Coevolving Communicative Behavior in a Linear Pursuer-Evader Game

Sevan G. Ficici, Jordan B. Pollack
DEMO Lab
Computer Science Department
Volen National Center for Complex Systems
Brandeis University
Waltham, MA
http://www.demo.cs.brandeis.edu

Abstract

The pursuer-evader (PE) game is recognized as an important domain in which to study the coevolution of robust adaptive behavior and protean behavior (Miller and Cliff, 1994). Nevertheless, the potential of the game is largely unrealized due to methodological hurdles in coevolutionary simulation raised by PE; versions of the game that have optimal solutions (Isaacs, 1965) are closed-ended, while other formulations are opaque with respect to their solution space, for the lack of a rigorous metric of agent behavior. This inability to characterize behavior, in turn, obfuscates coevolutionary dynamics. We present a new formulation of PE that affords a rigorous measure of agent behavior and system dynamics. The game is moved from the two-dimensional plane to the one-dimensional bit-string; at each time step, the evader generates a bit that the pursuer must simultaneously predict. Because behavior is expressed as a time series, we can employ information theory to provide quantitative analysis of agent activity. Further, this version of PE opens vistas onto the communicative component of pursuit and evasion behavior, providing an open-ended serial communications channel and an open world (via coevolution). Results show that subtle changes to our game determine whether it is open-ended, and profoundly affect the viability of arms-race dynamics.

1. Introduction

The pursuer-evader (PE) game is argued to be an important domain for coevolutionary simulation (Miller and Cliff, 1994) due not only to its ubiquity in nature, but also because it provides a parsimonious, yet powerful, framework of investigation: robust adaptive behavior, open-ended coevolution, and adaptively unpredictable, or *protean*, behavior all fall within its purview. In particular, the PE game is believed to be rich enough to support the evolution of sophisticated perception, control, and predictive ability; insight gained by evolving agents for the game may impact disciplines in the biological and ethological communities. Nevertheless, the PE game also presents methodological hurdles that highlight shortcomings in the state of the art of coevolutionary simulation itself. While some of these hurdles have diminished, most notably the *red-queen effect* (Cliff and Miller, 1995), many others remain that demand a more lucid understanding of coevolutionary learning. Key issues involve the conditions that instigate *competitive arms races* or, to the contrary, *mediocre stable-states* (MSS) (Angeline and Pollack, 1994; Pollack et al., 1997), neither of which are well understood.

We posit that the spatially-oriented PE game brings at least three difficulties to coevolutionary research: 1) the lack of a canonical form, which leads to 2) the question of open-endedness, and 3) the lack of a rigorous metric of agent behavior. First, there exist many versions of the PE game. The earliest formulations (Isaacs, 1965) are games of perfect information where optimal pursuit and evasion strategies follow analytically from agent capabilities. More recent work casts PE into a purely discrete, non-kinematic reactive game (Koza, 1992; Reynolds, 1994), or into a continuous game of, essentially, imperfect information that incorporates a two-dimensional physical model (Cliff and Miller, 1996), for example. Second, these differences in game formulation impact the nature of the solution space; where PE formulations have optimal solutions, the game is closed-ended and, therefore, categorically less interesting than an open-ended domain. Further, these optimal solutions are not compelling examples of evolved complexity; a formulation where optimal pursuit is defined simply by reactively moving directly towards the evader does not even require the pursuer to hold state. Third, and finally, agent behavior in PE is currently described exclusively in subjective, qualitative terms; as (Cliff and Miller, 1996) point out, there does not yet exist a quantitative, operational measure by which one can characterize and judge agent behavior.

The hypothesized potential of the PE game is thus largely unrealized due to the opaqueness of the domain, of coevolutionary dynamics, and their interaction; we present a reformulation of the pursuer-evader game that helps clarify all three. Simply, we move the arena of action from the two-dimensional plane to the one-dimensional bitstring; at each time step, the evader generates a binary symbol that the pursuer must simultaneously predict. Games last one thousand time steps. A tally of correct guesses is used to derive players' scores. Because behavior is expressed as a binary time series we can take advantage of information theory to provide precise characterizations of agent behavior and evolutionary progress.

As importantly, recasting the PE game in this way opens vistas onto the *communicative* component of pursuit and evasion behavior. Recent work by (Di Paolo, 1997) raises a concern that much research on communication conflates functional utility with operational process: the utility of communicative behavior may very well be that it confers adaptiveness, yet communication so defined fails utterly to describe the behavior itself. Similarly, the denotative function of 'information' is often conflated with the process that generates it. In our PE game we can characterize pursuit and evasion behavior outside of the context of selective advantage. Further, we operationalize the behaviors and unify the notion of information with the structure of behavior. Effective pursuit requires behavioral coordination with the evader. Effective coordination requires the induction of the evader's behavioral dynamics by the pursuer. This process of induction represents our notion of communication.

Nevertheless, "communication" obviously encompasses a much broader range of phenomena than pursuit and evasion — presently too broad to be captured entirely within a single experimental framework. A coarse taxonomy might divide much research to-date into two classes: *convergent* systems and *competitive* systems. Examples of the former include work that focuses on establishing common conventions of behavior. In (Werner and Dyer, 1991), female and male agents converge onto a signaling convention that facilitates reproduction. Each utterance is fed from female to male via a three-bit parallel channel; in the most adaptive pairs, the utterance, produced by a "sighted" but stationary female, causes the "blind" male to move in one of four directions such that it draws nearer to the female.

Other work builds a self-organizing system that converges onto a convention of word/meaning associations (Steels, 1997). The universe of discourse is the set of agents themselves, each of which is uniquely defined by a finite set of meanings, or feature/value pairs. Experiments described have on the order of twenty agents, five meanings, and five words. The agents build lexicons that associate the finite set of words to the feature/value pairs and use word groups to signify particular agents. This domain is claimed to be open in the sense that it can assimilate new meanings (features or values) and agents at any time, even after a convention has emerged. Nevertheless, the semantic and denotative material used to establish a communicative convention is supplied externally from the system, not created by it; any convention that emerges is merely a reflection of a pre-existing, hand-built world.

