Biasing Coevolutionary Search for Optimal Multiagent Behaviors

Liviu Panait[1], Sean Luke[1], and R. Paul Wiegand[2]
[1] Department of Computer Science, George Mason University, Fairfax, VA 22030
[2] Naval Research Laboratory, Washington, DC 20375
lpanait@cs.gmu.edu, sean@cs.gmu.edu, paul@tesseract.org

Abstract— **Cooperative coevolutionary algorithms offer great potential for concurrent multiagent learning domains and are of special utility to domains involving teams of multiple agents. Unfortunately, they also exhibit pathologies resulting from their game-theoretic nature, and these pathologies interfere with finding solutions that correspond to optimal collaborations of interacting agents. We address this problem by biasing a cooperative coevolutionary algorithm in such a way that the fitness of an individual is based partly on the result of interactions with other individuals (as is usual), and partly on an estimate of the best possible reward for that individual if partnered with its optimal collaborator. We justify this idea using existing theoretical models of a relevant subclass of coevolutionary algorithms, demonstrate how to apply biasing in a way that is robust with respect to parameterization, and provide some experimental evidence to validate the biasing approach. We show that it is possible to bias coevolutionary methods to better search for optimal multiagent behaviors.**

Index Terms— **Coevolution, cooperative coevolution, biased coevolution, multiagent learning, optimal collaboration, multi-population symmetric coevolution.**

I. INTRODUCTION

COEVOLUTIONARY ALGORITHMS (CEAs) are popular augmentations of traditional evolutionary algorithms (EAs). The basic elements of these augmentations lay in the adaptive nature of fitness evaluation in coevolutionary systems: individuals are assigned fitness values based on interactions with other individuals. Of particular interest are CEAs that have a kind of compositional nature that lend themselves toward learning behaviors for multiple, interacting agents. The most obvious CEAs for such problems are so-called cooperative[1] coevolutionary algorithms, which are typically formulated such that multiple interacting individuals succeed or fail together as a team.

One would imagine that cooperative coevolutionary algorithms may be effective for problems with certain structural properties among interacting components such as those problems that can be decomposed appropriately to limit particular types of nonlinear relationships [2], [3]. The intuition behind this advantage is that the algorithm searches only projections of the space at any time (one projection per population), thus reducing the search in a given generation from one exponentially large joint space to multiple simpler subspaces. Each decomposed subproblem may be cast as a projection of the (joint) problem space. Unfortunately, a great deal of information is discarded when basing an individual's fitness only on a projection of the joint space. One consequence of this is that a population's estimate of this projection is strongly influenced by the particular makeup of other populations, and this makeup is largely out of the original population's control. The result is that it is easy for a poor sample set to mislead the algorithm about the search space as a whole. This often leads to a preference for the kind of individual that partners well with a broad range of individuals from the other populations, whether or not that individual can form a globally optimal partnership.

The obvious countermeasure would be to bias the algorithm to seek optimal collaborations. In this paper we explore this option. We justify this idea by using existing theoretical models of certain cooperative coevolutionary algorithms, show how to apply biasing in a way that is robust with respect to parameterization, and provide some experimental evidence to validate the biasing approach. We show that it is possible to improve coevolutionary search for optimal multiagent behaviors using a biasing method.

Section 2 provides a brief background in multiagent learning and compositional approaches to multipopulation coevolution, as well as a dynamical systems approach for analyzing these types of algorithms. Here we describe precisely the subset of cooperative coevolutionary algorithms of interest to us, which we term *multipopulation symmetric coevolutionary algorithms* (or MPS-CEAs). Section 3 then proposes our approach to explicitly incorporating bias into MPS-CEAs, and provides a limited theoretical justification for our approach. In Section 4, we introduce a novel visualization method for observing basins of attraction in a multi-dimensional space, which allows us to demonstrate the efficacy of this biasing. In Section 5, we show that our biasing method is highly sensitive to a key parameter, and propose a modification of the method to mitigate this sensitivity. In Section 6, we provide experimental results for the application of our biased cooperative coevolutionary algorithm using a simple rote learning method for developing the bias over the run of the algorithm. We show that biasing enhances both a traditional MPS-CEA algorithm, as well as spatially-embedded MPS-CEAs. We complete the paper with a discussion of our conclusions and future work.

---

[1] The term 'cooperative' is problematic for reasons we discuss later in the paper. A better general term might be 'compositional' [1].

II. BACKGROUND

A. *Concurrent Multiagent Learning*

A large portion of multiagent research may be divided neatly according to whether or not it involves multiple learners [4]. Some research applies a single learner to improve the performance of an entire team of agents (for example, [5], [6], [7]). Other research applies separate learning processes to individual agents, while still assessing the quality of the agents as a team. The material in this paper concerns these latter approaches, which we will term *concurrent multiagent learning*.

The primary advantage of concurrent multiagent learning is that it projects the large joint team search space onto separate, smaller, component search spaces. If the problem can be decomposed such that individual agent behaviors are relatively disjoint, then this can result in a dramatic reduction in search space. A second, related advantage is that breaking the learning process into smaller chunks permits more flexibility in the use of computational resources because these chunks may, at least partly, be learned independently of one another.

Coordination games are standard problem domains for concurrent multiagent learning. A coordination game consists of a series of stages. At each stage, each agent is asked to choose an action without knowing which actions the other agents will choose. After each agent has decided on an action, they all perform their actions concurrently. A reward is then given to all agents, its value depending on their joint action. In some games, the agents may observe the other agents' actions upon receiving the reward.

Claus and Boutilier [8] introduced two simple benchmark coordination games, *Climb* and *Penalty*. The domains are difficult because of the penalties associated with miscoordinated actions and the presence of suboptimal collaborations that avoid such penalties. The joint payoff matrices for the two 3×3 coordination games are presented in Table I. If Agent 1 performs action 1 and Agent 2 performs action 2, we will refer to the joint action with the form $(1,2)$..

Agents in the Climb domain receive maximum payoff when they both choose action 1. However, the joint reward matrix has a second equilibrium: when both agents choose action 2. This is a suboptimal equilibrium, because the team reward is lower than at $(1,1)$. The Penalty domain has two optimal equilibria, at $(1,1)$ and $(3,3)$. However, there are significant penalties associated with agents choosing $(1,3)$ and $(3,1)$. Additionally, $(2,2)$ is a third equilibrium. This point is suboptimal, but there is no penalty for miscoordination if only one agent chooses action 2.

Claus and Boutilier [8] showed that a concurrent reinforcement learning algorithm is not guaranteed to find the optimal team behaviors for these games, even in the case when agents are able to observe other agents' actions. The authors then suggested that the search could be improved by using more optimistic exploration actions; this direction was further explored in [9], [10]. We will use the Climb and Penalty games later in this paper as two of our multiagent learning testbeds.

TABLE I

EXAMPLES OF COORDINATION GAME PAYOFF MATRICES: (A) THE 3X3 CLIMB GAME; (B) THE 3X3 PENALTY GAME. BOTH AGENTS RECEIVE THE SAME PAYOFF FOR A GIVEN JOINT ACTION, AS SHOWN IN THE MATRIX. PAYOFFS MARKED 'PENALTY' ARE FILLED WITH SOME NEGATIVE NUMBER DEPENDING ON THE EXPERIMENT.

		Agent 2		
		1	2	3
Agent 1	1	11	penalty	0
	2	penalty	7	6
	3	0	0	5

(A)

		Agent 2		
		1	2	3
Agent 1	1	10	0	penalty
	2	0	2	0
	3	penalty	0	10

(B)

B. *Cooperative Coevolution*

Cooperative coevolutionary systems are often a good fit for concurrent multiagent learning problems. When agents are working together as a team to perform some joint task, it is often very natural to represent individual behaviors separately and evolve multiple populations of such agents working in collaboration with one another. For examples of such cooperative coevolutionary algorithms, see Husbands and Mill [11] and Potter and De Jong [12].

Here we assume the use of the architecture defined by Potter [13]. The cooperative coevolutionary algorithm works as follows. Each population is assigned to search for a component of the solution (e.g., an agent in a multiagent team), and individuals represent candidate solutions for such components. One member from each population is needed in order to assemble a complete solution (e.g., the team). Evaluation of an individual from a particular population is performed by assembling that individual with collaborating individuals from other populations and testing the resultant full solution. To combat noise in the evaluation process due to choice of collaborators, multiple evaluations are usually performed, each with different collaborator sets. An individual's fitness could be the maximum (or the minimum or the average) over such evaluations, among other approaches. The effects of different evaluation schemes on performance are studied in [3], [14], [15], [16]. Aside from evaluation, the populations are evolved independently. Applications of this method include optimization of inventory control systems [17], learning constructive neural networks [18], multiagent systems [19], [20], [21], and rule learning [22], [23].

An example may serve to clarify this process. Suppose we are optimizing a three-argument function $f(x,y,z)$. Here, x, y and z represent possible strategies for three agents, and f is a function that indicates the performance of the team. One might assign individuals in the first population to represent the x argument, the second to represent y, and the third to represent z. Each population is evolved separately, except that when evaluating an individual in some population (e.g., x), collaborating individuals are chosen from the other populations (y and z) in order to obtain an objective function value with a complete solution, $f(x,y,z)$. This function value is the payoff that the evaluated individual (in this case x) receives. An individual's fitness is computed from the combination of payoffs from one or more evaluations with various sets of collaborators.

