Analyzing Cooperative Coevolution with Evolutionary Game Theory

R. Paul Wiegand
George Mason University
Computer Science Department
Fairfax, VA 22030
paul@tesseract.org

William C. Liles[1]
Central Intelligence Agency
Washington, DC 20505
wliles@gmu.edu

Kenneth A. De Jong
George Mason University
Computer Science Department
Fairfax, VA 22030
kdejong@gmu.edu

Abstract— **The task of understanding coevolutionary algorithms is a very difficult one. These algorithms search landscapes which are in some sense adaptive. As a result, the dynamical behaviors of coevolutionary systems can frequently be even more complex than traditional evolutionary algorithms (EAs). Moreover, traditional EA theory tells us little about coevolutionary algorithms. One major question that has yet to be clearly addressed is whether or not coevolutionary algorithms are well–suited for optimization tasks. Although this question is equally applicable to competitive, as well as cooperative approaches, answering the question for cooperative coevolutionary algorithms is perhaps more attainable.**

Recently, evolutionary game theoretic (EGT) models have begun to be used to help analyze the dynamical behaviors of coevolutionary algorithms. One type of EGT model which is already reasonably well understood are *multi–population symmetric games*. **We believe these games can be used to analytically model cooperative coevolutionary algorithms. This paper introduces our analysis framework, explaining how and why such models may be generated. It includes some examples illustrating specific theoretical and empirical analyses. We demonstrate that using our framework, a better understanding for the degree to which cooperative coevolutionary algorithms can be used for optimization can be achieved.**

I. INTRODUCTION

In the past few years there has been an increase in interest in coevolutionary algorithms (CEAs), i.e., evolutionary algorithms (EAs) in which the fitness of individuals is not determined in an independent fashion by invoking an external fitness function, but rather the fitness of an individual is determined by its interactions with other individuals in the evolutionary system. Much of the work on CEAs has focused on two kinds of interaction: *competitive* coevolutionary systems and *cooperative* coevolutionary systems. Competitive CEAs are natural models for evolving objects such as game playing programs for which it is difficult to write an external fitness function, but quite simple to define fitness in terms of competitive success against other programs in the evolving population (e.g., [1]). Cooperative CEAs are natural models for evolving complex objects by decomposing them into subassemblies that coevolve, and subassembly fitness is determined by how well it works with the other subassemblies in producing a complete object (e.g., [2]).

It is very clear from this work that the behavior of CEAs in general is quite different from standard (non-coevolutionary) EAs, and furthermore, competitive CEAs behave quite differently than cooperative CEAs. This raises two important questions: what body of EA theory can be used to help us understand CEAs better and what kinds of problems are CEAs best suited for? A key element in answering these questions is a better understanding of the dynamics of CEAs which are considerably more complex than standard (non-coevolutionary) EAs, since the fitness landscape is itself evolving over time. This means a CEA is in principle a much more open-ended adaptive system [3] in the sense that it can evolve entirely new fitness gradients over time[1].

As a consequence much of the initial work in theoretical CEA analysis has focused on tools for modeling and analyzing CEA dynamics. One of the more promising recent developments has been the use of evolutionary game theory (EGT) to provide a dynamical systems analysis of CEAs [4]. A dynamics systems analysis directly addresses the first of the two questions raised earlier by providing a better understanding of CEA dynamics, but only indirectly addresses questions such as how effective are CEAs at solving difficult optimization problems.

In this paper we focus on *cooperative* CEAs (CCEAs) and present two interesting results. First we show how EGT can be used quite naturally to model cooperative CEAs, even though intuitively one might expect that EGT is better suited for competitive CEA analysis. Second, because cooperative CEAs have a more natural interpretation as solving an optimization problem, the EGT analysis also provides some insight into the behavior of CCEAs as optimizers.

We begin with some background discussion regarding evolutionary game theory, and introduce the concept of *multi–population symmetric games* (MPS) [5], the subclass of EGT models that will be used to model and analyze these kinds of cooperative coevolutionary algorithms. We then describe the particular CCEAs under study and show how MPS can be used to model them. In section four we present our initial MPS analysis, introduce the idea of using *rain gauge measures* as a form of empirical model validation [6], and then use this method to shed light of the behavior of a CCEA as an optimizer on two simple functions. Finally, we will conclude by discussing the impact of these results, both in terms of the

---

[1] This material has been reviewed by the CIA. That review neither constitutes CIA authentication or information nor implies CIA endorsement of the author's views.

specifics of what is learned, as well as the general message regarding the applicability of the MPS game framework.