In the second category of the taxonomy, we have systems that eschew convention; competitive games are designed to promote ever increasing "complexity," the precise meaning of which is inevitably domain-dependent. One game of imitation asks agents to observe another's behavior for some time and then reproduce that behavior (Kaneko and Suzuki, 1994); the observation and reproduction phases of the game last 255 and 32 times steps, respectively. Agents are coevolved to become better imitators and more difficult to imitate. Though behavior does become more complex, the primary cause of complexity likely derives from inductive bias in the substrate: agent behavior is synonymous with the logistic map. The evolutionary algorithm performs a one-dimensional search in the space of the logistic map's constant, which entirely determines agent behavior. While this system reveals what the adaptive advantages of various behaviors are according to the rules of the game, the substrate is so impoverished that complexity of behavior bears no meaningful relationship to agent complexity or evolutionary search.

A comparable situation is found in a system where grammars evolve to climb the Chomsky hierarchy through a competitive game of string generation and parsing (Hashimoto and Ikegami, 1996). The substrate in this case consists of explicit production rules, however, and the mutation operators allow the system to traverse the hierarchy easily. Indeed, the complexity of the evolved grammars is not determined by observing agents' behaviors (word length is limited to six symbols), but rather by looking directly "under the hood" at their production rules.

The above examples are emblematic of issues common to a great deal of work on communication, namely: powerful representations, small-scale communicative (behavioral) acts, fixed channels, and a fixed world. Viewed as an experiment in communication, our modified PE game provides 1) an open-ended serial communications channel (Gregory M. Saunders, 1996), which can lead to 2) substantial communicative acts, and 3) an open world (via coevolution). Too, information theory provides a rigorous and accepted vocabulary for describing signal properties. Further, the substrate used in our experiments is a recurrent artificial neural network; though powerful, the substrate is non-specific to our domain, and thus "weak." Finally, as described below, we seek to bridge the gap between convergent and competitive systems and explore the tension thus created.

Results show our domain to be an effective instrument for research in coevolution, pursuer-evader dynamics, and communication. Our methodology is built around the analytical tools made available by information theory. We are able to analyze agents that evolve and use hand-built agents of known ability to influence evolution. By characterizing generator behavior, we are able to identify variations of the game that lead to mediocre stable-states or define optimal solutions. These results lead us to a variation of our game that avoids both.

This paper is organized as follows: Section 2 details our version of the PE game and contrasts it with the more traditional spatial formulations; Section 3 describes the recurrent artificial neural network substrate used to represent our agents, and introduces the evolutionary algorithm; Section 4 defines concepts from information theory that provide our metric of agent behavior and coevolutionary progress; Section 5 places these concepts within the framework of a coevolutionary arms-race; Section 6 analyzes our results; Section 7 summarizes our work and points to future directions for our research.

2. Game Setup

2.1 Two-Player Pursuer/Evader

Illustrated in Figure 1, our basic PE game consists of two agents that play in discrete time and space. The *generator* (evader) is an agent that ballistically produces a binary time series, that is, its behavior is determined solely by its own internal dynamics; at each time step, the generator simply outputs a new bit. The *predictor* (pursuer) is an agent that, given its own internal state along with the generator's output from the previous time step as input, simultaneously tries to predict the generator's output for the current time step. Each match has a duration of one thousand time steps. Thus, our generator corresponds to a "blind" spatial evader, as its behavior is not modulated by that of its opponent. This modification "clamps" one side of the pursuer/evader dynamical system; indeed, a "sighted" evader would simply be a negated pursuer — an agent that predicts its opponent in order to perform the opposite action.

In addition, our game does not define a "capture" predicate; agent performance is measured strictly in terms of the number of correct and incorrect predictions made, which is analogous to measuring time-in-contact between pursuer and evader over the course of a spatial match. Too, we discover that, due to its discrete behavior, our generator corresponds to a particularly nimble spatial evader, as its current "location" in no way constrains where it may be in the next time step; we have at best a very rudimentary kinematic model.

Our variant game may thus appear quite foreign. Indeed, it resembles the *penny matching* game, a version of which is used in a non-predictive context in parasite/host gene-matching simulations (Hamilton et al., 1990). Nevertheless, we contend that our modifications distill the traditional PE game to its *informational* essence: given such an agile evader, the (equally agile) pursuer must induce, from observation of behavior, a model of the evader in order to be effective. The quality of the model is demonstrated through the pursuer's prediction abilities. This game, therefore, highlights the need to evolve perceptual, computational, and "motor" abilities in the pursuer. The search for effective strategies conducted by the simulated evolution traverses a wide range of such abilities; thus, while our game is objectively one of *perfect information* for the pursuer — it has the opportunity to know the evader's actions — pursuers are not necessarily endowed to make use of this knowledge. In effect, the game easily admits the possibility of information loss, albeit within the evolutionary substrate. In contrast, our game is invariably one of *imperfect information* for the evader; only through coevolutionary feedback do the evaders receive any information.

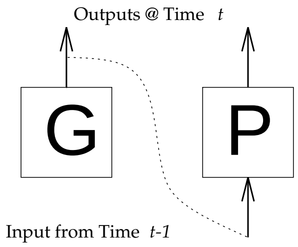

Figure 1 Basic Game Setup. At each time step, the predictor receives the output of the ballistic generator from the previous time step and outputs a prediction of the generator's behavior for the current time step.