One simple method is to choose collaborators by using the most-fit individual from each of those populations as determined by the previous round of evaluations. Another approach is to pick collaborators at random from the other populations. Once a complete solution is formed, it can be evaluated and the resulting score can be assigned to the individual. We also assume that learning in the populations is performed concurrently. That is, all populations advance to the next generation at the same time.

Though historically the term "cooperative" has been applied to such algorithms, "cooperative coevolution" is a confusing and ambiguous term, and as noted in [24], it is used inconsistently in the literature. The term has been variously applied to describe the emergent dynamics of systems [25], properties of the underlying payoff of the problem [26], and the compositional approach of the design engineer [13]. All these cases lack a clear definition for the term.

For this reason, for the rest of this paper we focus on a specific, well-defined subclass of cooperative coevolutionary algorithms that we call a *multipopulation symmetric coevolutionary algorithm* (MPS-CEA). Such algorithms are *symmetric* in a sense similar to "symmetric games" in game theory. When a particular individual is being evaluated as part of a particular collaboration (a set of components, a team, a solution, etc.), the individual will receive some payoff. In a symmetric cooperative coevolutionary algorithm, if that same collaboration is used to evaluate one of the other component individuals in the set, the same payoff will be awarded. That is, a given $f(x,y,z)$ is the same value regardless of whether x, y, or z is the payoff recipient.

In a traditional EA, when solutions have a genetic basis for their expressed behavior, increasing the explorative powers of the genetic operators in order to expand the search of some part of the space comes at the cost of increased destruction of the existing learned genetic material. In the MPS-CEAs, the decomposition of the solution in a sense protects some components of the solution while the search is being performed on a given component, thus allowing the operator to focus its search on each component separately [27]. This kind of *a priori* partitioning may have severe consequences if the true problem decomposition is poorly matched with the representational decomposition [3], [28], [29]. However, a well-matched decomposition may still perform admirably even with significant non-linear interaction among components [2]. Consequently, MPS-CEAs may well benefit multiagent learning even when dependencies exist among agents [2], [3].

C. Modeling Multipopulation Symmetric Coevolution

An appealing abstract mathematical model for multipopulation symmetric coevolution comes from the biology literature: evolutionary game theory (EGT) [30], [31]. EGT provides a formalism based on traditional game theory and dynamical systems techniques to analyze the limiting behaviors of interacting populations under long-term evolution. EGT has been previously applied to the analysis of single population competitive coevolution [32], [33], [34] and multipopulation symmetric coevolutionary algorithms [26], [35]. To a certain degree, the EGT model bears some similarity to Markov models for coevolutionary systems [36], [37], [15], [38].

In this paper, we consider only two-population models. In such models, a common way of expressing the rewards from individual interactions is through a pair of *payoff matrices*. Since we are concentrating solely on symmetric CEAs, we may assume that when individuals from the first population interact with individuals from the second, one payoff matrix A is used, while individuals from the second population receive rewards defined by the transpose of this matrix (A^T). In our theoretical exploration of EGT in this paper, we will use an *infinite population*: thus a population can be thought of not as a set of individuals, but rather as a finite-length vector $\vec{x}$ of proportions, where each element in the vector is the proportion of a given individual configuration (popularly known as a *genotype* or, as we will term it, a *strategy*) in the population. As the proportions in a valid vector must sum to one, all legal vectors make up what is commonly known as the *unit simplex*, denoted Δ^n, where n here is the number of distinct genotypes possible, $\vec{x} \in \Delta^n : x_i \in [0,1], \sum_{i=1}^n x_i = 1$. In a two-population model, the domain space of the system is a Cartesian product of two such simplexes, $\Delta^n \times \Delta^m$.

Formally we can model the effects of evaluation and proportional selection over time using a pair of difference equations, one for each population. The proportion vectors for the two populations are $\vec{x}$ and $\vec{y}$ respectively. Neglecting the issue of mutation and breeding and concentrating only on the effects of selection, we can define the dynamical system of a two-population symmetric coevolutionary algorithm as:

$$\vec{u} = A\vec{y} \tag{1}$$

$$\vec{w} = A^T\vec{x} \tag{2}$$

$$x_i' = \left(\frac{u_i}{\vec{x}\cdot A\vec{y}}\right)x_i \tag{3}$$

$$y_i' = \left(\frac{w_i}{\vec{y}\cdot A^T\vec{x}}\right)y_i \tag{4}$$

where $\vec{x}'$ and $\vec{y}'$ represent the new population distributions for the next generation. Here it is assumed that an individual's fitness is the mean payoff over pairwise collaborations with *every* member of the cooperating population. We call this *complete mixing*. The equations above describe a two-step process. First, the vectors $\vec{u}$ and $\vec{w}$ are derived; these represent the fitness assessments of individuals in the generations $\vec{x}$ and $\vec{y}$ respectively. Then selection is performed by computing the proportion of the fitness of a specific individual over the sum fitness of the entire population.

Using such a model, one may begin to answer questions about where long-term system trajectories will go. In dynamical systems parlance, this translates to questions about the existence of fixed points in the system, the stability of those points, and the basins of attraction that map to them. A fair amount is known formally about the fixed points and their stability [26]. For example, in the absence of variation, all basis vector points are fixed points, though most are unstable. Pure Nash equilibria are stable, attracting fixed points in the system. Since a Nash equilibrium does not necessarily imply global optimality, there may exist stable fixed points at basis vectors associated with suboptimal payoff values. This means that the number of stable attracting points is linearly bounded by the number of strategies available to one of the populations. The semantic interpretation of these facts becomes clearer in the context of performance questions about the algorithm when something is known about the basins of attraction of relevant fixed points. Unfortunately, such knowledge is difficult to come by analytically, and most research on the properties of the basins of attraction have been empirical in nature. The visualization of trajectories in our formal model (Section IV) falls in this empirical vein.

D. The Relative Overgeneralization Pathology

Coevolution can exhibit a number of pathologies such as *loss of gradient* [39] and *focusing* [40]. The focus of our paper is another prominent pathology called *relative overgeneralization* [24]. In this pathology, an MPS-CEA can prefer not optimal individuals but rather "jack-of-all-trades" individuals that dovetail nicely with most of the *current individuals from the other population*. While relative overgeneralization could suggest that MPS-CEAs might inherently favor robust solutions, it also means that MPS-CEAs are *not* necessarily optimizers in the sense that one might intuitively expect them to be. This is grim news for practitioners wanting to coevolve "optimal" (or perhaps, even "good") cooperative strategies using a coevolutionary algorithm — what might be referred to as the *optimal collaborations*.

The dynamics of relative overgeneralization may be described in this way. When applying coevolution to coordination problems, the goal of practitioners is usually to find the optimal joint strategy, that is, the set of strategies, one for each agent, that yields the highest payoff. This point is a stable attracting fixed point of the coevolutionary system, as well as a Nash equilibrium. However, the search space may contain additional Nash equilibria with possibly lower payoff. For example, the coordination game in Table I (B) has two Nash equilibria: the optimal (1,1) point with a payoff of 11, and the suboptimal (2,2) point with a lower payoff of 7. Unfortunately, it is possible that most, if not all, learning trajectories may be pulled toward the suboptimal points because relative overgeneralization produces large attractive "bowls" around the suboptima.

Here we should be clear: this pathology is *not the same* as the typical local convergence problems that plague many heuristic methods, nor is it due to stochastic sampling errors. A genetic algorithm, under the same theoretical conditions including infinite populations, *will* be attracted to parts of the space associated with a unique maximum. The relative overgeneralization pathology says that, even in the very idyllic conditions of infinite populations and no genetic operations of any kind, MPS-CEAs are *still* not necessarily attracted to the optimum.

E. Memory Mechanisms

Attempts to augment CEAs to address their many challenges include a wide range of methods, often employing certain types of memory mechanisms. In competitive coevolution, explicit memory methods have been used to retain outstanding opponents [41], to keep track of currently known Nash equilibria [42], and to maintain as many informatively distinguishing test cases as possible [43]. Additionally, similar kinds of explicit memory methods have been recently applied to cooperative coevolution in an attempt to address the relative overgeneralization pathology via Pareto dominance [44]. Our biasing technique, when applied to "realistic" applications as discussed later, also uses a memory mechanism.

In addition to these direct approaches, memory can be implicitly added to an evolutionary system by embedding the populations into a spatial geometry. Researchers in the field of evolutionary computation have studied these models in some detail [45], [46], [47]; however, until recently analysis of coevolutionary spatial models has focused primarily on general CEA performance measures [48]. Nevertheless, applications of spatially distributed coevolutionary systems have proven effective [49], [50], [51], sometimes demonstrating clear advantages over non-spatial CEAs [52]. Still, the underlying reasons for these advantages have only begun to be explored [53], [54], and though there is no reason to believe they specifically address relative overgeneralization, there is evidence that they may help maintain adaptive gradients for certain kinds of problems. Since memory is an important element in our approach, though, we will experiment with such spatial models later in the paper in order to provide a contrast between these explicit and implicit memory mechanisms.

III. BIASING TOWARD OPTIMAL COLLABORATION

How might the MPS-CEA be modified such that it is more suitable for optimization tasks? Our approach to solving the relative overgeneralization issue is to bias the search by computing an individual's fitness based on two components: its immediate reward while interacting with individuals in the population, and a *heuristic estimate* for the reward it would have received had it interacted with its optimal collaborators. The first part of this reward will be called the *underlying objective function*, and the second will be called the *optimal collaborator estimate*. We will use δ to denote the degree of emphasis the fitness places on the optimal collaborator estimate.

