Game Theory and the Simple Coevolutionary Algorithm: Some Preliminary Results on Fitness Sharing

Sevan G. Ficici and **Jordan B. Pollack**
DEMO Lab—Computer Science Department
Brandeis University
Waltham Massachusetts USA
www.demo.cs.brandeis.edu

Abstract

Coevolutionary domains are fundamentally game-theoretic in nature because the merit of an agent cannot be surmised without having it interact with other agents. In this paper, we explore the relevance of game-theory to the analysis of coevolutionary algorithm dynamics. Particularly, we show how a game-theoretic framework allows us to ascertain the effects that fitness sharing can have on a coevolutionary algorithm in a variable-sum game. We show analytically that competitive fitness sharing and similarity-based fitness sharing can cause a coevolutionary algorithm to converge to solutions that lack game-theoretic justification.

1 Introduction

We define *domain merit* to be the worth an entity has with respect to some problem domain; the *objective function* is the evaluator of merit. If our domain task is to optimize a mathematical function, then the function itself provides the natural evaluator of merit; our objective (maximizing the function) coincides perfectly with the behavior of the evaluator function, and merit is expressed with complete fidelity. But, many problem domains we care about do not have such precise and transparent ways to characterize the merit of a candidate solution. Often, we need to apply heuristics to estimate merit, and, when we do, the intended objective and expressed objective diverge. The extent of divergence is generally unknowable. Thus, we are not infrequently surprised to find an obviously poor candidate that the objective function opines "good." Such is also, and perhaps characteristically, the case for coevolutionary domains.

The essential difference between an ordinary evolutionary algorithm and a *co*-evolutionary algorithm lies in the evaluation process. We define a coevolutionary domain to be one that is *interactive*. By this we mean that the computation of an entity's domain merit cannot be made without having the entity interact with other entities. The outcome of assessment is *contingent* upon those other entities with which the interaction takes place; some interactions may cause an entity to perform well in the domain, while others may cause it to perform poorly. Thus, an interactive domain is fundamentally game-theoretic in nature.

For example, how do we construct a heuristic to measure the quality of a checkers player? One approach is to make a list of characteristics that describe components of an expert strategy, and rate players according to the number of such characteristics they display. This metric is difficult to construct. Another approach is to test players against all possible opponent strategies and rate them accordingly. This method is computationally infeasible. A much simpler heuristic is to play the evolving strategies against each other and use the outcomes as indicators of over-all quality. This heuristic is precisely what a simple coevolutionary algorithm implements. And, like any heuristic, the outcomes of a limited number of interactions can, at times, be poor indicators of domain merit. For example, we can have one strategy A that is generally good, but has a specific weakness, and another strategy B that is generally poor, but happens to exploit the weakness of A. When played against B, strategy A actually looks worse than B. Intransitive superiority chains are often a major feature of interactive domains [Cliff and Miller, 1995]. If the domain is open-ended, as in the Iterated Prisoner's Dilemma [Axelrod, 1984], for example, then domain merit can only be defined with respect to a known set of strategies.

We believe that game theory provides an effective formalism for investigating the ways in which the coevolutionary heuristic may fail. In this paper, we discuss the relevance of game theory to coevolutionary algorithms and use game theory to examine the behavior of fitness sharing methods in coevolutionary domains. We show analytically that fitness sharing methods can cause a coevolutionary algorithm to converge to solutions that lack game-theoretic justification in variable-sum games.

This paper is organized as follows. Section 2 discusses how we come to view game theory as an effective formalism for understanding coevolutionary algorithms. Section 3 introduces the discrete-time equations used in evolutionary game theory (EGT). Section 4 discusses the kinds of equilibria that EGT can reach, looks at the equations to determine *polymorphic* equilibria in two-strategy games, and examines the equilibria and dynamics of the *Hawk-Dove* game. Section 5 contrasts the equilibria and dynamics of competitive fitness sharing [Rosin, 1997] in two-strategy games with results from Section 4. Section 6 conducts a similar examination of similarity-based fitness sharing [Goldberg, 1989]. Finally, Section 7 offers some closing remarks.