II. EVOLUTIONARY GAME THEORY

EGT describes a set of dynamical systems models for which modern dynamical systems theory can be used to analyze evolutionary processes [5], [7], [8] and, as we will see, coevolutionary algorithms fit very nicely into this game–theoretic framework.

In evolutionary game theory, we are working with models of populations of individuals who are interacting with one another. Individuals repeatedly meet and receive some reward or punishment for the experience (payoff), based on how their genes tell them to ”play the game” (strategy). This process, influenced by selection, continues and we (the modelers) observe the change in the population(s) that result. Traditionally standard, fitness proportional selection is assumed and is used in this paper as well..

While single population models are common in EGT, this paper will be focusing entirely on two population models, in which individuals interact only with members of the other population, not their own. Further, though many kinds of interactions are possible, we will assume *complete mixing*, meaning that during the evaluation phase of the algorithm, individuals in one population are assumed to have interacted with all members of the alternate population in pair–wise collaborations, and vice–versa.

This research also assumes that populations are infinite in size, and we deal with expected fitness based on distributions across the finite number of pure strategies (genotypes). Additionally, we will concern ourselves with discrete time (generational) models.

Finally, although EGT models typically do not include variation operators, it has been recently noted that modeling variation is possible [4]. We are currently studying formalisms that include variational operators but will not discuss variation in this paper.

A. Notation

Before introducing the mathematical model, some basic notional conventions must be established. Lower case letters will be used to represent real numbers and functions. Vectors will also be denoted with a lower case letter, but will be distinguished with an arrow above the letter (e.g. $\vec{x}$).

Capital letters will be used to denote sets and matrices. Euclidean spaces will be denoted $\Re^n$, where n is a positive integer and indicates the dimensionality of the space. Given a set of connected points in Euclidean space, X, the *interior* of is denoted as $int(X)$. The boundary of X is denoted $bnd(X)$.

The n dimensional unit simplex is denoted Δ^n. The simplex is the subset of Euclidean space such that all components of vectors in that space are inclusively between 0 and 1, and the sum of all the components of a given vector equals one, ($\Delta^n = \{\vec{x} : \vec{x} \in \Re^n, x_i \in [0,1] \forall i = 1 \ldots n, \sum_i^n x_i = 1\}$). The cartesian product of two simplexes is denoted $\Delta^n \times \Delta^m$.

B. Two Population Evolutionary Game Models

Virtually any EA can be considered a dynamical system [9]. By assuming that populations consist of an infinite number of individuals, but a finite number of n distinct possible genotypes, one can represent the state of a population at any given time by a vector of real values each element of which represents the fraction of each genotype in the population. Since such vectors must sum to one, they are all in the unit simplex ($\vec{x} \in \Delta^n$). Hence, changes in the state of a population over time are modeled as dynamical system trajectories in the simplex.

In our case, the system that defines the dynamics of a CCEA is developed using an N population evolutionary game theoretic model. To keep things simple in this paper we show how a two population model is constructed. The key element is understanding how replication is handled. The equations which follow show the two population, discrete time version of replicator dynamics.

$$\begin{aligned} \vec{u} &= A\vec{y} && (1) \\ \vec{w} &= B\vec{x} && (2) \\ x'_i &= x_i \left(\frac{u_i}{\vec{x} \cdot \vec{u}} \right) && (3) \\ y'_i &= y_i \left(\frac{w_i}{\vec{y} \cdot \vec{w}} \right) && (4) \end{aligned}$$

Where $\vec{x}'$ and $\vec{y}'$ represent the new population distributions for the next generation. A and B describe the payoffs associated with each pair of possible interactions. In the above system, A describes the payoffs that the first population, $\vec{x}$, receives when it interacts with members of the second population, $\vec{y}$, and B describes the reverse.