We note that this notion of bias toward maximum possible reward has also been used in literature in subtly different ways than we use it here. Maximum reward has been used in multiagent reinforcement learning by [8], [9], [10]. To some extent, the "Hall of Fame" method introduced by [41] for competitive coevolution is also related to biased coevolution; however, that technique samples randomly in the Hall of Fame to increase robustness, while we tend to deterministically select the ideal partners.

Although several experiments in this paper use an exact computation for the optimal collaborator estimate, we refer to a *heuristic estimate* because in practice it is highly unlikely that the algorithm will be able to easily compute the *actual* ideal collaborators. We envision a variety of approaches to computing a heuristic estimate. The estimate might be based on partnering with the most successful collaborators known so far in the population; or with collaborators chosen (or constructed) based on the success they have had with individuals that are structurally "similar" to the test individual; or with collaborators chosen based on past history with the individual's ancestors; and so forth. We reserve comparison of such approaches to future work. Here we will concentrate on the foundations of the approach itself.

Our first, and perhaps the most obvious, approach to biasing is a modified fitness assessment method that is simply a weighted sum of the result of the collaborations and the result of the estimated maximum projection. Equation 5 below describes this idea mathematically. Here the fitness of argument x_a is being assessed by combining the result of the underlying objective function, g, with the optimal collaborator estimate, g'_a.

$$\begin{aligned} f(x_1,\ldots,x_a,\ldots,x_k) = {} & (1-\delta)\cdot g(x_1,\ldots,x_a,\ldots,x_k) \\ & + \delta \cdot g'_a(x_a) \end{aligned} \tag{5}$$

At one extreme, when $\delta = 1$, the algorithm will trust only the estimate, and the states of the other populations are entirely irrelevant. It is no longer a coevolutionary system at all; there are rather k EAs searching the k-projected component spaces independently in parallel. At the other extreme, when $\delta = 0$, the algorithm trusts only the underlying objective function. This is the traditional MPS-CEA.

Using this definition of δ, we may now modify the dynamical system model; however, the specifics of the modification depend on how evaluation with "typical" collaborators is performed. One simple approach is to use as "typical collaboration" the evaluation model applied previously. Recall we stated that if an individual's fitness is based on its immediate interaction with individuals from the other population, then $\vec{u} = A\vec{y}$ and $\vec{w} = A^T\vec{x}$, as described in Equations 1 and 2. Now, let us consider a function $maxrow(A)$ that returns a column vector corresponding to the maximum value of each row in matrix A. If an individual's fitness is based on its maximum possible performance in conjunction with any individual from the other population, then we may modify equations 1 and 2 to be $\vec{u} = (1-\delta)A\vec{y} + \delta\, maxrow(A)$ and $\vec{w} = (1-\delta)A^T\vec{x} + \delta\, maxrow(A^T)$.

In this modified system, the tendency to optimize performance is clear when $\delta = 1$ and there is a unique global optimum. At each iteration of the model, the fitness of each component will be its best possible fitness. If there is a unique global maximum, its components will have the highest fitness in each population, and so the proportion of the corresponding components will increase in the next step. When there are multiple global maxima, setting $\delta = 1$ is not necessarily a good choice because it provides no incentive for the joint populations as a whole to converge to a single solution. Furthermore, setting $\delta = 1$ may place too much faith on an inaccurate heuristic estimate for the optimal collaborators. When δ is set appropriately, however, biasing can have a considerable (positive) effect, as we will demonstrate next.

A crucial issue remains: what are the effects of choice of δ? As it turns out, naive approaches to defining δ can result in high sensitivity to the exact value of δ. This is a serious problem if (as would usually be the case) the experimenter does not know the best value of δ beforehand, or chooses to adjust it dynamically during the run. In Section V we will construct a class of problem domains intended to demonstrate this sensitivity, propose an alternative biasing mechanism that does not have this sensitivity problem.

IV. VISUAL DEMONSTRATION OF BIASING BENEFITS

In [55] we developed a novel approach to visualizing the degree to which collaborative techniques will tend to converge to various equilibria. We now use this technique to argue for the benefit of biasing. We do so by augmenting a common MPS-CEA fitness-assessment method (the maximum return over N collaborations), and compare this augmentation against this fitness-assessment method alone.

A. Maximum of N Collaborations

The maximum of N collaborations method is a common way to perform fitness assessment in an MPS-CEA. The idea is to evaluate an individual not with one collaborator, but with some N collaborators, and return the maximum reward obtained. In some sense, this method might be seen as a kind of bias toward collaboration with more optimal partners. But as we will see, it alone may not be as effective as when augmented with the proposed biasing method (or indeed as effective as the proposed biasing method alone).

The evolutionary game theory model used here assumes that an individual's fitness is the average of its rewards obtained when involved in all possible collaborations with individuals from the other populations. We may modify this definition of fitness to reflect the maximum of N collaborations method as follows:

Theorem 1: Let the payoff for individual i when teamed with individual j be a_{ij}, and $(p_j)_{j\in 1..n}$ be the probability distribution for the individuals in the population of collaborators for i (here, n is the number of distinct genotypes in the population of collaborators). If the a_{ij} values are sorted in increasing order ($a_{i1} \leq a_{i2} \leq .. \leq a_{in}$), the expected maximum payoff of i over N pairwise combinations with random collaborators $j_1 \ldots j_N$ chosen with replacement from the other population is $\sum_{j=1}^{n} a_{ij}\left(\left(\sum_{k=1}^{j} p_k\right)^N - \left(\sum_{k=1}^{j-1} p_k\right)^N\right)$.

Proof: We appeal to order statistics. The expected maximum payoff is a linear combination of the actual payoff a_{ij} times the probability that it is the maximum of pairwise combinations with N random collaborators. As the a_{ij} values are sorted, then the probability that a collaborator is chosen with payoff $\leq a_{ij}$ is simply $\sum_{k=1}^{j} p_k$. Likewise, the probability that a collaborator is chosen with payoff $< a_{ij}$ is $\sum_{k=1}^{j-1} p_k$. Thus the probability that *all* N collaborators have payoff $\leq a_{ij}$ is $\left(\sum_{k=1}^{j} p_k\right)^N$ and the probability that all N collaborators have payoff $< a_{ij}$ is $\left(\sum_{k=1}^{j-1} p_k\right)^N$.

Now, the probability that j is the maximum of N collaborators is the probability that the collaborators are chosen only from the individuals $1 \ldots j$ minus the probability that they are chosen from that set excluding j, that is, $1 \ldots (j-1)$. This is $\left(\sum_{k=1}^{j} p_k\right)^N - \left(\sum_{k=1}^{j-1} p_k\right)^N$. Thus the expected maximum payoff is $\sum_{j=1}^{n} a_{ij}\left(\left(\sum_{k=1}^{j} p_k\right)^N - \left(\sum_{k=1}^{j-1} p_k\right)^N\right)$. ■

Unfortunately, there are two reasons why this method alone is insufficient to correct the relative overgeneralization pathology. First, in order to increase the (supposed) bias, one must increase the number of collaborators chosen, which will have severe computational consequences (particularly when the number of populations is large). Second, though this type of collaboration method certainly biases the *current* search sample, there is no retained memory of particularly good collaborations that were identified previously. The chief difficulty remains: the algorithm searches only a projection of the problem at a time and that projection is always changing — we are merely biasing how we look at that projection.

B. *Visualizing the Basins of Attraction*

We use a visualization method to show that selecting the maximum of N collaborators will not always properly bias the system toward optimal collaboration, and that augmenting with explicit biasing helps considerably. To do so, we employ the EGT model with expected maximum fitness as described in Sections II and III, using fitness proportionate selection and no variation operators. We iterate the model until the proportion of one of the genotypes in each population exceeds a threshold of 0.99995, or until 50000 generations. Given the initial configuration, EGT models the coevolutionary search as a deterministic process. That is, for each initial point in the search space, we can approximate the equilibrium to which it converges. As [24] shows, the populations are expected to converge to pure Nash equilibria in the payoff matrix (elements that are maximal on their row and column).

The visualization approach we will take is detailed in [55]. In summary, we will present a 2D graph of white, gray, and black points, representing convergence to the various Nash equillibria of the Climb and Penalty games. In the Climb game, the white points indicate convergence to the optimal equilibrium, and the black points likewise indicate the suboptimal equilibrium (gray points indicate those few places where convergence did not occur within 50000 iterations of the model). In the Penalty game, the white and gray points are the two optimal equilibria, and the black points are the suboptimal equilibria. In both cases, the crucial desired feature is a decrease in black points.

Each axis of the 2D graph is the simplex of one or the other population. We have employed a special ordering of this simplex in order to cluster the points around the relevant Nash equilibria to make the graph more clear. Complete details on this ordering may be found in [55]. In summary, the simplex is divided into 3 areas: the initial populations containing more 1s, more 2s, and more 3s, respectively. Each such area is then allocated a segment in the projection: the first one-third contains results for initial populations with more 1s, while the last one-third of the projection contains results for initial populations with more 3s. Then, each area is divided in two such that, when projected, the degenerated populations containing a single genotype appear in the middle of the segment. For example: the top one-third area in the 2-D visualization refers to initial first populations with a majority of 1s, while the right-most one-third area refers to initial second populations with a majority of 3s.

C. *Visualization of Maximum of N Collaborations*

Figure 1(a) shows the basins of attraction for the Climb coordination game when using different numbers of collaborators in each population for the traditional MPS-CEA. The images show that the "deceptiveness" of the problem domain decreases as the number of collaborators is increased. When using a single collaborator, it appears that the coevolutionary search will find the global optimum if at least one of the populations starts with a large number of 1s. As the number of collaborators is increased we observe that the basin of attraction for the suboptimal equilibrium reduces to areas where at least one of the initial populations has a very large proportion of 2s or 3s: as more collaborators are used, the proportion required to converge to the suboptimum increases.

