signals directly about the location of the target (through the Green pheromone flavor), whereas Blue does not. Another way of putting this is that Blue is perpetually in reactive mode, doing its best to respond to the presence of Red but not directly knowing what Red is targeting. Therefore, it will seldom be successful in denying all (or even most) of the Red avatars access to the target.

We next looked at scenarios in which Blue has a much larger force than Red (NA for Red = 5, NA for Blue = 25), while varying the forecast horizon FH. Specifically, we looked at FH = 0, 1-5, 10, and 15 and TR=New, while keeping other parameters constant at their default values. Figure 3 summarizes the results of the scenario runs (10 runs for each value of FH).

[6] Cycles in which avatars were killed are not shown on the graph.

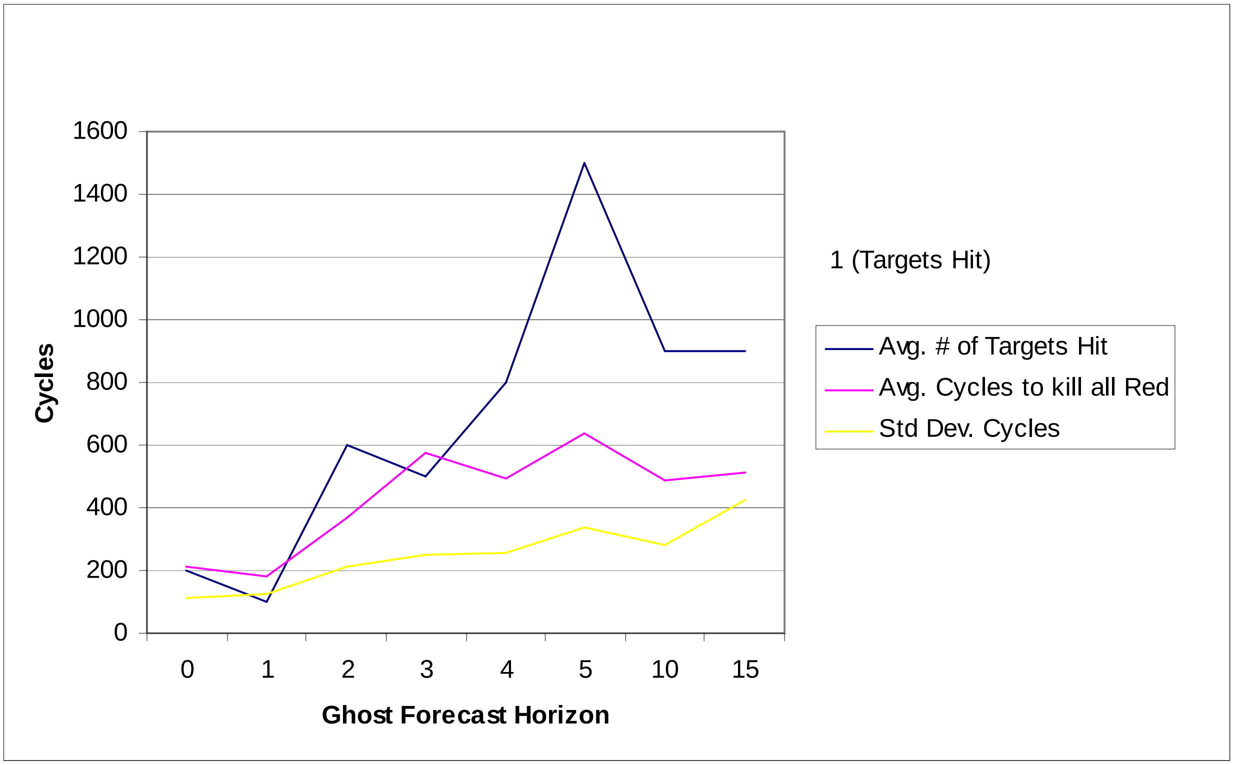

Figure 3: Polyagent Model Over Different FH Values

Table 3 shows the summary statistics for number of target hits and number of cycles to eliminate all Red avatars across each value of FH:

FH Value	Avg # of targets hit	Std Dev. of # of targets hit	Avg # of cycles to eliminate Red	Std Dev. # of cycles to eliminate Red
0	0.2	0.42	212	112
1	0.1	0.31	181	127
2	0.6	0.69	369	212
3	0.5	0.70	578	251
4	0.8	0.42	494	257
5	1.5	0.85	636	340
10	0.9	0.73	490	281
15	0.9	1.1	515	428

Table 3: Summary Statistics Across FH Values

In general, because of its asymmetric information advantage, Red was still successful at reaching at least one of the targets, even when vastly outnumbered. However, one of the more interesting results from this set of simulations is that Red success is an increasing function of FH, at least up to a point.