2 Game Theory in Coevolution

2.1 Goals vs. Dynamics

To comprehend the operation of a coevolutionary algorithm, we can either attempt to understand it purely as a dynamical system, or we can attempt to understand its dynamics with respect to some desired result or behavior. The approach we choose depends upon the purpose for which the algorithm is constructed. The former approach is appropriate if, for example, the algorithm is meant to model specific mechanisms that we hypothesize to operate in nature. In this case, the behavior of the algorithm is viewed as a consequence of algorithm design. In contrast, the latter approach is appropriate if we begin with a particular outcome in mind and seek to build a mechanism capable of achieving or approximating that outcome. In this case, the design of the algorithm is viewed as a consequence of the desired behavior. We use this second approach to consider coevolutionary algorithm dynamics, for we presume that the coevolutionary algorithm performs optimization.

2.2 Optimality vs. Adaptation

The question as to whether evolutionary algorithms really do perform optimization is contentious (e.g., [De Jong, 1993]). Note that, though we presume coevolutionary algorithms optimize, we do not claim that they optimize. The utility of our presumption is that it allows us to invoke a principled optimality concept against which to measure observed behavior, thereby allowing us ascertain the degree to which optimization actually occurs and explicate the dynamics of coevolutionary algorithms.

2.3 Nash Equilibrium

Let us assume an interactive domain, or *game*, where each interaction involves m participants. Let us further assume that we can enumerate the entire space of possible behaviors, or *strategies*, that any participant can exhibit. From this enumeration, we can construct an m-dimensional table, or *payoff matrix*, that lists the (expected) outcome of every possible m-way interaction of strategies. The outcome of an interaction is specified by the *payoff*—a scalar—each participant is to receive as a result of the interaction. For any interactive domain that can be fully described by a payoff matrix, the only reasonable optimality concept to apply is that of Nash equilibrium [Binmore, 1992].

3 Evolutionary Game Theory

The primary contribution of evolutionary game theory (EGT) is the concept of the *evolutionary stable strategy* (ESS) [Maynard-Smith, 1982]. The ESS is a refinement of Nash equilibrium that does away with the traditional assumption of agent rationality. Instead, Maynard-Smith shows that game-theoretic equilibria can be achieved through a process of Darwinian selection. Nevertheless, the ESS is defined as a static concept, and since its introduction many other stability concepts have been proposed [Lessard, 1990], including those that are more properly rooted in dynamical systems theory [Rowe et al., 1985]. For the games studied in this paper, the ESS corresponds to a dynamical attractor [Strogatz, 1994].

A symmetric two-player game consisting of n *pure strategies* is described by an n by n *payoff matrix* $\mathbf{G}$. If Player 1 plays strategy i and Player 2 plays strategy j, then their payoffs are given by matrix entries $\mathbf{G}_{i,j}$ and $\mathbf{G}_{j,i}$, respectively. We assume a single infinitely large population of agents, where each agent plays some pure strategy. The state of a population is represented by column vector $\mathbf{p}$, where element $\mathbf{p}_i$ represents the proportion with which game strategy i appears in the population of agents. Thus, each element $\mathbf{p}_i$ has a value between zero and one (inclusive), and the elements must sum to one.

We assume *complete mixing*: all possible pair-wise interactions between agents takes place at each time step (i.e., generation). The cumulative payoff received by an agent playing strategy i upon interacting with all other agents is the weighted sum of row i (of the payoff matrix), the weights being the proportion vector $\mathbf{p}$. We can compute these cumulative payoffs for all strategies by multiplying vector $\mathbf{p}$ by matrix $\mathbf{G}$, as shown in Equation 1. The constant w_0 (derived in Equation 2) is added to each strategy's cumulative payoff to obtain strategy *fitness*, where $\mathbf{w}_i$ is the fitness of strategy i (and hence all agents that play strategy i). We can think of w_0 as a baseline fitness with which each strategy is initially endowed before interaction takes place. (Note that $\mathbf{w_0}$ refers to either a column vector or matrix, as appropriate, where all elements are w_0.)