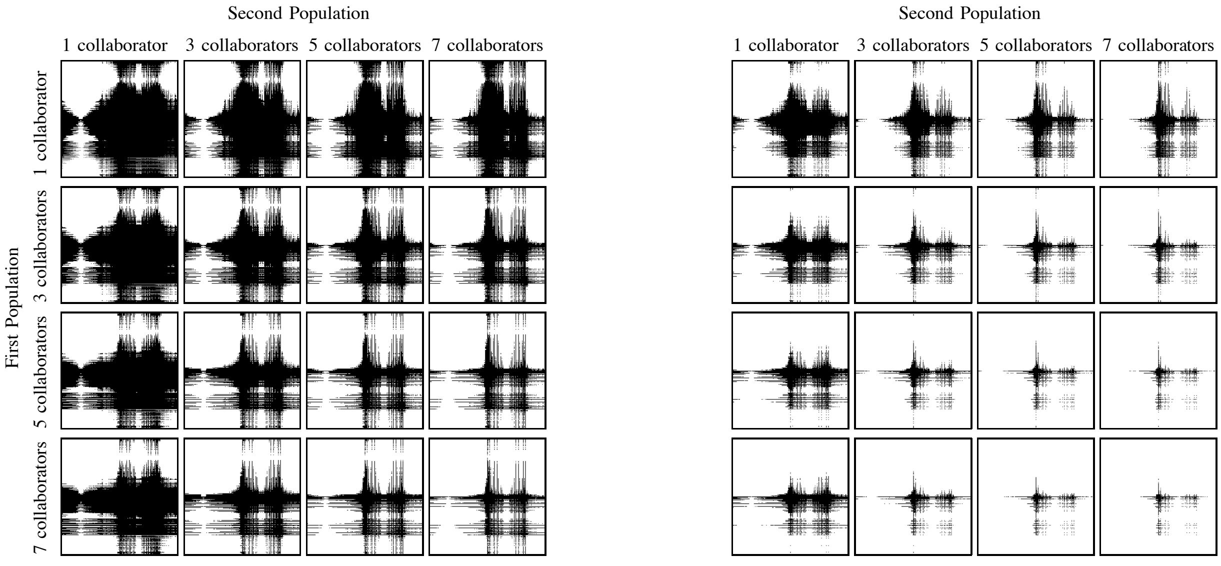
Figure 2(a) presents the basins of attraction for the Penalty game. We observe that the two global optima cover most of the space even when a single collaborator is used; the suboptimal equilibria covers mainly areas where at least one of the population started with a high percentage of 2s, and the other population has 1s and 3s equally distributed — this increases the percentage of miscoordinations. As the number of collaborators is increased, the basin of attraction for the (2,2) point reduces to areas where *both* populations start with almost all 2s.

D. *Visualization of Maximum of N Collaborations Augmented with a Biased MPS-CEA*

Using N collaborations alone does help reduce penalties due to miscoordination and helps the system find more optimal equilibria. However, augmenting this with a more explicit biasing can have a considerable (positive) effect on the resulting basins of attraction. Indeed, the augmented biased MPS-CEA using a very small N is superior to a traditional MPS-CEA using larger values for N.

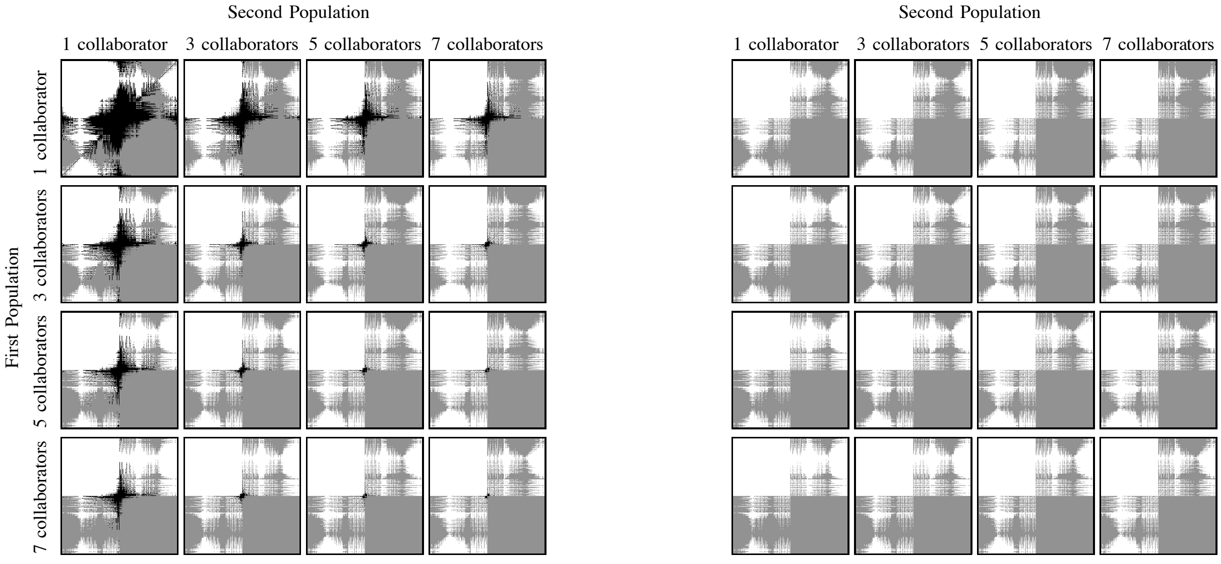
Our augmented biased MPS-CEA works as follows. The fitness will be based partly on the maximum of N collaborations with randomly chosen partners (the underlying objective function) and partly on the the reward obtained when partnering with the optimal collaborator. We set δ to 0.5 — although we assume the optimal collaborator is known for any individual, setting δ to 1.0 would cause the biased MPS-CEA to converge to mixed equilibria in the Penalty domain due to the presence of two equal global optima.

Figures 1(b) and 2(b) show the basins of attraction for the equilibria for the biased algorithm. The figures strongly suggest that this augmentation of the best-of-N approach further reduces the basins of attraction for suboptimal equilibria, and when biasing, increasing the number of collaborators helps even further. In fact, in the Penalty domain, the basins of attraction for the two globally optimal equilibria cover the entire space, even with a single collaborator.



(a) Traditional Coevolution **(b) Biased Coevolution**

Fig. 1. Basins of attraction in the Climb problem when using (a) traditional and (b) biased coevolution at $\delta = 0.5$, with 1, 3, 5, and 7 collaborators per population. White and black mark the basins of attraction for the (1,1) and (2,2) equilibria.



(a) Traditional Coevolution **(b) Biased Coevolution**

Fig. 2. Basins of attraction in the Penalty problem when using (a) traditional and (b) biased coevolution with $\delta = 0.5$, with 1, 3, 5, and 7 collaborators per population. White, black and gray mark basins of attraction for the (1,1), (2,2), and (3,3) equilibria.

V. ANALYSIS OF SENSITIVITY TO THE BIASING RATE

As we will show, the current formulation of the explicit biasing method is very sensitive to δ. This is important because it is not immediately apparent what value of δ the experimenter should use. Large amounts of bias may be unwise if optimal-collaborator estimates are poor. But depending on problem properties, small amounts of bias may have almost no effect. Further, certain system settings may decrease an algorithm's sensitivity to the degree of bias or exacerbate it. A smooth, relatively insensitive biasing procedure is necessary for the success of biased coevolution, as it mitigates radical, unexpected changes in the algorithm properties due to changes in choice of δ.

To examine this sensitivity to δ, we make two relatively straightforward and obvious simplifications. In Section VI we will then relax both simplifications to account for more realistic problems and algorithms. First, we consider only a static value for δ throughout a run and focus our attention on how different static values affect final runtime performance. Though a more realistic algorithm (such as in Section VI) would likely adjust the degree of bias dynamically throughout the run, it is unclear how best to do this. Second, we take a big step and assume that the biasing information (the optimal collaborator for each individual) is known *a priori*. In other words, we assume that the function g'_a is known beforehand. This simplification is made in order to reduce the variables involved in the experiment, and it is reasonable because there remain important sensitivity issues to consider even with this simplification.

The results of these experiments will lead us to propose an alternative approach to applying the bias. As we shall see, this alternative significantly reduces the sensitivity of the algorithm to the effects of changing the degree of bias.

A. Problem and Algorithm Properties

We begin by constructing the *maximum of two quadratics* (MTQ) class of problem domains, which can offer a range from simple to very difficult instances. MTQ is a class of two-dimensional functions loosely defined as $\max(f_1, f_2)$, where f_1 and f_2 are two quadratic polynomials. Such a class may of course be used for a variety of optimization problems in both traditional EAs and CEAs. If we view the x axis as representing strategies for one agent (and thus assign a population to coevolve it), and the y axis as representing the strategies for a second agent (and similarly assign it to another population), the MTQ class may be seen as defining a game payoff for an abstract multiagent problem.

The reason we construct the MTQ class is that these problems can generate payoff matrices that resolve or exacerbate the relative overgeneralization pathology of MPS-CEA by adjusting the relative contributions of joint rewards implicitly [24]. Another advantage of this particular problem class is that the maximum projection can be determined easily from the derivatives of the two underlying quadratic functions.