2.2 Three-Player Pursuer/Evader

To synthesize convergent and competitive system dynamics, we introduce a third player into our game — a "friendly" predictor. We imagine the evaders to be individuals literally under constant predation who must, therefore, mate while "on the run." But, the mate must be able to keep up with the evader. The *three-player PE* game now consists of one ballistic generator that attempts to behave in a manner that is both predictable to the partner (friendly predictor), to allow mating, and simultaneously unpredictable to the pursuer (hostile predictor), to allow evasion.

This modification to the game is far from capricious. Prediction is intrinsically more difficult than generation; if we believe the Machine Learning aphorism that "you can only learn what you almost already know," then we must be mindful that generators might become too complex for predictors to learn against, breaking the coevolutionary arms race. Worse, generators might become inherently unpredictable. The friendly partner serves to dampen any such tendency by forcing generators to be predictable to someone. In terms of communication, this arrangement can be viewed as a competitive game between the two agent populations (generators and partners) trying to evolve a proprietary behavioral convention, and the third agent population (pursuers) trying to crack the convention.

For each generation of evolution, all generators are played against all partner and pursuer predictors. Scores across all games are averaged to derive fitness values. Game scores for all agents range between [0, 1]. The exact formulas used for scoring predictors are discussed below in the experiment descriptions. A generator's score is computed by subtracting the average scored by its pursuers from the average scored by its partners and normalizing the result to fall within the range [0, 1]; values above 0.5 thus indicate that a generator is able to make itself more predictable to partners than pursuers.

3. Substrate and Evolutionary Algorithm

Our agent substrate is an enhanced version of the discrete-time, deterministic recurrent artificial neural network used in the *GNARL* system (Angeline et al., 1994); the network enhancement consists of a set of nine new transfer functions, *min*, *max*, *sum*, *product*, *sigprod* (sigmoid applied to product), *unit-time-delay*, *sign*, *uni-linear-truncate* (truncate outside range [-1, 1]), *dec-linear-truncate* (truncate outside range [-10, 10]), in addition to the traditional sigmoid function. These supplementary transfer functions increase the range of behavior significantly, though by no means do they guarantee agent success. Unfortunately, these new functions also make network analysis much more difficult.

GNARL coevolves the networks' internal architectures and weights. Generators may have up to 60 hidden nodes and 400 weights, while predictors are given more generous limits of 150 hidden nodes and 700 weights due to the relative difficulty of their task. These limits are not known to be optimal and are, in fact, never reached.

The input and output layers of the networks are fixed. All networks have a single, real-valued output that is thresholded to a binary value. Though the game formally defines friendly and hostile predictors to have a single input, we currently provide both predictor roles with a small buffer to enhance performance: the predictors have five binary-valued inputs, corresponding to the last five outputs of the generator at times $t-1, ..., t-5$. This enhancement in no way obviates the need for a recurrent network architecture: predictors must still induce generator behavior by observation over time.

The GNARL algorithm performs its search solely through mutation — crossover is not used. Five mutation operators are implemented: *change-weights*, *add-hidden-nodes*, *remove-hidden-nodes*, *add-weights* (connections), and *remove-weights* (connections). When a hidden node is removed, all efferent and afferent connections from and to that node are removed as well. New nodes are added with no connections. Only a single mutation operator is applied to a network when it is modified. Network weights are modified by adding a Gaussian to each weight. The overall severity of mutation performed to a network is determined by its *temperature*, which is an inverse function of its fitness. The higher the temperature of a network, the more severe the mutation will be.

Three distinct agent populations of size 75 are maintained, one each for the evaders, friendly partners, and hostile pursuers. In each generation of coevolution, all evaders are matched with all friendly and hostile predictors. After evaluation, approximately 75% of the worse half of each population is replaced by mutated versions of agents in the better half.

4. Metric of Behavior

Two key notions from the field of information theory provide our game with a rigorous and quantitative metric of agent and system behavior, namely *entropy* and *order*. Rather than give their formal mathematical definitions, we emphasize a more intuitive explanation of these concepts and their implications as they relate to our domain. Formal detail can be found in (Hamming, 1980).

4.1 Entropy

Information theory is concerned with characterizing signals and their transmission. A signal *source* produces some symbol, which is passed through a *channel* to a *receiver*. We assume, for our purposes, that the channel does not distort the signal. The *entropy*, h, of a source reflects the receiver's uncertainty as to what it will receive. The higher the entropy, the less certain the receiver is, and the more it learns once the symbol is actually received. Thus, entropy is a measure of the amount of *information* in a signal. More precisely, the entropy of a source is equal to the average number of bits of information produced (conveyed) per generated symbol.

By indicating the uncertainty of the receiver, entropy inversely indicates the degree to which the source can be *predicted* by the receiver, that is, the receiver's certainty. We must be careful to point out that the receiver's opinion of what the next symbol will be is based exclusively upon the *observed behavior* of the source — assumptions about the source's internal operation are not made.

4.2 Order

If the receiver's certainty is based upon observation of the source, we can ask "How much observation is required to maximize the receiver's certainty?" For example, let us consider some binary source, $\mathcal{S}$. If the receiver only tallies the number of occurrences of *0* and *1*, this source may be found to produce each 50% of the time. With this amount of behavioral context, the receiver's certainty of the next symbol is zero and entropy is measured at $h = 1.0$. Nevertheless, it may be that if the receiver keeps track of the previous symbol received, then the source will be found simply to be alternating between *0* and *1*; in this case, a behavioral context of one symbol makes the source completely predictable. Measured entropy is now $h = 0.0$. If the receiver keeps track of yet another symbol, now the previous two, no additional advantage is gained with respect to the source, $\mathcal{S}$.

The minimal amount of behavioral context needed to maximize the receiver's certainty of a source is the *order* of the source. The order is equal to the number of symbols that must be tracked, that is, the size of the *history window* needed to maximize receiver certainty. The entropy measured when using a window size equal to a source's order is the *true entropy* of the source; window sizes larger than a source's order will produce measurements equal to the source's true entropy, but not lower. Thus, a receiver cannot increase its certainty of a source by using a window size larger than the source's order.