Given the population state and fitness vectors at time t, $\mathbf{p}^t$ and $\mathbf{w}^t$, we compute the new state of our evolving population, $\mathbf{p}^{t+1}$, with Equation 3. This difference equation is known as a *replicator*, and causes the number of agents playing each strategy to increase (replicate) in proportion to fitness. We then re-normalize strategy proportions to keep the population state within the simplex. Note that strategies absent from the initial population cannot appear at later time steps, since we are describing a selection-only system—no variational operators (i.e., mutation or recombination) exist to introduce strategies not already present.

$$\mathbf{w} = \mathbf{G}\mathbf{p} + \mathbf{w_0} \tag{1}$$

$$w_0 = |\ \min(\min(\mathbf{G}), 0)\ | + k, k \geq 0 \tag{2}$$

$$\mathbf{p}^{t+1} = \text{normalize}(\text{diag}(\mathbf{p}^t)\mathbf{w}^t) \tag{3}$$

Since an agent can have no fewer than zero offspring, and reproductive success is proportional to fitness, an agent's fitness value must be greater than or equal to zero. Thus, we define w_0 to be large enough to ensure that fitness values are non-negative, even if cumulative payoffs are negative. Finally, as shown in Equation 4, the baseline fitness w_0 can instead be added to each element in the payoff matrix $\mathbf{G}$ before multiplication with vector $\mathbf{p}$ to achieve the same fitness values as in Equation 1 (recall that the elements of $\mathbf{p}$ sum to one). Therefore, given a game $\mathbf{G}$ (with payoffs $a, b, \ldots$) and a background fitness w_0, we can define a new game $\mathbf{G}'$ (with payoffs $A, B, \ldots$), as shown in Equation 5.

$$\mathbf{w} = (\mathbf{G} + \mathbf{w_0})\mathbf{p} = \mathbf{G}\mathbf{p} + \mathbf{w_0}\mathbf{p} = \mathbf{G}\mathbf{p} + \mathbf{w_0} \tag{4}$$

$$\mathbf{G}' = \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \mathbf{G} + \mathbf{w_0} = \begin{pmatrix} a + w_0 & b + w_0 \\ c + w_0 & d + w_0 \end{pmatrix} \tag{5}$$

4 Standard Replicator

We are interested in how the dynamics and equilibria of the standard discrete-time replicator (Equation 3) are affected when the calculation of strategy fitness (Equation 1) is modified to implement various forms of fitness sharing. In this section, we examine the replicator's normative behavior (i.e., in the absence of fitness sharing) with the *Hawk-Dove* game, which we will use to illustrate analytical results throughout the paper.

The payoff matrix to the *Hawk-Dove* game is shown in Equation 6. This is a symmetric, variable-sum game of two pure strategies (Hawk and Dove). Such games always cause the replicator to converge to a point attractor that is an ESS, and hence a Nash equilibrium [Maynard-Smith, 1982]. Point attractors are either *monomorphic* or *polymorphic*. A monomorphic attractor is a population state comprised of a single pure strategy (where all the other strategies are driven to "extinction," that is, an infinitely small proportion of the infinitely large population). A polymorphic attractor is a population state comprised of more than one pure strategy, where the surviving strategies have reached a *fitness equilibrium* at some particular strategy proportion. As we will see below, the Hawk-Dove game has a polymorphic attractor.

$$G = \begin{array}{c|cc} & \text{H} & \text{D} \\ \hline \text{H} & -25 & 50 \\ \text{D} & 0 & 15 \end{array} \tag{6}$$