The MTQ class is defined as:

$$MTQ(x,y) \leftarrow \max \begin{cases} H_1 * \left(1 - \frac{16*(x-X_1)^2}{S_1} - \frac{16*(y-Y_1)^2}{S_1}\right) \\ H_2 * \left(1 - \frac{16*(x-X_2)^2}{S_2} - \frac{16*(y-Y_2)^2}{S_2}\right) \end{cases} \quad (6)$$

where x and y take values ranging between 0 and 1. Figure 3 illustrates some example MTQ problem instances. Different settings for H_1, H_2, X_1, Y_1, X_2, Y_2, S_1, and S_2 affect the difficulty of the problem domain in one of the following aspects:

a) Peak height: H_1 and H_2 affect the heights of the two peaks. Higher peaks may increase the chances that the algorithm converges there.

b) Peak coverage: S_1 and S_2 affect the area that the two peaks cover: a higher value for one of them result in a wider coverage of the specific peak. This makes it more probable that the coevolutionary search algorithm will converge to this peak, even though it may be suboptimal.

c) Peak relatedness: The values X_1, Y_1, X_2, and Y_2 affect the locations of the centers of the two quadratics, which in turn affect the relatedness of the two peaks: similar values of the x or y coordinates for the two centers imply higher overlaps of the projections along one or both axes.

Aside from the impact of the properties of the problem domain, our sensitivity study targets three algorithmic settings:

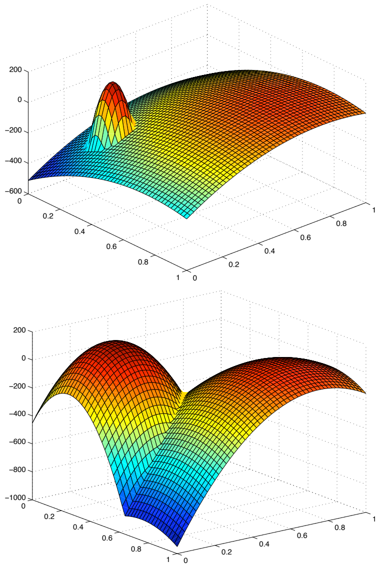


Fig. 3. Example *Maximum of Two Quadratics* problems, illustrating ($S_1 = 1.6, S_2 = 0.5$) and ($S_1 = 0.5, S_2 = 0.5$).

d) Biasing rate: We are interested in how performance degrades as the biasing rate δ changes. We set δ to various rates in the range $[0,1]$.

e) Population size: The population size affects how much the coevolutionary algorithm samples the search space. It is easy to show that coevolution using a bias rate δ of 1.0, combined with infinite populations and perfect knowledge of maximal projections, will converge to the unique optimum with probability 1. We expect similar results for large populations as well.

f) Collaboration scheme: The MPS-CEA algorithm tries to simplify the search process by decomposing the candidate solutions into components and coevolving them in separate populations. The only information such a population can get about the overall progress of the search process is through collaborators — samples that are usually representative of the status of the other populations. For some spaces, an increased number of collaborators may better capture the intricacies of the search space [14], [16].

Though the MTQ class is capable of generating challenging problems (with respect to the relative overgeneralization pathology), it lacks some characteristics that may have important effects on the performance of algorithms under more realistic conditions. For example, though there is some element of non-linearity present in the problem, this effect is produced by our somewhat artificial maximization of two, otherwise linearly separable quadratics. It is unclear the degree to which more realistic problems with truly non-linear relationships between represented components will have similar characteristics. However, the problem class does allow us very explicit control over a wide range of salient properties for our study (discussed below), and it *does* demonstrate a kind of coevolutionary deception (when broad suboptimal peaks are more attractive than narrow optima) [24].

B. Sensitivity Results

All experiments here used the MTQ class of problems. The coevolutionary search process used two populations, one for each variable. Each such population used a real-valued representation, with individuals constrained to values between 0 and 1 inclusive. Non-adaptive Gaussian mutation (mean 0 and standard deviation 0.05) was the only variational operator. Each population used tournament selection of size 2, and the best individual survived automatically to the next generation. The search lasted for 50 generations, after which the best individuals in each population were at, or very near, one of the two peaks. Each point in Figures 4–7 was computed over 250 independent runs. All experiments were performed with the ECJ system [56].

Unless stated otherwise, each population consisted of 32 individuals. The default collaboration scheme used two collaborators from each population: the best individual in the previous generation was always selected, and the other individual was chosen at random[2]. Our biasing method combines the *a priori* " fitness with the better of the results obtained when the individual was teamed with each of the two collaborators. The default values of the parameters for the first (suboptimal) peak were H_1=50, X_1=$\frac{1}{4}$, Y_1=$\frac{1}{4}$, and S_1=1.6. The second (optimal) peak was characterized by H_2=150, X_2=$\frac{3}{4}$, Y_2=$\frac{3}{4}$, and S_2=$\frac{1}{32}$. With these settings, the two peaks were nearly at opposite corners of the domain space.

1) Biasing and Domain Features: The first set of experiments investigated the relationship between the biasing rate and the three problem domain features described previously: the relative heights, coverages and locations of the peaks. There were 11 experimental groups for each property, one for each value of $\delta \in [0,1]$ in increments of 0.1. Figure 4 shows the mean final results of these 33 groups.

a) Peak height: We kept H_1 constant at 50, and set H_2 to 75, 150, and 300. The results indicated that less than 10% of runs converged optimally when the rate of biasing was low, while the ratio increased to greater than 90% when using high biasing rates. Unfortunately, there was no smooth transition between these two extremes: rather, small modifications to the biasing rate could change the rate of convergence to the optimum by as much as 70–80%. Moreover, the relative difference in peak height directly affected where these sudden jumps in performance appeared. This suggests that the algorithm may not only be quite sensitive to δ with respect to changes in relative peak height, but also suggests that it may be difficult to predict *where* the sudden transitions occur.

b) Peak coverage: S_2 was set to $\frac{1}{128}$, $\frac{1}{64}$, $\frac{1}{32}$, $\frac{1}{16}$, $\frac{1}{8}$, and $\frac{1}{4}$, while S_1 was constantly 1.6. Here, the location of the transition was more consistent among the various values, but the transitions themselves were still abrupt. It also appears that the relative peak coverages caused more variation in results when the bias rate was small, while the curves at the other extreme of the graph appeared close together. The results indicated that certain settings of peak coverage will affect the algorithm's sensitivity to the δ parameter: the wider the peak, the more gradual the transition when varying the bias rate.

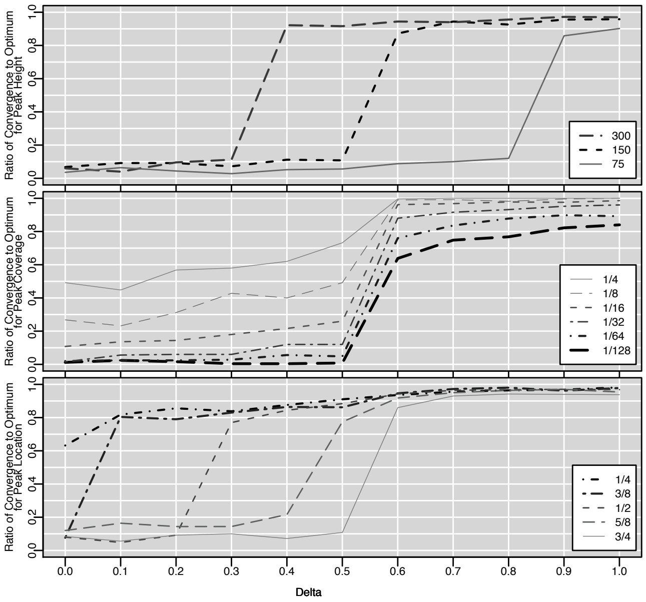


Fig. 4. Convergence ratios for peak height (top), peak coverage (center) and peak relatedness (bottom). x axis shows biasing rate δ, and y axis shows ratio of the 250 trials that converged to, or very near, the global optimum.

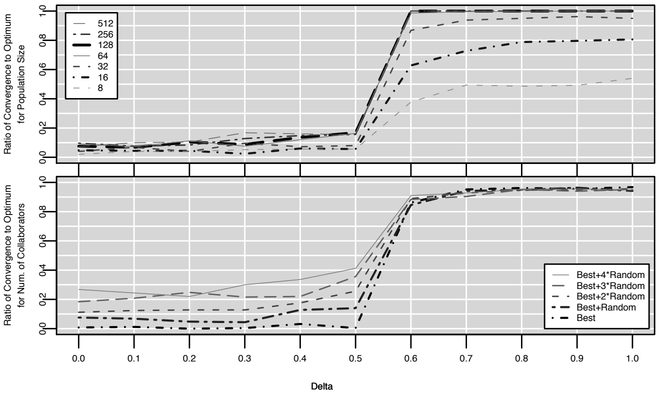


Fig. 5. Convergence ratios for population size (top) and collaboration scheme (bottom). x axis shows biasing rate δ, and y axis shows ratio of the 250 trials that converged to, or very near, the global optimum.

c) Peak relatedness: The Y_2 parameter was set to $\frac{1}{4}$, $\frac{3}{8}$, $\frac{1}{2}$, $\frac{5}{8}$, and $\frac{3}{4}$. These settings incrementally transitioned the relative peak positions from diagonally opposite locations to ones aligned along one axis. Similar to peak height, the peak relatedness had a significant effect on the ratio of runs that converged to the global optimum: the more related the peaks, the less biasing was required to assure good performance. However, the curves had an abrupt transition between lower and higher rates of convergence to the optimum. Moreover, the location of this transition depended on the actual degree of peak relatedness, which suggests that the algorithm may be highly sensitive to δ with respect to this parameter.