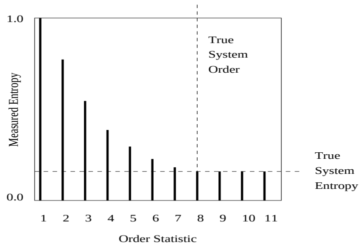

Figure 2 Measured vs. True System Entropy and Order.

4.3 Order Statistics and Measured Entropy

With our example source, $\mathcal{S}$, above, we first measured entropy without keeping track of the previously generated symbol; this is equivalent to measuring entropy with a window size of zero, or measuring with zero-order statistics. Our second measurement, then, used a window size of one, or first-order statistics. Our zero-order measurement gave us an entropy of $h = 1.0$, but the first-order measurement fell to the true entropy of $h = 0.0$. Indeed, measured entropy will always monotonically decrease as window size is increased, and eventually reach a source's true entropy, as illustrated in Figure 2.

A source with true entropy $h = 0.0$, such as $\mathcal{S}$, is completely predictable and regular. In contrast, a binary source with maximal true entropy of $h = 1.0$ entirely lacks structural regularity and cannot be predicted better than random, on average, without specific knowledge of its internal works. For a source with true entropy somewhere in between, $0.0 < h < 1.0$, there exists both a *regular component* and an *irregular component* to the source's signal. The regular component is that portion of the signal that can be reliably predicted, while the irregular component is that portion that cannot. By definition, the information content of a source must reside exclusively in the irregular component.

4.4 Complexity

System order is also equal to the logarithm of the maximal number of states required for a Markov model to reproduce behavior statistically identical to a source; entropy reflects the degree of certainty in the model's state transitions. Consider a randomly behaving binary source, with true entropy of $h = 1.0$. We find that the minimal window size needed to maximize a receiver's certainty of this source is *zero*. Since the order of such a source is zero, the equivalent Markov model requires $2^0 = 1$ state to reproduce statistically identical behavior. This result is understandable since there exists no signal structure to capture through state. Thus a random source is considered to be simpler than a completely predictable source of higher order; the size and structure of the Markov model is what counts, not the compressibility of the produced signal. This view is substantially similar to the notion of *statistical complexity* found in (Crutchfield, 1994).

5. Where's the Arms Race?

When considered together, order and entropy form the nexus between generator complexity and predictor power: if a signal has a regular component, then that component can be predicted assuming that the power of the predictor is sufficient; that is, the predictor must use an order statistic, i.e., history window, of size $m \geq n$, where n is the order of the signal being predicted. If the predictor's window size is m, such that $m < n$, then it will be able to predict only that portion of the signal's regular component that is detectable when measuring the signal's entropy with m^{th}-order statistics. Recall that as window size decreases, measured entropy increases; thus, predictors using smaller windows will necessarily be weaker than those using larger windows.

Because irregular signal components are inherently unpredictable, and our three-player game requires generators to be predictable to friendly predictors, generators must maintain substantial regular components in order to succeed. Nevertheless, generators need to be unpredictable to the hostile predictors. The only way both goals can be effectively met is for the generators and friendly predictors to evolve system order and predictive power that are closely matched, yet greater than the predictive power of the hostile predictors.

Regular signals allow for a general solution to the prediction task. A predictor of power n can predict any generator of order $m \leq n$; to escape prediction, therefore, a generator has no choice but to increase its order above n. The amount by which the generator increases its order and the unpredictability it exhibits at lower order-statistics determines how much the predictor's performance degrades. Of course, a generator may increase its true entropy instead; doing so, however, will also defeat any hopes of being predicted by the friendly predictor.

The assumption up to now has been that predictors will actually evolve such a general prediction algorithm. Of course, this represents an idealized solution; in reality, the issue of generalization vs. specialization is intimately tied to that of population diversity and domain "physics." Nevertheless, having some notion of what idealized predictors can and cannot do, and what an idealized arms race looks like, provides a useful framework in which to examine empirical results.

6. Results

6.1 Adaptive Pursuit Behavior

Our first experiments test the nominal capabilities of our recurrent artificial neural networks with respect to our problem domain. Due to the nature of our game, we can easily *hand-build* predictors of known power through a simple modification of the algorithm used to compute entropy. We thus craft an environment of two predictors of unequal power and evolve generators to become maximally unpredictable to the weaker predictor while remaining maximally predictable to the more powerful one. The generators that evolve successfully fill the niche between the two predictors. We use a variety of such hand-built predictors to evolve a set of 70 generators that range mostly over system-orders two through eight.

We then evolve predictors against this fixed set of generators. The operation of one such predictor is shown in Figure 3. The figure depicts the operation of our predictor over 800 time steps. During these 800 time steps, the predictor consecutively plays against eight different generators (against which it evolved), each for one hundred time steps. The generator's behavior is shown directly above the predictor's in each match. Each generator network has its activation levels reset to zero at the beginning of its match. In contrast, the activation levels of the predictor network is reset only once, at the beginning of the first match at time-step zero; thus, from the predictor's point of view, the generator changes its behavior every one hundred time steps.

During evolution, however, predictors *do* have their activation levels reset at the beginning of each match. Therefore, predictors are not explicitly evolved to adapt to such changes in generator behavior. Nevertheless, Figure 3 shows very compelling evidence of adaptive "pursuit" behavior against these generators; the sequence in which the generators are met does not affect the predictor's performance. Of special interest is the entrainment behavior exhibited by the predictor at the beginning of each match; in all matches (except the first), we see the predictor continue its pursuit behavior from the previous match for some period of time before it begins to entrain to the new generator's behavior.