4.1 Equilibria

Given an arbitrary game of two pure strategies x and y, we can find the proportion at which the strategies reach fitness equilibrium (assuming that such a proportion exists) by setting $\mathbf{w}_x = \mathbf{w}_y$ and solving for $\mathbf{p}_x$. Because a two-strategy game yields a one-dimensional system, the value of $\mathbf{p}_y$ can be inferred: $\mathbf{p}_y = 1 - \mathbf{p}_x$. The solution is shown in equation 7. If the game in question does not have a polymorphic equilibrium, then one of $\mathbf{p}_x$ or $\mathbf{p}_y$ will be negative.

The important feature of Equation 7 is that the fitness equilibrium ratio between $\mathbf{p}_x$ and $\mathbf{p}_y$ does not depend on the value of w_0. Indeed, the polymorphic equilibria of Equation 3 are independent of w_0 regardless of the number of strategies [Rowe et al., 1985]. If this is the case, we may ask why we need w_0. Though the value of w_0 is unimportant at fitness equilibrium, it is important when the population state is away from equilibrium—recall that we need it to prevent negative fitness values. Thus, there exist uncountably many games with the same set of fitness equilibria.

$$\mathbf{p}_x = \frac{D-B}{A-B-C+D} =$$

$$\frac{d+w_0-b-w_0}{a+w_0-b-w_0-c-w_0+d+w_0} =$$

$$\frac{d-b}{a-b-c+d} \tag{7}$$

4.2 Dynamics

Since two-strategy games give one-dimensional systems, we can depict the dynamics of the Hawk-Dove game with a *return map*; the x-axis represents the state of the population $\mathbf{p}_x$ (proportion of Hawks in the population) at time t and the y-axis represents $\mathbf{p}_x$ at time $t+1$. Figure 1 shows the return map of the Hawk-Dove game at two different values of w_0. Both curves intersect the diagonal line at the same three points; each of these intersections represents a *fixed point* of the dynamical system, where the population state does not change from one time-step to the next. The fixed-point in the middle of the graph is the polymorphic fitness equilibrium of the Hawk-Dove game. The proportion of Hawks at this equilibrium is $\mathbf{p}_x = (15-50)/(-25-50-0+15) = 7/12$.

The slope of the curve at a fixed-point determines the fixed-point's *stability*. If the absolute value of the slope is less than one, then the fixed-point is stable. We see that both curves in Figure 1 meet this requirement, and so the polymorphism is an (indeed, the only) attractor of the dynamical system. Though w_0 affects neither the stability nor the location of the polymorphism, we see that larger values of w_0 bring the curve closer to the diagonal, which results in a dynamical system that moves more slowly to the attractor.

5 Competitive Fitness Sharing

Competitive fitness sharing is a diversity maintenance technique for coevolution proposed by Rosin [Rosin, 1997]. The games Rosin considers are a subset of zero-sum games—a strategy either wins or loses to another strategy. When an agent α defeats an opponent β, the reward given to α is $1/n$, where n is the total number of agents capable of defeating β.

5.1 Equilibria

We can easily extend competitive fitness sharing to variable sum games. The payoff agent α (playing strategy i) receives from interacting with agent β (playing strategy j) is $\frac{\mathbf{G'}_{i,j}}{d}$, where d is the total payoff awarded