---

[2] To reduce noise in the evaluation process, the experiments in this section employed the same random collaborator for all individuals in the population at that generation. We later discovered that using different random collaborators for different individuals results in a slightly better performance due to better sampling of the search space — but sensitivity to δ was unaltered.

2) Biasing and Algorithm Features: A second set of experiments investigated the relationship between the biasing rate and the population size and collaboration scheme. Again, there were 11 groups for each of these two parameters corresponding to each of the δ settings. The results are presented in Figure 5.

a) Population size: We set the size of each of the two populations to 8, 16, 32, 64, 128, 256, and 512. As expected, extremely small populations were less likely to reach the optimum, even with high biasing rates. We observed the same abrupt shift in performance as we saw in the previous experiments. The results suggested that increasing the population size does not necessarily alleviate algorithm sensitivity to δ.

b) Collaboration scheme: The default setting in all previous experiments used two collaborators to evaluate the fitness of each individual: the best-performing individual from the other population in the previous generation, and also a random individual. To test sensitivity to this collaboration scheme, we varied the number of random individuals from 0 to 4; the best individual from each population in the previous generation was always used. Varying the number of collaborators presents a tradeoff between computational complexity and the efficiency of the algorithm [14]: more collaborators induce an increased computational complexity, but the performance of the search might also be significantly improved. The bottom graph in Figure 5 shows that the collaboration scheme had some influence over the performance of the algorithm at low biasing rates, but it had no effect when higher biasing rates were used. Again, the abrupt change in performance indicated that the algorithm could be highly sensitive to the δ parameter, regardless of the collaboration methodology.

The result of nearly all of these experiments is that the biased MPS-CEA we have so far described seems to be very sensitive to δ, and we cannot alleviate this by adjusting the algorithmic parameters. It would appear that finding a suitable value for this parameter in practice may be difficult as it is applied currently. Fortunately, this need not be the case.

C. An Alternative Stochastic Biasing Mechanism

To uncover a possible simple alternative that does not share this problem, recall that for larger differences in peak heights, a wider range of biasing rates resulted in a high ratio of convergence; however, when one peak was only slightly higher than the other, the range of high convergence ratios was much smaller. The transition was abrupt, but the location of the transition shifted depending on the peak height differences.

Our hypothesis is that this extreme sensitivity of the algorithm to the biasing method with respect to the relative peak heights is caused by the linear combination of the two fitness components: the fitness when teamed with collaborators, and the fitness when in combination with the optimal collaborator. The higher the optimal peak, the lower the bias rate it needs to dominate the other term. However, if one peak is slightly higher than the other, the algorithm requires more biasing to locate the optimum.

To counter this, we used a non-parametric comparison method which considers the relative order of two components rather than their exact values. The justification for this technique is similar to that of non-parametric selection methods such as tournament selection [57], rank selection [58], and truncation selection [59]. In our technique, each individual is assigned two fitnesses: the underlying objective one when combined with the collaborators from other populations, and another one indicating the performance of the individual when in combination with its optimal collaborator. When comparing two individuals, with probability δ we compare based on the first "fitness"; else we compare based on the second.

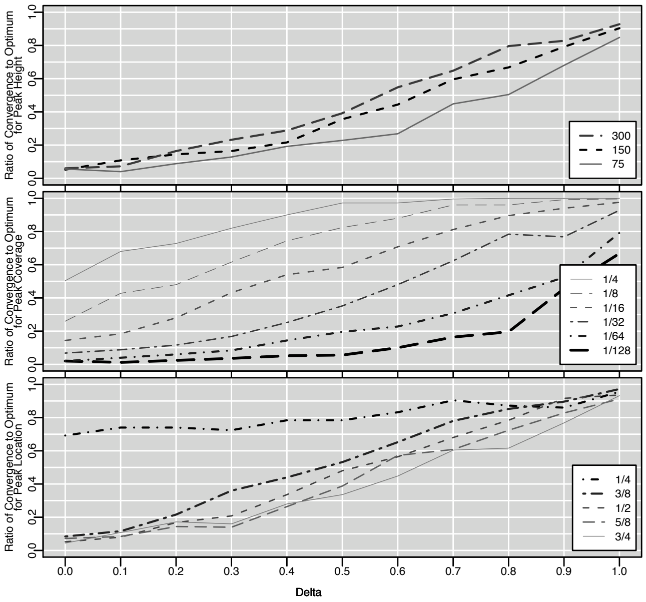


Fig. 6. Convergence ratios for probabilistic biasing when varying peak height (top), peak coverage (center) and peak relatedness (bottom). x axis shows biasing rate δ, and y axis shows ratio of the 250 trials that converged to, or very near, the global optimum.

We performed the same sensitivity analysis for the new algorithm. The results are presented in Figures 6 and 7. In all cases, the new algorithm does not exhibit the sudden jumps in performance as did the original. This suggests that it is an improvement resulting in significantly less sensitivity to the settings we have investigated.

Drawing from research in multiobjective optimization [60], [61], we experimented with some additional ways to counter the effects of linear combination . One alternative normalizes the two components before adding them; but our experiments using this mechanism still revealed abrupt transitions in performance. Another approach is to compare pairs of components based on Pareto dominance: one pair is better than another if both of its components are equivalent to or better than the corresponding components in the other pair, and at least one of its component is better than its corresponding component in the other pair. Our attempts to use Pareto dominance were again not successful at removing the abrupt transitions in performance as δ was changed.

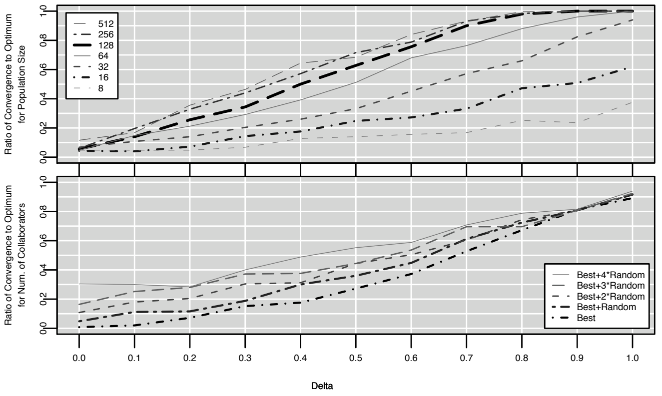


Fig. 7. Convergence ratios for probabilistic biasing when varying population size (top) and collaboration scheme (bottom). x axis shows biasing rate δ, and y axis shows ratio of the 250 trials that converged to, or very near, the global optimum.

VI. COMPARING REALISTIC IMPLEMENTATIONS OF TRADITIONAL AND BIASED MPS-CEAS

A. *Method of Study*

While the previous theoretical discussion, as well as the sensitivity analysis just discussed, helps justify our intuition for biasing the fitness evaluation, neither is immediately applicable to real problems. In a more realistic setting, simplifying model assumptions such as infinite populations, lack of variational operators, complete mixing, and *a priori* knowledge of the maximum payoff are not possible.

A previous simplifying assumption (*a priori* known biasing information) allowed us to keep the biasing rate constant during the coevolutionary search. To convert theory into practice, we adopted a rote learning algorithm for learning the biasing information. Specifically, if an individual selects action i, we assume its optimal collaborator picks that action j which *so far* has shown the highest performance when paired with i. As evolution progresses, the action chosen by the optimal collaborator changes to reflect the better (i,j) pairs that are evaluated. The main difference between the two representations in Sections VI-C and VI-D is primarily in how an individual chooses its actions.

We employed the stochastic biasing mechanism described in Section V-C. We decided on a simple proof-of-concept rule for updating the biasing rate. The algorithm started with $\delta = 1$, decreasing linearly until reaching $\delta = 0$ at 75% of the total number of generations, at which point it stayed at 0 until the end of the run. While our dynamic adjustment of δ was ad-hoc, it was a sufficient method to demonstrate our point in this context. Indeed, we will show that the biased coevolutionary algorithms outperformed their unbiased counterparts even with this ad-hoc setting.

We performed several experiments to compare traditional coevolution with biased coevolution in this context. We tested on the Climb and the Penalty coordination games introduced in Section II-A, and on a variation of MTQ which we call the Two Peaks domain, with a joint reward function of the form

$$f(x,y) = \max \begin{cases} 0 \\ 10 - 32 * ((x - \frac{1}{3})^2 + (y - \frac{1}{3})^2) \\ 15 - 128 * ((x-1)^2 + (y-1)^2) \end{cases}$$

with x and y taking values between 0 and 1. Finally, we tested the methods in a cooperative learning domain with increased non-linear interactions: the joint reward function was based on the two-dimensional Rosenbrock function

$$f(x,y) = -\left(100 * \left(x^2 - y\right)^2 + (1-x)^2\right)$$

x and y taking values between -2.048 to 2.048. For simplicity, we discretized each axis into 16, 32, 64, and 128 segments for experiments with both the Two Peaks and the Rosenbrock functions. Increased discretization resulted in larger search spaces, but not necessarily more difficult ones — when searching for pure strategies for the Rosenbrock domain (Table V), the rate of finding the global optima for all coevolutionary methods for 32 intervals is lower than that for 64 intervals. The experiments again used the ECJ software package [56]. In order to establish statistical significance, all claims of "worse" or "better" were verified using nonparametric tests. We used the Welch test (a variation the Student t-test that does not assume equal variance for the samples) repeatedly for pairs of samples. Given that the samples were rarely following a normal distribution, we first ranked the set of observations from both samples, then we performed the Welch test on those ranks. We also used the Bonferroni inequality to adjust the p-value level for each test such as to obtain 95% confidence over all comparisons; as a consequence, each Welch test was applied at a 99.95% confidence level.