From Figure 3, we see that the predictor is always "paying attention" to the generator, that is, the predictor does not settle into a single, immutable pattern of behavior. The predictor shows sensitivity to the frequency, phase, and duty-cycle of the first six generators. Further, generators 7 and 8 show that the predictor is capable of inducing more complex, "compound" signals, though generator 7 is not perfectly induced. In summary,

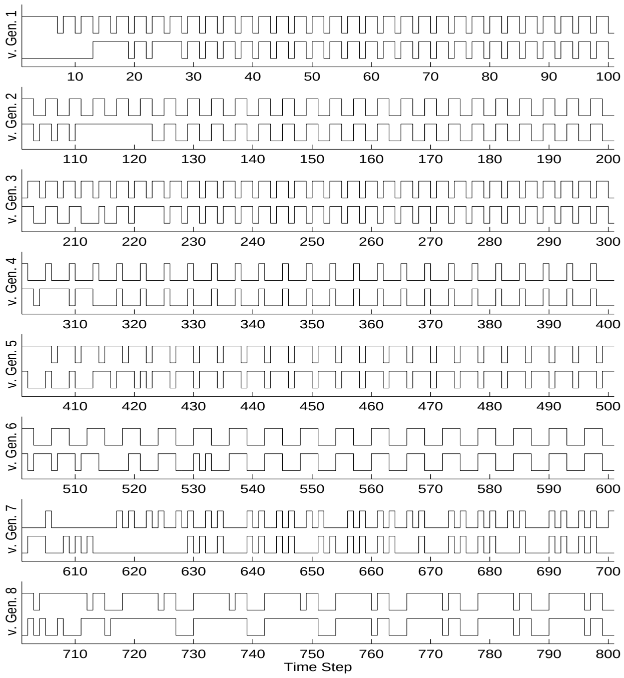

Figure 3 Adaptive Predictor Behavior. An evolved predictor consecutively plays against eight different generators, each for 100 time steps. Generator behavior is shown directly above the predictor's in each match.

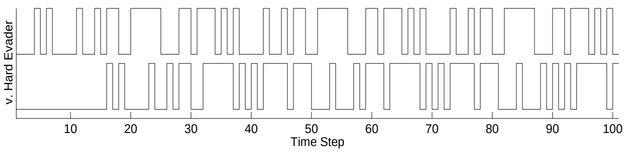

Figure 4 Predictor vs. Hard Generator. The evolved predictor is unable to entrain to this 5th-order generator.

these initial tests suggest a space of network behaviors rich enough to support coevolution.

6.2 Complexity and Computational Demand

Because the generators described above are ballistic and evolved to be predictable to someone, they settle quickly into regular (periodic) patterns of behavior. Though all regular generators have true entropies of zero, they can make very different demands of a predictor. For example, the generators in Figure 3 all have a true entropy of zero, but are of orders 2, 2, 2, 3, 3, 3, 13, and 6 (from top to bottom). The predictor in Figure 3 does very well against these generators. Yet, Figure 4 shows our predictor to perform no better than random guessing against a particular 5th-order generator. Since order indicates the minimal number of symbols that must be tracked to maximize prediction, this 5th-order generator should be easier to predict than either the 6th or 13th-order generator, against which the predictor does well.

When we measure the entropies of these three generators with window sizes zero through thirteen, we see why

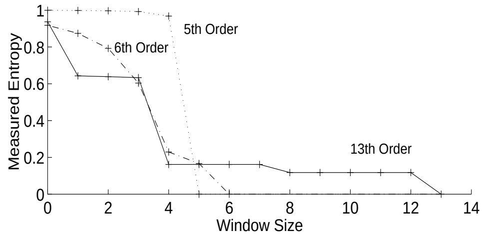

Figure 5 Entropy Curves for the 5th, 6th, and 13th-Order Generators. The X axis represents the window size used to measure entropy. The Y axis represents the measured entropy.

neither entropy nor order tells the complete story alone. Figure 5 graphs the *entropy contours* that are produced by measuring entropy with different window sizes. The solid line is the entropy contour for the 13th-order generator; we see that this generator is already substantially predictable with a much smaller window of four, but requires many more symbols to be tracked to be completely predictable. The dashed line is the contour for the 6th-order generator; this and the 13th-order generator are almost equally predictable with a window of size five. Yet, to be fully predictable, only one more symbol need be tracked for the 6th-order generator. The dotted line shows the entropy contour for the hard 5th-order generator; this generator's behavior appears random until the window size reaches five, whereupon it is completely predictable. That is, for *all* length-four subsequences of this generator's output, the next symbol could be *0* as easily as *1*; all 32 possible length-five subsequences thus appear in the output. For each length-five subsequence, however, the next symbol is uniquely determined.

Thus, while the 5th-order generator requires the fewest symbols to be tracked to be fully predictable, it also has the most contingencies to be resolved; representing and resolving these contingencies is computation that a generalized predictor must perform. All of the 32 possible five-bit patterns must be tracked to know unambiguously what the following bit will be. In contrast, the 13th-order generator requires only 17 of the 8192 possible 13-bit patterns (0.2%) to be tracked to determine subsequent bits.

The number of unique n-bit patterns found in an n^{th}-order periodic sequence is equal to the period of the sequence. Thus, the 5th-order generator of Figure 4 has a period of 32, whereas the 13th-order generator of Figure 3 has a period of 17. Further, the period of a sequence indicates the number of functionally distinct states that a predictor must maintain in its steady-state behavior (i.e., after entrainment). Thus, we can say that the 5th-order generator is more difficult to predict simply because it requires more states. Nevertheless, the steady-state demands placed on a predictor tell us nothing about the computational "effort" the predictor spends to arrive at its steady-state; the measure of period does not allow us to distinguish between the $\Theta(2^n)$ distinct period-n sequences. A periodic sequence of 31 ones followed by a zero, a sequence of 16 ones followed by 16 zeros, and the sequence seen in Figure 4 do *not* place identical demands upon a predictor and are not equally difficult to evolve against. Information theory thus gives us a way to more effectively characterize "hardness." This ability is invaluable when investigating coevolutionary learning.