Frequently an additive constant appears in the first equation, $\vec{u} = A\vec{y} + \omega_o$, where $\omega_o = 1 - \min(A)$, as well as a similar constant for the second equation. These constants are used to make sure elements of $\vec{u}$ and $\vec{w}$ are non-negative. This is so that under proportional selection the system remains invariant in the simplex (that points remain in the unit simplex as trajectories are advanced through the replicator dynamics). Since we can increment the entire payoff matrix by a constant amount without affecting the dynamics of the game, we can assume that all our payoff matrices have the property that the payoffs are non-negative [7]. As a result the constants are not really necessary and we will not use them.

One thing worth noting is that the two population model operates as a dynamical system in a somewhat different type of space than a single population model. In both cases the space is invariant to the model, but while a single population EGT model is confined to a single simplex, the two population model moves about the cartesian product of *two* unit simplexes, $\Delta^n \times \Delta^m$.

C. Multi–Population Symmetric Games

The defining characteristic of MPS games is that they are symmetric with respect to their payoff matrices. For two population games, this means they are symmetric when $A = B^T$.

These types of models are very useful for a variety of reasons. First, they closely resemble the particular class of coevolutionary algorithms we will be discussing. In addition, the symmetry assumptions allow for some subtle simplification of the mathematics involved. For example, simple algebraic expansion will show that the weighted average payoff of the first population is the same as that of the second, $\vec{x} \cdot A\vec{y} = \vec{y} \cdot A^T\vec{x}$. These simplifications allow for some interesting observations to be made about dynamics. We will discuss a few of these in a moment, after introducing a somewhat simpler form of the replicator equations.

$$\vec{u} = A\vec{y} \tag{5}$$

$$\vec{w} = A^T\vec{x} \tag{6}$$

$$x'_i = x_i\left(\frac{u_i}{\vec{x} \cdot A\vec{y}}\right) \tag{7}$$

$$y'_i = y_i\left(\frac{w_i}{\vec{x} \cdot A\vec{y}}\right) \tag{8}$$

III. MODELING COOPERATIVE COEVOLUTION

A. Coevolution and Optimization

In coevolutionary algorithms we are evaluating *interactions between individuals*, not the individuals themselves. As a result, the fitness of individuals is subjective in some sense, since it will depend on the interactions formed, and consequently the state of the current populations. With traditional evolutionary algorithms, it is usually clear how a problem can be encoded such that meaningful results are possible. We can construct an objective function to compute fitness and use the limiting population state distributions to indicate which genotype corresponds to the point in the domain of the objective function space. This is possible with a coevolutionary algorithm too, except that the limiting distributions correspond with *interactions* and not objective function domain values. In order to decode this to a meaningful solution, we must have a pretty strong understanding of what the interactions themselves mean in the context of our objective function. For most coevolutionary algorithms, we have no such understanding.

In some sense, optimality of a CEA is best expressed in terms of a type of *adaptive balance* achieved in the system. In the game–theoretic sense, this is generally expressed in terms of Nash equilibrium. In fact, in the two population EGT model we've just defined (including MPS games), dynamically stable fixed points of the system are also Nash equilibria. Clearly if we have a formally defined dynamical systems model, we can begin to analytically address questions of when and how (and if) these fixed points are reached.

But even assuming we had a strong notion of what our ultimate external optimization objective was, is it the case that Nash points (or other equilibria) correspond directly with this notion? In most systems, there are multiple Nash equilibria. When this is so, is it clear that equilibria corresponding to ”more optimal” objectives are more likely to be achieved?

While these issues are true in a general sense, our belief is that it may be easier to resolve such questions for cooperative CEAs since they have a natural interpretation as an optimizer.

B. Cooperative Coevolution

One very simple form of cooperative coevolutionary algorithm (CCEA) has been used successfully to perform function optimization [10]. In this approach, each population is assigned a specific argument for the function to represent, and individuals in a given population must collaborate with individuals from other populations in order to obtain a fitness value (the value of the objective function).