B. *Competing Techniques*

We consider both biased and unbiased versions of three MPS-CEA algorithms. The first such algorithm is a "traditional" MPS-CEA. The others are two spatially-embedded MPS-CEAs similar to those discussed in Section II-E. We detail each of these next.

For the traditional MPS-CEA algorithm we chose a common approach to MPS coevolution fitness assessment: an individual was assessed twice to determine its fitness, once with a collaborator chosen at random and once partnered with the individual in the other population that had received the highest fitness in the previous generation. An individual's fitness was set to the maximum of these two assessments. This is termed *Traditional* in the remainder of this section.

In a spatially distributed MPS-CEA the individuals are positioned at specified locations in geometric space, such that a notion of a neighborhood exists among individuals. For consistency across small and moderate population sizes, we embedded each population in a one-dimensional ring. A neighborhood of radius 1 for an individual consisted of three individuals in this case: the specific individual, together with the individuals to its immediate left and right (on the ring). The spatial embedding of the populations influences the breeding process as follows: for each location, a number of individuals are selected with replacement from a local neighborhood (the radius of the neighborhood is detailed for each problem domain later), and the better ones are selected for breeding (the best individual is selected for mutation alone, or the better two individuals are selected for crossover, followed by mutation). When creating a child for location i, the parent at location i always competed for selection to breed.

The spatial embedding also influences the scheme to select collaborators. We experimented with two spatial collaboration schemes. First, we evaluated each individual with the unique collaborator from the other population that had the same location in space. We refer to this setting as *Spatial*. We doubled the population size for Spatial to allow it to have the same total number of evaluations as the other methods. A second spatially-embedded MPS-CEA algorithm evaluated each individual with two collaborators: the collaborator at the same location in space (as before), and a random collaborator from a small neighborhood (the radius of the neighborhood is detailed later). We term this second technique *Spatial2* for the remainder of this section.

The combination of biasing with each of the three algorithms is termed *Biased Traditional*, *Biased Spatial*, and *Biased Spatial2* respectively.

TABLE II

PERCENTAGE OF MPS-CEA RUNS THAT CONVERGED TO GLOBAL OPTIMUM, CLIMB DOMAIN WITH PURE STRATEGY REPRESENTATION

	Penalty			
	-30	-300	-3000	-30000
Traditional	56.1%	56.9%	56.8%	56.9%
Biased Traditional	79.9%	77.6%	80.8%	81.0%
Spatial	76.4%	79.7%	77.0%	77.2%
Biased Spatial	85.6%	88.2%	88.4%	87.0%
Spatial2	67.1%	69.8%	71.5%	70.0%
Biased Spatial2	82.4%	80.9%	81.7%	82.7%

TABLE III

PERCENTAGE OF MPS-CEA RUNS THAT CONVERGED TO GLOBAL OPTIMUM, PENALTY DOMAIN WITH PURE STRATEGY REPRESENTATION

	Penalty			
	-10	-100	-1000	-10000
Traditional	88.2%	89.4%	90.3%	88.4%
Biased Traditional	93.2%	93.5%	91.9%	93.4%
Spatial	99.3%	98.9%	99.3%	98.9%
Biased Spatial	99.7%	99.4%	99.3%	99.4%
Spatial2	93.5%	93.1%	94.8%	92.6%
Biased Spatial2	94.6%	95.8%	96.2%	94.2%

C. Searching for Pure Strategies

A first set of experiments encoded a single action (an integer) in each individual. In other words, each individual deterministically specified an action. In game-theory parlance, each individual thus represented a “pure strategy”. Such an individual bred children through mutation: the individual’s integer was increased or decreased (the direction chosen at random beforehand with probability 0.5) while a biased coin came up heads (with probability $\frac{1}{8}$ for Climb and Penalty, and with probability $\frac{1}{4}$ for Two Peaks and Rosenbrock). Evolutionary runs in the Climb and Penalty problem domain used only 3 individuals per population (Spatial used 6 individuals) and they lasted for 40 generations. Runs in the Two Peaks and Rosenbrock domains used 20 individuals per population (Spatial used 40) and they lasted for 200 generations. Spatial2 selected the second collaborator randomly using a neighborhood of radius 1. Selection for breeding used tournament selection with size 2. Parents were selected from neighborhoods of radius 1 using tournament selection with size 2 for each location in the spatially-embedded models. The most-fit individual survived automatically from one generation to the next in the non-spatially-embedded models.

Results Summary: The use of the proposed biasing mechanism usually resulted in statistically significant improvements in the rate of finding the global optima. In the few situations where biasing did not help, it did not hurt performance either. As a side-note, the Spatial algorithm consistently outperformed the traditional MPS-CEA.

Results Specifics: Tables II–V present the average percentage (out of 1000 runs) that converged to the global optimum. Overall, the spatial methods outperformed the traditional methods — not surprising, given the positive results in the literature as discussed in Section II-E — but the biased version of any method generally outperformed the unbiased version of that method. In the Climb domain, Spatial was significantly better than Traditional; Spatial was no better than Spatial2 for Penalty=-3000 at the properly adjusted confidence level of 99.95%, but it was superior using only 99.914% confidence. For all three methods, biasing significantly improved performance — Biased Spatial in particular converged to the global optima in about 90% of the runs, significantly better than all five other methods.

Spatial was better than both Traditional and Spatial2 in the Penalty domain. Except for significant improvements of Biased Traditional over Traditional when Penalty=-10 and Penalty=-10000, biasing was not effective at improving results at the 99.95% confidence level (though it did not damage results either). We performed three additional tests using all 4000 runs for each of the methods (1000 for each value of the penalty); the increased number of observations allowed us to establish that biasing was effective at significantly improving the performance of Traditional and (with only 99.89% confidence) Spatial2.

In the Two Peaks domain, Spatial was again better than Spatial2, which was better than Traditional. Enhancing the techniques with the proposed biasing mechanism resulted in significant improvements for Traditional (with only 99.4% confidence for 128 discretization level), and for Spatial2 (only for a discretization level of 8). All other differences were statistically insignificant.

In the Rosenbrock domain, Spatial was better than Traditional (with confidence level 99.95% for discretization level equals 128, and only with confidence levels 99.9% and 99.85% for discretization levels 8 and respectively 16) and Spatial2 (with confidence level 99.95% for discretization levels of 64 and 128, and only with confidence levels 99% for discretization level 8). Additional nonparametric tests using all

TABLE IV

Percentage of MPS-CEA runs that converged to global optimum, Two Peaks domain with pure strategy representation

	Discretization Level (Number of Actions)			
	16	32	64	128
Traditional	51.6%	50.8%	49.6%	49.0%
Biased Traditional	68.5%	65.8%	59.2%	59.4%
Spatial	86.5%	91.2%	89.3%	85.6%
Biased Spatial	84.1%	88.9%	87.9%	86.6%
Spatial2	72.0%	73.2%	69.3%	66.5%
Biased Spatial2	78.6%	74.7%	72.5%	68.4%

TABLE V

Percentage of MPS-CEA runs that converged to global optimum, Rosenbrock domain with pure strategy representation

	Discretization Level (Number of Actions)			
	16	32	64	128
Traditional	82.5%	33.6%	37.9%	16.5%
Biased Traditional	87.1%	48.8%	51.9%	21.9%
Spatial	84.3%	40.8%	45.4%	22.1%
Biased Spatial	86.6%	56.6%	82.0%	39.8%
Spatial2	78.2%	33.5%	41.2%	16.4%
Biased Spatial2	74.5%	42.0%	67.3%	22.8%

TABLE VI

Percentage of MPS-CEA runs that converged to global optimum, Climb domain with mixed strategy representation

	Penalty			
	-30	-300	-3000	-30000
Traditional	25%	20%	19%	21%
Biased Traditional	100%	100%	100%	100%
Spatial	67%	28%	27%	26%
Biased Spatial	100%	100%	100%	100%
Spatial2	50%	26%	25%	27%
Biased Spatial2	99%	99%	99%	99%

TABLE VII

Percentage of MPS-CEA runs that converged to global optimum, Penalty domain with mixed strategy representation

	Penalty			
	-10	-100	-1000	-10000
Traditional	100%	99%	99%	99%
Biased Traditional	100%	100%	100%	100%
Spatial	100%	99%	99%	98%
Biased Spatial	100%	100%	99%	99%
Spatial2	99%	99%	98%	98%
Biased Spatial2	100%	100%	100%	99%

4000 runs established that Spatial was significantly better than Traditional and Spatial2 with 99.95% confidence. The methods in combination with biasing usually performed better than alone; biasing never decreased significantly the performance of a method.

D. Searching for Mixed Strategies

Though using a pure strategy representation provides a clear connection to theory and emphasizes the problem properties in which we are interested, using such an encoding in these simple problems results in very small search spaces. It would be nice to consider larger problems with similar properties. We accomplished this by encoding a "mixed strategy" (to again use game theory parlance) in each individual. More specifically, individuals consisted now of a probability distribution over the available actions. When evaluating such individuals with a collaborator (another mixed strategy), 50 independent interactions were performed, each consisting of a joint action chosen at random according to the individuals' mixed strategies. The joint reward for the two individuals was computed as the average reward over the 50 joint rewards. Observe that using mixed strategies creates a potentially more difficult problem domain than using pure strategies for reasons of both search space size and the stochastic nature of the fitness result.