6.3 "Boilerplate" Strategies

The generality of a predictor strongly reflects the diversity of the generators against which it evolved. For example, when we evolve predictors exclusively against 4th-order and 5th-order generators, we find the predictors to be rather specialized; the ability to predict these generators does not necessarily confer an ability to predict lower-order behaviors. Indeed, despite its success, the predictor in Figure 3 performs poorly against a simple, period-eight square-wave behavior.

If we evolve predictors against generators of various long periods, but of near 50% duty cycle, we find that a simple *copier* prediction strategy becomes tenable — one that simply outputs what it saw in the previous time step; such a strategy only misses on the generator's transitions between zero and one and "predicts" the rest (majority) of the sequence correctly. Other copier variants seen appear tuned to particular periods, such that any pattern of the appropriate period will be matched after one cycle. Such mediocre solutions occur easily where (a lack of) generator diversity allows.

6.4 Mediocrity and Protean Behavior

We begin our analysis of *co*evolutionary results by questioning the earlier-stated need for a friendly partner predictor. We describe two versions of the two-player PE game in a coevolutionary setting; the friendly partners are omitted. The versions are differentiated solely by the scoring method used to reward the pursuer predictors. The first scoring method (A) gives predictors a point for each correct prediction. The tallies are then normalized to the percentage of correct predictions. A generator thus receives maximal reward when a predictor is unable to make any correct predictions. The second scoring method (B) tallies correct predictions, like method A, but predictors are now rewarded only to the extent that they perform better than random; predictors that get less than or equal to 50% correct prediction receive a score of zero. In this case, a generator receives maximal reward for causing a predictor to perform no better than random. In both cases, the stage appears set for an arms race. We find, however, that the scoring methods have important repercussions in coevolutionary dynamics.

We gauge generator behavior by again taking entropy measurements over a range of window sizes to get entropy contours. Figures 6 and 7 graph the contours for the best generators that evolve over evolutionary time in sample runs for scoring methods A and B, respectively. Both graphs show that generator behavior remains consistent over the course of the run; we have no indication of an arms race. The generators coevolved with method A are substantially predictable and regular, as evidenced by the rapidly dropping contours. The generators coevolved with method B, on the other hand, are considerably more irregular (unpredictable), as the contours decline gradually and consume a much greater volume of space in the graph. This contrast in outcome is found even in pairs of runs that start with identical initial populations.

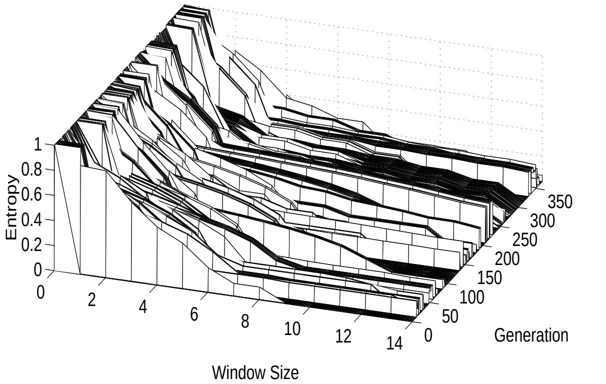

Figure 6 Scoring Method A — Reward for Each Correct Prediction. Rapidly declining entropy contours indicate substantially predictable behavior.

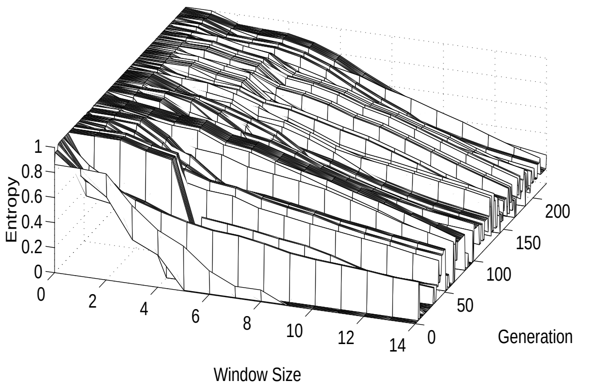

Figure 7 Scoring Method B — Reward for Predicting Better than Random. Gradually declining entropy contours indicate significantly more complex behavior than that seen by scoring method A.

The difference between scoring methods is manifested quickly, if not immediately. Since the initial conditions are similar (or, indeed, identical if we wish) we know that we have isolated a selection pressure that stems simply from how games are scored: regular behavior is more adaptive to scoring method A than is irregular behavior. That regular generators out-score irregular ones by method A indicates that simple prediction strategies can be elicited from the predictor population — prediction strategies against which the most adaptive generators act as potent *anti-signals*, sequences that make predictors perform worse than random (Zhu and Kinzel, 1997). Indeed, the champion generators shown in Figure 6 receive scores that indicate exactly this. Nevertheless, if the ballistic generators are to be viable as anti-signals, then there must exist some sufficient amount of homogeneity within the predictor population. This homogeneity appears to reverberate through the system; rather than enter an arms race, the two populations fall into a circular pattern of mutual specialization, or *convention chasing* — a mediocre stable-state.

In contrast, scoring method B selects for an irregular behavior that reasonably captures the notion of proteanism in the PE game. Figure 8 provides a sample of behavior from a typical generator produced by this run. This behavior is adaptive because it guarantees a minimal predictor score, regardless of what the predictor does, short of specialized memorization. Indeed, while the generators in Figure 7 are not actually fully irregular, they are complex enough to cause the best predictors from this run to score only an average of 6% better than random guessing (53% correct prediction); that is, these generators are near optimal with respect to their predictor opponents. By defining an optimal strategy, namely random behavior, method B makes the game closed-ended.