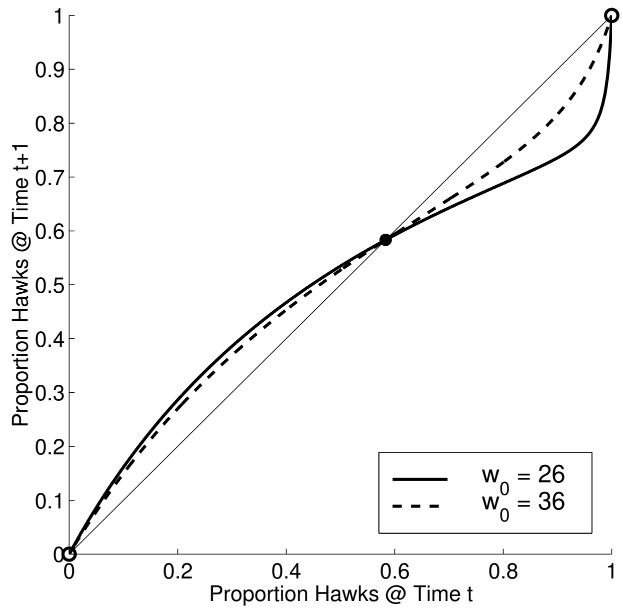

Figure 1: Return map of Hawk-Dove game using different values of w_0.

to all agents by interacting with agent β. In a two-strategy game, we calculate the "shared" fitnesses for strategies x and y as shown in Equation 8.

Given these equations, we may set them to be equal and solve for $\mathbf{p}_x$ to find the proportion at which the strategies achieve fitness equilibrium, assuming an equilibrium exists. (Again, if a polymorphism does not exist, then one of $\mathbf{p}_x$ or $\mathbf{p}_y$ will be negative.) Rosin [Rosin, 1997] proves that the equilibria for shared and non-shared fitness are the same in win-lose games. But, as Equation 9 shows, this is not true in the general case. We see instead that, for an arbitrary two-strategy game, the equilibrium point depends upon the value of w_0. Further, as w_0 gets larger, the equilibrium approaches that of non-shared fitness.

$$\begin{aligned} \mathbf{w}_x &= \frac{\mathbf{p}_x A}{\mathbf{p}_x A + \mathbf{p}_y C} + \frac{\mathbf{p}_y B}{\mathbf{p}_x B + \mathbf{p}_y D} \\ \mathbf{w}_y &= \frac{\mathbf{p}_x C}{\mathbf{p}_x A + \mathbf{p}_y C} + \frac{\mathbf{p}_y D}{\mathbf{p}_x B + \mathbf{p}_y D} \end{aligned} \tag{8}$$

$$\mathbf{p}_x = \frac{C(D-B)}{BA-2BC+CD} =$$

$$\frac{c(d-b)+w_0(d-b)}{ab-2bc+cd+w_0(a-b-c+d)} \tag{9}$$

5.2 Dynamics

Figure 2 shows the return map of the Hawk-Dove game under competitive fitness sharing with the same two values of w_0 used in Figure 1. As Equation 9 indicates, the fitness equilibrium moves closer to 7/12 as w_0 increases. Again we see that the curves approach the diagonal as w_0 increases, meaning that the difference between two consecutive population states decreases. A value of w_0 sufficiently large to reasonably approximate the equilibrium of 7/12 will cause the dynamics to move very slowly, indeed.

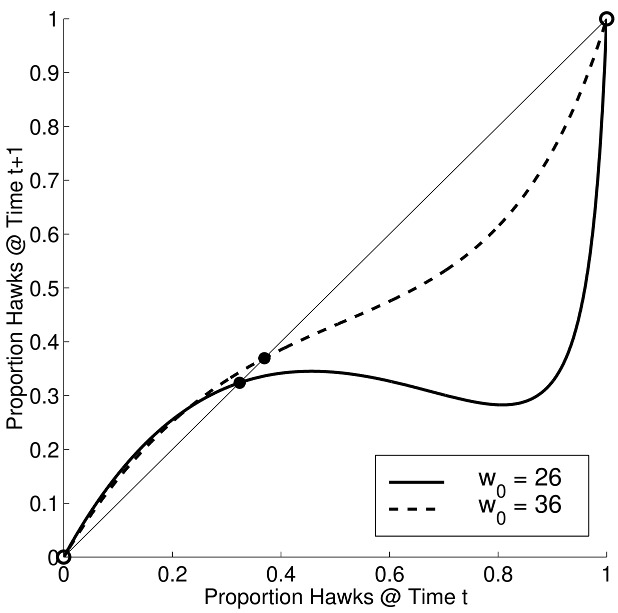

Figure 2: Return map of Hawk-Dove game with competitive fitness sharing using different values of w_0.