A simple two argument function serves as an example. Suppose we would like to optimize the function $f(x, y) = x^2 + y^2$ using cooperative coevolution. We could represent potential x argument values in one population, and potential y argument values in a second population. We would evolve the two populations separately (i.e.: they do not interbreed), but when it comes time to evaluate an individual in the x population, we will need to use an individual from the y population in order to obtain a value from the objective function. In fact, we will evaluate a given member of x with *every* member of y (a collaborator), and take the average of the resulting fitness values (i.e. complete mixing). The same process is true in reverse for the y population, with respect to collaborations from x .

We focus on this model for analysis for three main reasons. First, it is an example of coevolution which empirically seems to perform well on optimization tasks. Second, it is very clear how to encode an optimization problem in such a way that solutions to the systems correspond with meaningful results. Finally, we can use MPS game model to characterize these systems.

A bit more about the second point should be discussed. The encoding/decoding question is addressed by two properties: distinct populations and symmetry in the fitness mechanism. The symmetric property affords the algorithm with the simplicity that the two populations are ”playing the same game”, that is: optimizing the same function (even though they play different roles). We can encode the objective function right into the collaboration mechanism, and we can easily understand what objective domain value is indicated by the converged populations.

C. Modeling CCEAs with MPS Games

It should be clear by now that a MPS game is a direct analogy of a cooperative coevolutionary algorithm. Assuming the number of distinct genotypes for each population is finite, we can elicit a payofff matrix for the first population by simply determining the objective function values at each combination of genotypes with the opposing population's genotypes. Since the game is symmetric, the second population uses the transpose of this matrix.

Such matrices are relatively easy to construct, given some objective function. For this article, we draw two examples from a simple class of functions: MAX OF TWO QUADRATICS (f_{MTQ}). The equations below describe this function.

$$
\begin{aligned}
quad_1(x,y) &= k_1 - s_1 \cdot \left[(\bar{x}_1 - x)^2 + (\bar{y}_1 - y)^2 \right] \\
quad_2(x,y) &= k_2 - s_2 \cdot \left[(\bar{x}_2 - x)^2 + (\bar{y}_2 - y)^2 \right] \\
f_{MTQ}(x,y) &= \max\left(quad_1, quad_2\right)
\end{aligned}
$$

This parameterized landscape defines a a class of maximization problems with two peaks located at the points $(\bar{x}_1, \bar{y}_1)$ and $(\bar{x}_2, \bar{y}_2)$, respectively. The s_1 and s_2 parameters will be used to scale the two peaks, and the k_1 and k_2 constants are used to translate them up or down. We chose two landscapes from this family which have one global optimal peak, and one suboptimal local peak, as illustrated in Figure 1 and Figure 2. The parameter settings for landscape f_1 are $k_1 = 200$, $k_2 = 120$, $s_1 = 0.1$, and $s_2 = 0.03$. The parameter setting for f_2 are the same except that $s_1 = 1.2$.

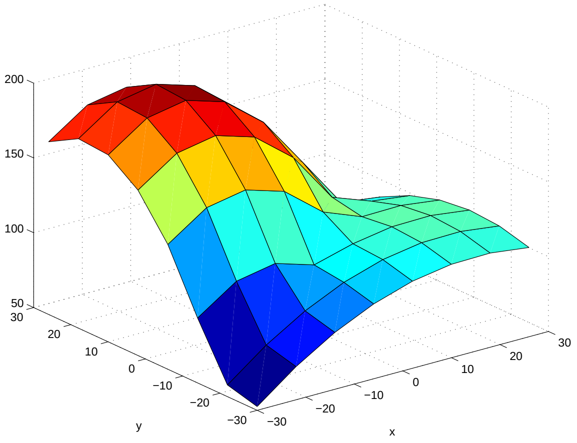

Fig. 1. 2D f_1 landscape

In order to apply our MPS model, the intervals $[-30, 30]$ must be discretized into a finite number of "genotypes". For illustration purposes, if we discretize the interval into six evenly spaced sample points, the result is a 6×6 payoff matrix in which the entries are simplly the function values at the sampled points. Table I illustrates this for landscape f_1.

In the game–theoretic framework, the genotypes from the x population to determine which row is of the payoff matrix will be used, and genotypes from the y population to determine which column of the payoff matrix will be used. For example, the payoff for the 4^{th} genotype from the x population interacting with the 2^{nd} genotype from the y population in the above example is 198.20, which also happens to be the global maximum of this payoff matrix.