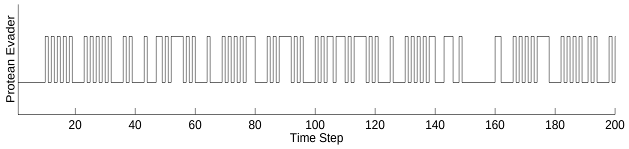

Figure 8 Typical Behavior of a Protean Generator.

6.5 Pursuing a Communicative Convention

The above results suggest that the two-player PE formulation may be problematic if we wish to develop an open-ended arms race. Particularly, the two-player game may oversimplify the ecology by inadequately placing constraints upon generator behavior. In the three-player PE game, we maintain scoring method B to promote complexity and simply add the friendly predictor partner. Now, the evader and friendly partner must establish a convention of behavior that allows the partner to entrain to (coordinate with) the evader, but not the hostile pursuer. We find that the tension created by the need to "mate" with (be predictable to) the partner profoundly affects evasion behavior. Far from extinguishing heated competition, the triangle of relationships gives rise to much more lively system dynamics than those found in the absence of friendly partners.

In contrast with Figures 6 and 7, Figure 9 shows that generator complexity in the three-player game is not at all consistent over evolutionary time; we see an overall

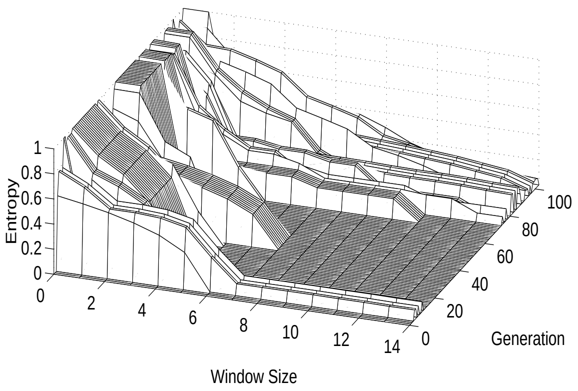

Figure 9 Entropy Contours from Full Game. Overall increase in complexity indicates features of an arms race.

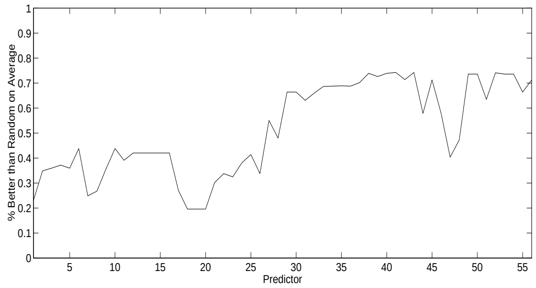

Figure 10 Skill Accumulation in Pursuer. Champion predictors that arise later in evolutionary time are more generalized and perform better against the set of champion generators.

increase in both order and entropy. Figure 10 graphs the performance of the champion pursuers from this run against the set of champion generators of Figure 9. We see clearly that the pursuers created later in evolutionary time are more skillful. The abilities of the friendly predictors likewise increase. Thus, we have reasonable indications of an arms race between the cooperating agents (evaders and friendly partners) and the hostile pursuers.

More typically, however, the three-player game produces punctuated shifts in generator behavior, from simple to complex and back again, rather than the monotonic increase in complexity indicative of an arms race. Indeed, Figure 9, used to argue for the presence of an arms race above, also shows abrupt retreats in generator complexity. What might the adaptive utility of these sudden changes be? Data from a different run suggests a hypothesis. The top three plots in Figure 11 graph the average population fitness of the friendly predictors, hostile predictors, and generators, respectively. Because average generator fitness values are almost always above 0.5, we know that the generators usually succeed in making themselves more predictable to the friendly partners than to the hostile pursuers. But, the degree of their success varies considerably during the run.

The bottom plot in Figure 11 depicts the order of the best generator in each generation. We see that generator

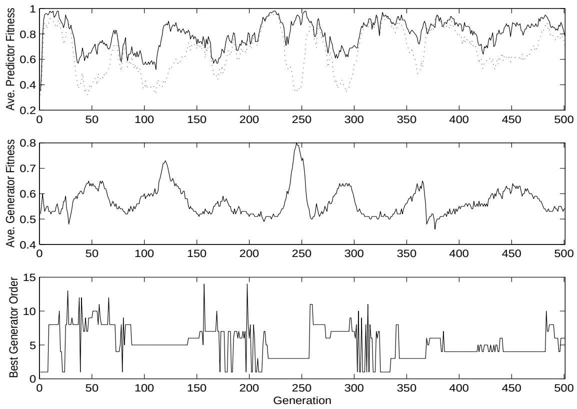

Figure 11 Punctuated Shifts in Generator Behavior in Full Game. From top to bottom, average fitness of friendly predictor partners, pursuer predictors, and generators, and order of best generator in generation, over evolutionary time.

order is most stably maintained over the onset and duration of periods of generator success — periods where the evaders and partners share a private convention for coordinated behavior. This very stability is what allows the hostile pursuers to eventually learn the convention, however. Once this happens, the propriety of the convention is lost and generator fitness reaches its nadir. This tends to be the moment at which the order of the best generators becomes unstable, often alternating between high-order and low-order behaviors. We suspect that this instability is adaptive because it forces the two, now similarly skilled, predictor populations to diverge in ability. Once sufficiently differentiated, some new behavioral convention can be found that only the friendly partner can predict. A new period of evader/partner coordination thus emerges.

The system dynamics of the three-player domain can resemble pursuit and evasion behavior; the *meta*-PE game exists between populations and is played over evolutionary time. The hostile pursuer population is "chasing" the behavioral convention of the evader and partner populations. To escape, the evader and partner populations must alter their convention. In this particular run, the specific escape strategy looks like a type of protean behavior on the part of the generator population; when the pursuer population "catches" the behavioral convention, the erratic changes in generator order serve to mislead the pursuer's evolutionary course. The earlier reference to cycling in host/parasite gene sequences is particularly apt here.