6 Similarity-Based Sharing

An older method for diversity maintenance discounts an agent's fitness based upon how genotypically or phenotypically redundant it is with respect to the rest of the population. This method is known as *similarity-based sharing* [Goldberg, 1989] and was originally used in non-coevolutionary algorithms. Since this method requires a similarity metric to measure redundancy, and we can imagine a number of similarity metrics, there exists many ways to formulate this procedure. Here we investigate one very simple formulation: we consider two agents to be similar only if they are playing the same strategy. This metric is analytically convenient, but unlikely to be used in real-world domains.

6.1 Equilibrium

Given a game of n pure strategies $\mathbf{G}'$, the shared fitness of an agent playing strategy i is the dot product of payoff matrix row i and population state vector $\mathbf{p}$, divided by the number of agents playing strategy i: $\mathbf{w}_i = \sum_j \mathbf{G'}_{i,j}\mathbf{p}_j/\mathbf{p}_i$. Equation 10 shows how we calculate shared fitness based on similarity in a two-strategy game.

$$\begin{aligned} \mathbf{w}_x &= \frac{\mathbf{p}_x A + \mathbf{p}_y B}{\mathbf{p}_x} = A + \frac{\mathbf{p}_y}{\mathbf{p}_x} B \\ \mathbf{w}_y &= \frac{\mathbf{p}_x C + \mathbf{p}_y D}{\mathbf{p}_y} = \frac{\mathbf{p}_x}{\mathbf{p}_y} C + D \end{aligned} \tag{10}$$

Given Equation 10, we can solve to find the proportion at which two strategies reach fitness equilibrium, if they do at all. This is somewhat more complex than in competitive fitness sharing because we must find the roots of the polynomial in Equation 11. Using the quadratic equation, we get Equation 12. We again find that the location of the fitness equilibrium is dependent upon the magnitude of w_0.

Indeed, as w_0 approaches infinity, $\mathbf{p}_x$ approaches 1/2 regardless of the payoffs in the game. For this to be true, the square root term must behave as shown in Equation 13. The $(a-d)^2$ term in Equation 13 becomes insignificant as w_0 grows, so we concentrate on the remaining terms, as shown in Equation 14. The square root of Equation 14 is approximated by the square root of Equation 15 as w_0 increases, which is the right side of Equation 13.

$$\mathbf{p}_x^2(-A+B-C+D)+\mathbf{p}_x(A-2B-D)+B=0 \tag{11}$$

$$\mathbf{p}_x = \frac{-a+2b+d+2w_0}{2(-a+b-c+d)} \pm \frac{\sqrt{(a-d)^2+(2b+2w_0)(2c+2w_0)}}{2(-a+b-c+d)} \tag{12}$$

$$\sqrt{(a-d)^2+(2b+2w_0)(2c+2w_0)} \equiv_{w_0\to\infty} b+c+2w_0 \tag{13}$$

$$(2b+2w_0)(2c+2w_0) = 4bc+4bw_0+4cw_0+4w_0^2 \tag{14}$$

$$(b+c+2w_0)^2 = b^2+bc+c^2+4bw_0+4cw_0+4w_0^2 \tag{15}$$

6.2 Dynamics

Figure 3 shows the return map for the Hawk-Dove game using our formulation of similarity-based fitness sharing. The values of w_0 we use are 26 and 56. We see that the polymorphic equilibrium moves towards 0.5 as w_0 increases. For smaller values of w_0, such as 26, this system approaches equilibrium via a damped oscillation rather than a monotonic convergence. Another feature of this map is that it is discontinuous at $\mathbf{p}_x = 0$ and $\mathbf{p}_x = 1$, unlike the previous two maps.