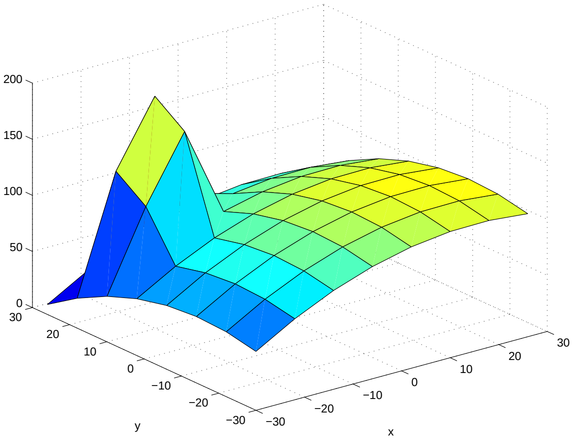

Fig. 2. 2D f_2 landscape

TABLE I

EXAMPLE 6×6 PAYOFF MATRIX FOR OUR f_1 OF THE MAX OF TWO QUADRATICS LANDSCAPES. THE DOMAINS FOR BOTH x AND y ARE $[-30.0, 30.0]$.

52.50	80.58	100.02	110.82	112.98	106.50
68.60	90.20	106.50	117.30	119.46	112.98
133.40	155.00	147.80	115.14	117.30	110.82
169.40	191.00	183.80	147.80	106.50	100.02
176.60	**198.20**	191.00	155.00	90.20	80.58
155.00	176.60	169.40	133.40	68.60	52.50

IV. ANALYZING THE MODEL

A. *Theoretical Analysis*

From a theoretical perspective, there are two ways one can pose questions about how EAs work. The first perspective focuses on the dynamical behavior of an EA in the limit and tries to identify fixed points to which populations converge (if such exist), and understanding basic dynamical properties of EAs near these points. The second perspective is more behavior–oriented. It asks the question: what areas of the space do trajectories traverse during the search. This case recognizes the pragmatics of how evolutionary algorithms are actually used in the sense that solutions are obtained by retaining the optimal point of the search space that trajectories *pass through*. In our study so far, we focus primarily on the former viewpoint and try to address questions about the stability of fixed points and the relative sizes of their basins of attraction; however, clearly EGT is a general framework which can be used to address both perspectives.

Fortunately, work in EGT in general, and MPS games specifically, has already addressed many of these questions analytically. For example, any *strict* Nash equilibria must contain only pure strategies (must be at the basis vectors, the corners of the simplexes) [7]. This means that in the absence of variational operators, in many cases we can expect the populations in our systems to converge to homogeneity. We also know, however, that mixed strategy equilibria are possible on the $bnd(\Delta^n \times \Delta^m)$ when the Nash points are not strict. This can happen when there are plateaus or ridges in the objective landscape, for instance.

Though not presented in this paper for reasons of size and scope, we have proved some useful properties about discrete time MPS models [11]. For instance, fixed points at the basis vectors of the product simplexes which are associated with strictly maximal values in the payoff matrix are stable, while those associated with the minimum are purely unstable. Those fixed points on the basis vectors associated with values which are less than the maximum value of their column or row in the matrix, but greater than the minimum are unstable, saddle points.

However, knowing the stability of a fixed point in a system does not necessarily tell you whether it is more or less likely to be reached by any arbitrary initial condition, unless more is known about the dynamical system ([7]). The question of how *likely* is it that some initial point will move to a particular fixed point is really a question of the size of basins of attraction in the system.

The basin of attraction of a given fixed point (or, indeed, any limiting behavior) is the set of initial points that will eventually map to that point. The question posed in the last paragraph can now be rephrased: what is the measure of a fixed point's basin of attraction relative to the other fixed points to which trajectories go?

Our preliminary analysis shows that there is reason to believe that the size of the basins of attraction of a fixed point indicated by a basis vector has more to do with relative local column and row values, than how large the specific payoff value is at that point. In other words, this may mean there is reason to suggest that broad, suboptimal peaks will pull trajectories away from taller, more narrow peaks. This is a form of local convergence that has been recognized in CEAs [4].