We must point out, however, that this run presents a particularly vivid example of the meta-PE phenomenon. Other runs show the predictor population to be, as a whole, unable to converge onto the behavioral convention of the evader and partner populations; in these cases, changes in generator behavior cannot so easily be tied to those of the pursuers. Nevertheless, we believe that our methodology will allow us to clarify our data by continuing work with hand-built environments.

7. Conclusions

We show that pursuer-evader can be reformulated as a one-dimensional, time-series prediction game. Information theoretic tools provide quantitative analyses and help operationalize the domain. Because the linear PE game emphasizes the informational (in the technical sense) aspects of pursuit and evasion behavior, it captures a fundamental aspect of communication.

Our behavioral metrics also yield informative views of coevolutionary dynamics. Though we can create good evaders and pursuers with simple evolution, we find that successful coevolution does not automatically follow. Subtleties in scoring method strongly influence the outcome of the two-player game; one method leads to mediocrity while the other defines an optimal strategy (protean behavior) and closes the world. The combination of competitive and convergent pressures of the three-player game is needed to avoid simple mediocre stable-states while keeping an open world.

Tools exist to characterize generator behavior and hand-build predictors of known power. This toolset has recently been augmented to include new methods that allow us to construct by hand generators with particular entropy contours and systematically analyze the generality and power of evolved predictors. Our future work will feature these techniques in a continued investigation of generator complexity, the computational demands generators place on predictors, and the ease with which predictors evolve against them. The final results we report are not fully understood and will require these tools to elucidate.

8. Acknowledgments

The authors gratefully acknowledge the many hours of conversation that have contributed to this work provided by Alan Blair, Marty Cohn, Paul Darwen, Pablo Funes, Greg Hornby, Ofer Melnik, Jason Noble, Elizabeth Sklar, and particularly Richard Watson.

References

Ackley, D. H. and Littman, M. L. (1994). Altruism in the evolution of communication. In (Brooks and Maes, 1994), pages 40–48.

Akiyama, E. and Kaneko, K. (1997). Evolution of communication and strategies in an iterated three-person game. In Langton, C. G. and Shimohara, K., editors, *Artificial Life V (1996)*, pages 150–158. MIT Press.

Angeline, P. J. and Pollack, J. B. (1994). Competitive environments evolve better solutions for complex tasks. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*. Morgan Kaufmann.

Angeline, P. J., Saunders, G. M., and Pollack, J. B. (1994). An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5:54–65.

Brooks, R. A. and Maes, P., editors (1994). *Artificial Life IV*. MIT Press.

Cliff, D. and Miller, G. F. (1995). Tracking the red queen: Measurments of adaptive progress in co-evolutionary simulations. In Moran, F. et al., editors, *Third European Conference on Artificial Life*, pages 200–218. Springer Verlag.

Cliff, D. and Miller, G. F. (1996). Co-evolution of pursuit and evasion 2: Simulation methods and results. In (Maes et al., 1996), pages 506–515.

Crutchfield, J. P. (1994). The calculi of emergence: Computation, dynamics, and induction. *Physica D*, 75:11–54.

Di Paolo, E. A. (1997). Social coordination and spatial organization: Steps towards the evolution of communication. In Husbands, P. and Harvey, I., editors, *Fourth European Conference on Artificial Life*, pages 464–473. MIT Press.

Gregory M. Saunders, J. B. P. (1996). The evolution of communication schemes over continuous channaels. In (Maes et al., 1996), pages 580–589.

Hamilton, W. D., Axelrod, R., and Tanese, R. (1990). Sexual reproduction as an adaptation to resist parasites (a review). *Proc. Natl. Acad. Sci. USA*, 87:3566–3573.

Hamming, R. W. (1980). *Coding and Information Theory*. Prentice-Hall, Inc., Englewood Cliffs, NJ.

Hashimoto, T. and Ikegami, T. (1996). Emergence of net-grammar in communicating agents. *BioSystems*, 38(1):1–14.

Isaacs, R. (1965). *Differential Games*. John Wiley and Sons, New York.

Kaneko, K. and Suzuki, J. (1994). Evolution to the edge of chaos in an imitation game. In Langton, C. G., editor, *Artificial Life III (1992)*, pages 43–54. Addison-Wesley.

Koza, J. (1992). *Genetic Programming*. MIT Press.

Langton, C. G. and Shimohara, K., editors (1997). *Artificial Life V (1996)*. MIT Press.

Maes, P. et al., editors (1996). *From Animals to Animats IV*. MIT Press.

Miller, G. F. and Cliff, D. (1994). Protean behavior in dynamic games: Arguments for the co-evolution of pursuit-evasion tactics. In Cliff, D. et al., editors, *From Animals to Animats III*, pages 411–420. MIT Press.

Oliphant, M. and Batali, J. (1996). Learning and the emergence of coordinated communication. *(Submitted)*.

Pollack, J. B., Blair, A., and Land, M. (1997). Coevolution of a backgammon player. In (Langton and Shimohara, 1997).

Reynolds, C. (1994). Competition, coevolution and the game of tag. In (Brooks and Maes, 1994), pages 59–69.

Steels, L. (1997). Self-organising vocabularies. In (Langton and Shimohara, 1997), pages 136–141.

Werner, G. M. and Dyer, M. G. (1991). Evolution of communication in artificial organisms. In Langton, C., Taylor, C., Farmer, J., and Rasmussen, S., editors, *Artificial Life II (1990)*, pages 659–687. Addison-Wesley.

Zhu, H. and Kinzel, W. (1997). Anti-predictable sequences: Harder to predict than a random sequence. *(Submitted)*.