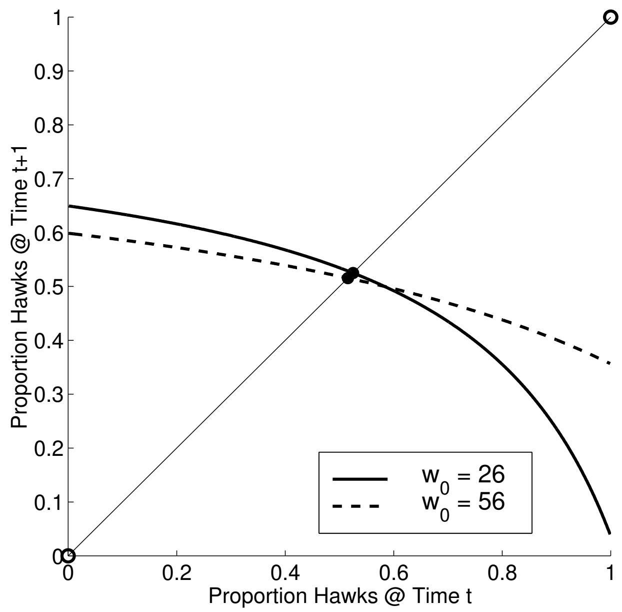

Figure 3: Return map of Hawk-Dove game with similarity-based fitness sharing using different values of w_0.

7 Conclusions and Future Work

We argue that game theory is relevant to the study of coevolutionary algorithm dynamics and use it to show that fitness sharing methods, when applied to variable-sum games, can alter selection pressures such that a coevolutionary system will converge to solutions that lack game-theoretic justification. Specifically, we show that fitness-sharing procedures cause the polymorphic attractor of the Hawk-Dove game, which is a Nash equilibrium in the absence of fitness sharing, to deviate such that it is no longer Nash. We believe that our result is particularly relevant to the vast body of research on the Iterated Prisoner's Dilemma.

Nevertheless, we are clearly in the early phases of our investigation. For example, we need to expand our mathematics to understand how fitness sharing techniques behave as the number of strategies in a polymorphism exceeds two. We are also currently conducting several coevolutionary simulations where agents are allowed to evolve *mixed-strategies* (probability distributions over pure strategies) for the Hawk-Dove game. Preliminary results indicate that fitness sharing methods continue to move the system away from Nash equilibria.

Acknowledgements

The authors wish to thank members of the DEMO Lab and the reviewers for many helpful comments.

References

[Axelrod, 1984] Axelrod, R. (1984). *The Evolution of Cooperation.* Basic Books.

[Binmore, 1992] Binmore, K. (1992). *Fun and Games: A Text on Game Theory.* D.C. Heath, Lexington, MA.

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

[De Jong, 1993] De Jong, K. A. (1993). Genetic algorithms are not function optimizers. In Whitely, L. D., editor, *Foundations of Genetic Algorithms 2 (FOGA)*, pages 5–18. Morgan Kaufmann.

[Goldberg, 1989] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

[Lessard, 1990] Lessard, S. (1990). Evolutionary stability: one concept, several meanings. *Theoretical Population Biology*, 37:159–170.

[Maynard-Smith, 1982] Maynard-Smith, J. (1982). *Evolution and the Theory of Games.* Cambridge University Press.

[Rosin, 1997] Rosin, C. D. (1997). *Coevolutionary Search Among Adversaries.* PhD thesis, University of California, San Diego.

[Rowe et al., 1985] Rowe, G. W., Harvey, I. F., and Hubbard, S. F. (1985). The essential properties of evolutionary stability. *Journal of Theoretical Biology*, 115:269–285.

[Strogatz, 1994] Strogatz, S. H. (1994). *Nonlinear Dynamics and Chaos.* Addison-Wesley.