The proofs and mathematical evidence for these and other properties of these models are currently being refined for publication [11]. However, this paper will provide an initial set of model validation studies which seek to measure the relative sizes of the basins of attraction of payoff landscapes which have varying sized peaks as a means of illustration.

B. Empirical Analysis

Measuring the sizes of the basins of attraction of all the various limiting behaviors of a dynamical system is far from easy. First of all, there is no guarantee that there is any analytical way to do so in general. Second, it is generally difficult to definitively *know* all the possible limiting behaviors, much less measure their basins. Moreover, the dimensionality of the spaces of the systems we are interested in our very large, so even if we restrict ourselves to fixed points, and assume there are no cyclical or chaotic orbits, the space of potential attractors (in general) may be quite large.

In our case though, there are three things we have or can do to make this a more tractable problem. First, we actually *do know* some useful things about the systems we are studying which help us. For instance, we know that as long as the maximum values on the rows and columns are unique, the only strict Nash equilibria are at the basis vectors. Second, knowing this, we construct our problem such that this property is true of our payoff matrix. Finally, as an initial validation study method, we use an empirical method in order to perform this measure.

In fact, the method we are using is called a *rain gauge measure* and the idea is quite simple. An initial point is selected uniformly at random from the product simplexes, a trajectory through the space is computed using the initial point and the system model by iterating the system some large number of times, then we look at the limiting behavior. In our case all trajectories move to the basis vector, so we maintain a histogram corresponding to these points. If the trajectory seems to have converged ”very close” to a particular basis vector, we increment its value in the histogram. We then repeat this process some large number of times.

While the condition ”very close” is somewhat qualitative, and in *general* may not be sufficient (e.g., unstable points will push points that are ”very close” away), we *can* be more comfortable with this choice if an observed trajectory approaches a known stable fixed point (which you will soon see is true in this case). Trajectories were run for 5000 iterates, or until they were ”very close” meaning within a delta of 10^{-4} in terms of variational distance. All iterates met this condition.

We chose the two example functions (f_1 and f_2) in order to illustrate how this technique can be used to help understand some of the dynamics of these systems. For these simple studies, we choose to use an 8×8 sized payoff matrix, representationally equivalant to a 6 bit GA. As illustrated in figure 1, the first function, f_1, has roughly the same domain area under each quadratic peak, that is about 55% of the domain values achieve a maximum with $quad_1$, and about 45% with $quad_2$. The purpose of the second landscape is to show what happens when the peaks remain the same relative difference in height, but the area *under* each peak is changed. For f_2, less than 10% of the payoff entries are due to the $quad_1$ peak (see figure 2).

The results for the first function are not surprising. We chose 5000 different initial points for the system, and every point mapped to the basis vector associated with the global peak. An alternate way of describing this is that, when we choose an initial starting state for the population at random, the model always indicated that a CCEA algorithm would converge to homogeneity at the global maximum of first function. The second function is a more interesting example. In this case, the exact same initial points are used, but now only 48.2% of the initial points mapped to the global peak, the remaining 52.8% converged to the local, $quad_l$ peak. The measure of the basins of attraction of all the other basis vector fixed points was zero.

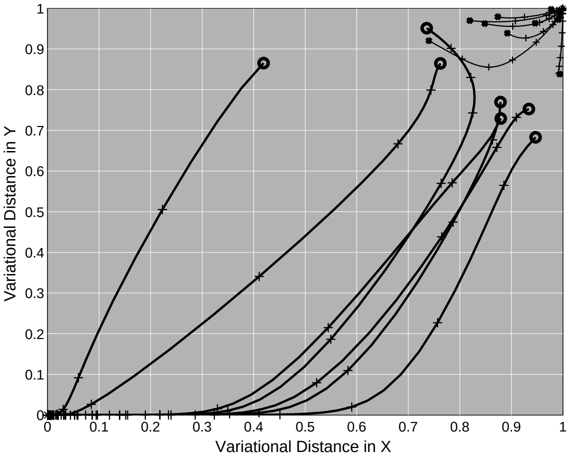

Fig. 3. MPS trajectories on f_2. Open circles represent initial points for trajectories converging to global maximum. Closed circles represent initial points bound for smaller local maximum.