Specifically, Red became more and more successful as it looked up to 5 cycles ahead (even as the Blue forces did also), but Red's success dropped off when attempting to look further ahead than that. One interpretation of this finding is that as the forecast horizon increases, the ghosts are exploring increasingly unlikely scenarios, so the extra information being fed back to the avatar is of limited value or is even misleading. For at least the application and set of conditions, this helps us address of the question of “how much forecast information is enough”?

Looking at the average number of cycles needed to kill all Red avatars, note that this value also peaks at FH=5, and drops off significantly at larger forecast horizons. This suggests that looking ahead further helps Red “stay in the game” longer, up to a point. Again, our interpretation the drop-off in this value at FH > 5 is that looking ahead further at increasingly unlikely scenarios does not benefit the Red forces.

Finally, although the standard deviation of the number of targets hit and the number of cycles needed to kill all Red avatars generally trends up as FH increases, the rate of increase is lower and the data is “noisier” than the data for the averages. Therefore, it would be premature to suggest that this constitutes a significant relationship. Further investigation would be required to better understand this relationship.

5. CONCLUSIONS

This paper showed how the polyagent modeling construct can be used to implement a series of exploratory incremental attack planning scenarios. This was achieved through the use of the pheromone fields and “next move” algorithms reflecting the goals and motivations of the Red and Blue forces. Polyagent modeling provides a novel way of exploring a great variety of probabilistic scenarios for command and control in a computationally efficient fashion.

The initial simulations performed have suggested both the benefits of planning ahead in the modeled command and control scenarios as well as the limitations of attempting to plan ahead too far. For the particular parameters and assumptions embedded in the present model, the benefits of planning ahead peaked at a forecast horizon of 5.

More work is needed to understand the role of other variables in the polyagent model, and further refine it. As noted in [6], the application of the polyagent construct is presently more art than science, and further investigation will help us to better understand its mechanics and “tune” these types of models. Specifically, it would be interesting to run further simulations to better understand the potential influence of NG (number of ghosts), WR (the weight of the random factor) and KP (the kill probability) on the success of the Red and Blue forces. As well, we have not yet explored the impact of varying any of the parameters across Red and Blue, apart from number of avatars per side. Further applications of the forecast horizon FH could also be explored; for instance, the model could be modified to investigate how to set FH to best trigger when to call for reinforcements or change objectives. Finally, it would be useful to modify the model to enable the Blue forces to have some more explicit awareness of the target location, as this is clearly the case in many real-world security and counterterrorism settings.

Acknowledgments

The authors wish to thank Carl Hunt of the Institute for Defense Analyses for his insightful review of an earlier draft of this paper.

References

[1] Ilachinski, A., *Artificial War: Multiagent-Based Simulation of Combat*, World Scientific Publishing Company, 2004.

[2] Moffat, J., *Complexity Theory and Network Centric Warfare*, Command and Control Research Program Publications Series, Washington, D.C., 2003

[3] Liu, Q., and Xue, H., "Modeling Intelligence C2 Using Technology of Multi-Agent," 11th CCRTS, Coronado, CA, June 19-22, 2006.

[4] Ruan, S., Gokhale, S., and Pattipati, K., "An Agent-Based Model for Organizational Analysis, 11th CCRTS, Coronado, CA, June 19-22, 2006.

[5] Brueckner, S., and Parunak, V., "Resource-Aware Exploration of the Emergent Dynamics of Simulated Systems," AAMAS '03, July 2003, Melbourne, Australia.

[6] Parunak, V. and Brueckner, S. "Modeling Uncertain Domains with Polyagents", AAMAS '06, May 8-12, 2006 Hakodate, Hokkaido, Japan

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

[8] Parunak, V., Purcell, M., and O'Connell, R. "Digital Pheromones for Autonomous Coordination of Swarming UAV's", American Institute of Aeronautics and Astronautics, 2002

Appendix

Configuration Points in Polyagent Model

Type	Name	Description	Default
Cell Coordinates (Grid)	minLongitude, max Longitude	Defines size of grid	25 x 25
Cell Coordinates (Grid)	stepLongitude, stepLatitude	Defines units of grid	1
Pheromone Flavors	RedThreat	Strength of Red pheromone	0.9
Pheromone Flavors	BlueThreat	Strength of Blue pheromone	0.9
Pheromone Flavors	GreenThreat	Strength of Blue pheromone	0.9
Agents	maxGhostForecastHorizon	Maximum Forecast Horizon that can be set in the model	25
Agents	ghostForecastHorizon	Forecast Horizon (FH)	<None>
Agents	maxRandomWalkFraction	The weight of the random factor in the next move vector (WR)	0.1
Agents	avoidDeathThreat	Does the polyagent avoid threats from the opposition?	True for Red, False for Blue
Agents	killProbabilityByThreatEncounter	Kill probability (KP)	0.9
Agents	InsertionDataConfig (count)	Number of avatars (NA)	<None>
Agents	ghostsPerTimeSlice	NG	<None>
Agents	recreateAfterDeath, ignoreInsertionCoordinates	TR	<None>
Agents	StepLength	Max distance agents can travel per cycle	0.5