Getting insight into whether or not there are topological properties governing the behaviors of the trajectories is difficult due to the high dimensionality of the space. In the case of function two, f_2, it is our supposition that trajectories starting closer to the global peak in the domain space are drawn towards it, while those which start farther are drawn to the suboptimal peak. One way to try to answer this is to look at the *variational distance* of points in the trajectories from the basis vector indicating the global peak, $\delta_x = \max(abs(\vec{x} - \vec{x}_g))$ and $\delta_y = \max(abs(\vec{y} - \vec{y}_g))$. Figure 3 illustrates this for f_2. Here it seems that some minimal amount of initial probability in the components associated with the global peak is necessary to converge to the global optimum. This confirms our understanding that cumulative payoff values local to some suboptimal maxima can distract trajectories from finding the global peak when they are sufficiently large.

V. CONCLUSIONS

We believe EGT will help us understand the complicated dynamics of coevolution, and in particular address specific questions about how and when they can be used best. In order to illustrate this point, we have introduced a framework for formal analysis of cooperative coevolutionary algorithms using MPS games from evolutionary game theory.

We focused on cooperative coevolutionary function optimizers to show how one might begin to address questions about how well CEAs perform on optimization tasks since CCEAs have a natural interpretation as an optimizer. In particular, we showed how one can use MPS games to obtain a more formal understanding of the behavior of CCEA optimizer.

Our early formal analysis of CCEAs suggests that we can expect stability from unique maximum values, but the condition for this stability is local with respect to the payoff matrix columns and rows. However, we still do not know how likely trajectories are to go to those stable points.

We believe that insight into the limit behavior of trajectories can be obtained from empirical analysis of the formal models. We introduce one such method (a rain gauge technique) for getting a measure of the relative sizes of the basins of attraction of simplex corner fixed points. Knowing that trajectories move either to the global or local peak, we constructed a graph of the variational distances in the $\vec{x}$ and $\vec{y}$ populations vectors. This graph shows that initial points close to the global optimal go to that fixed point, but points farther away typically do not. Thus, by demonstrating how these methods can be used on two example landscapes, we uncover the interesting fact that the discrete time model may have local convergence issues.

These empirical methods give us hope that there are analytical properties which may be found to help us understand such issues. Our goal in the future is to use the formalism of MPS to better understand CCEAs by identifying the formal properties which govern this behavior, and to discover the role variation operators play. We are also considering formalisms which do not require the assumptions of infinite populations and complete mixing. Finally, we are are interested in merging this research with our component analysis research on applied CCEAs. The result should be a clearer picture for how and when practitioners can apply cooperative coevolution.

REFERENCES

[1] C. Rosin and R. Belew. New methods for competetive coevolution. *Evolutionary Computation*, 5(1):1–29, 1997.

[2] M. Potter and K. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.

[3] S. Ficici and J. Pollack. Challenges in coevolutionary learning: Arms–race dynamics, open–endedness, and mediocre stable states. In Adami et al, editor, *Proceedings of the Sixth International Conference on Artificial Life*, Cambridge, MA, 1998. MIT Press.

[4] S. Ficici and J. Pollack. A game-theoretic approach to the simple coevolutionary algorithm. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo, and H.-P. Schwefel, editors, *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, pages 467–476. Springer-Verlag, 2000.

[5] J. Weibull. *Evolutionary game theory*. MIT Press, Cambridge, MA, 1992.

[6] Kathleen Alligood, Tim Sauer, and James Yorke. *Chaos: An Introduction to Dynamical Systems*. Springer–Verlag, 1996.

[7] Josef Hofbauer and Karl Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.

[8] John Maynard-Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.

[9] M. Vose. *The Simple Genetic Algorithm*. MIT Press, 1999.

[10] M. Potter and K. De Jong. A cooperative coevolutionary approach to function optimization. In Y. Davidor and H.-P. Schwefel, editors, *Proceedings of the Third International Conference on Parallel Problem Solving from Nature (PPSN III)*, pages 249–257. Springer-Verlag, 1994.

[11] R. Paul Wiegand, William Liles, and Kenneth De Jong. Multi–population symmetric game dynamics, 2001. In preparation.