Results Summary: The results suggest that the mixed strategy representation induces a significantly more complex search space than the pure strategy representation: mixed strategies usually have a non-zero probability of exploring different actions that may incur penalties. For this reason, we argue that the slope around the optimal peak has an abrupt gradient that may explain the decrease in performance. Consistent with the previous experiments involving the pure strategy representation, the results indicate that biasing never decreases the performance of a method, but it rather improves it significantly in many domains.

Results Specifics: Traditional and Biased selected parents via tournament selection with size 2; breeding involved one-point crossover, followed by mutation by adding random Gaussian noise (mean 0 and standard deviation 0.25) with probability $\frac{1}{L}$ for each of the distribution values (where L is the number of actions in the problem domain), followed by renormalization of the distribution. We performed an extensive sensitivity study to set the parameters of the spatially-embedded coevolutionary algorithms. We found that lower mutation rates worked better (following crossover, we added Gaussian random noise to each gene with probability 0.2 for the Climb and Penalty domains, and only with probability $\frac{1}{3L}$ for Two Peaks and Rosenbrock). When using the Traditional and the Spatial2 methods, each population contained 20 individuals for Climb and Penalty, and 100 individuals for the Two Peaks and Rosenbrock domains (as noted, Spatial used twice the population size but an equivalent number of evaluations). The parents were selected using tournament selection with size 2 and with a neighborhood radius of 1 for Climb and Penalty. Given the larger population sizes for Two Peaks and Rosenbrock, parents were selected from neighborhoods of radius 3; the sensitivity study also indicated a tournament selection size of 5 for the Two Peaks domain, and of 3 for the Rosenbrock domain. Runs lasted for 200 generations in the Climb and Penalty domains, and for 1000 generations in the Two Peaks and Rosenbrock domains. We performed 1000 runs for each treatment to obtain statistical significance.

The mixed representation introduces an intriguing problem: what does the optimal collaborator for a mixed strategy look like, and how can it be learned? Our estimate for the optimal collaborator is done exactly as was done in the pure-strategy

TABLE VIII

PERCENTAGE OF MPS-CEA RUNS THAT CONVERGED TO GLOBAL OPTIMUM, TWO PEAKS DOMAIN WITH MIXED STRATEGY REPRESENTATION

	Discretization Level (Number of Actions)			
	16	32	64	128
Traditional	0%	0%	0%	0%
Biased Traditional	100%	100%	100%	100%
Spatial	0%	0%	0%	0%
Biased Spatial	100%	100%	100%	100%
Spatial2	0%	0%	0%	0%
Biased Spatial2	100%	100%	100%	100%

TABLE IX

PERCENTAGE OF MPS-CEA RUNS THAT CONVERGED TO GLOBAL OPTIMUM, ROSENBROCK DOMAIN WITH MIXED STRATEGY REPRESENTATION

	Discretization Level (Number of Actions)			
	16	32	64	128
Traditional	62%	12%	0%	0%
Biased Traditional	100%	100%	99%	82.9%
Spatial	93%	38%	3%	0%
Biased Spatial	100%	100%	100%	84.3%
Spatial2	85%	25%	1%	0%
Biased Spatial2	100%	100%	100%	79.8%

case: after selecting the action i (chosen from the individual's mixed strategy distribution), we then select j based on j's historical success when paired with i. To update this history information, we use only the first joint reward (of the total of 50) from each evaluation of a pair of individuals. To do otherwise would give the estimation procedure an undue advantage.[3]

Tables VI–IX present the percentages of runs that converged to the global optimum when using the mixed strategy representation in the Climb, Penalty, and the discretized Two Peaks and Rosenbrock domains. As the evaluation of an individual is averaged over 50 interactions, we considered that a run converged to the global optimum if the fitness of the best individuals (one per population) in the last generation was within 10% of the value of the global optimum — to exceed this threshold, each of the mixed strategies should have probability close to 1 for picking the action corresponding to the global optimum, as the joint reward for any other pair of actions was less than this threshold.

In the Climb domain, both Spatial and Spatial2 significantly outperformed Traditional. However, enhancing any of them with our biasing method resulted in convergence to the global optimum in almost every run. The Penalty domain was again easier then Climb — most runs found the global optimum.

The Two Peaks domain was consistently too difficult for either Traditional, Spatial and Spatial2, but all of them found the global optima in 100% of the runs when in combination with biasing. The Rosenbrock domain was relatively easier for coevolution, especially at low discretization levels. Traditional was again significantly worse than Spatial2, which in turn was significantly worse than Spatial. However, the performance of all methods was significantly superior when in combination with biasing.

VII. CONCLUSION

Coevolutionary algorithms offer great potential for concurrent multiagent learning domains. Their ability to focus on decomposed partitions of a larger, structured joint problem space make them very natural algorithms to consider for such problems. Despite this, pathologies resulting from the game-theoretic nature of CEAs, namely their propensity toward relative overgeneralization, interfere with finding solutions that correspond to optimal collaborations of interacting individuals. Some basic changes in the algorithm are necessary to correct this problem.

Our approach to address this problem was to alter the CEA such that the fitness of an individual was based partly on the result of interaction with other individuals, and partly on an estimate of the best possible reward for that individual if partnered with its optimal collaborator. This form of bias drew its inspiration from similar methods in reinforcement learning literature, and its justification from a limited theoretical analysis. We used a novel visualization method to help demonstrate the efficacy of the method from a theoretical viewpoint. Empirically, we explored the sensitivity of the method to the degree of bias, offering a mechanism to mitigate this sensitivity by probabilistically combining these two parts of the fitness evaluation. Finally, we provided early experimental evidence that our biasing method has merit, even as the biasing estimate is also learned during the search.

This work clearly reflects the early stages of studying biasing methods for coevolutionary algorithms. In more difficult problems, rote learning methods will be impractical and more complex learning methods must be employed to establish trustworthy and computationally efficient estimates of optimal collaboration. Moreover, we have provided no general understanding about how to adjust δ dynamically. Finally, a comprehensive investigation of when such methods are likely to succeed or fail has not, as yet, been undertaken. We intend to pursue each of these. Regardless, preliminary empirical evidence suggests that biasing CEAs toward optimal collaboration may be beneficial in practice.

ACKNOWLEDGMENTS

R. Paul Wiegand's participation in this work was made possible by the American Society of Engineering Education. The authors would like to thank Anthony Bucci, Edwin de Jong, Anthony Liekens, Elena Popovici, David Fogel, Xin Yao, and the anonymous reviewers for their helpful comments and suggestions. Some of the text and figures in this article previously appeared in earlier publications [55], [62] and has been reused with permission from Springer Science and Business Media.

REFERENCES

[1] R. Wiegand and M. Potter, "Robustness in compositional coevolution," in *Proceedings from the 2006 Genetic and Evolutionary Conference*. ACM Press, 2006.

---

[3] We also performed experiments using *all* 50 joint rewards to improve the optimal collaborator estimate, and the results improved further — all methods in combination with biasing found the global optimum in most cases.

[2] T. Jansen and R. P. Wiegand, "The cooperative coevolutionary (1+1) ea," *Evolutionary Computation*, vol. 12, no. 4, pp. 405–434, 2004.

[3] R. P. Wiegand, K. A. De Jong, and W. C. Liles, "The effects of representational bias on collaboration methods in cooperative coevolution," in *Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature (PPSN VII)*, J. J. *et al*. Merelo Guervós, Ed. Berlin, Germany: Springer, 2002, pp. 257–268.

[4] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387–434, 2005.

[5] S. Luke and L. Spector, "Evolving teamwork and coordination with genetic programming," in *Genetic Programming 1996: Proceedings of the First Annual Conference*, J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, Eds. Stanford University, CA, USA: MIT Press, 1996, pp. 150–156.

[6] A. Hara and T. Nagao, "Emergence of cooperative behavior using ADG; Automatically Defined Groups," in *Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO-99)*, 1999, pp. 1038–1046.

[7] K. Stanley, B. Bryant, and R. Miikkulainen, "Real-time neuroevolution in the NERO video game," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, pp. 653–668, 2005.

[8] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *Proceedings of National Conference on Artificial Intelligence (AAAI-98)*, 1998, pp. 746–752.

[9] M. Lauer and M. Riedmiller, "An algorithm for distributed reinforcement learning in cooperative multi-agent systems," in *Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kaufmann, 2000, pp. 535–542.

[10] S. Kapetanakis and D. Kudenko, "Reinforcement learning of coordination in cooperative multi-agent systems," in *Proceedings of the Eighteenth National Conference on Aritificial Intelligence (AAAI-02)*. MIT Press, 2002, pp. 326–331.

[11] P. Husbands and F. Mill, "Simulated coevolution as the mechanism for emergent planning and scheduling," in *Proceedings of the Fourth International Conference on Genetic Algorithms*, R. Belew and L. Booker, Eds. Morgan Kaufmann, 1991, pp. 264–270.

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

[13] M. Potter, "The design and analysis of a computational model of cooperative coevolution," Ph.D. dissertation, George Mason University, Fairfax, Virginia, 1997.

[14] R. P. Wiegand, W. Liles, and K. De Jong, "An empirical analysis of collaboration methods in cooperative coevolutionary algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2001*, L. *et al*. Spector, Ed. Morgan Kaufmann, 2001, pp. 1235–1242.

[15] L. Bull, "On coevolutionary genetic algorithms," *Soft Computing*, vol. 5, pp. 201–207, 2001.

[16] ——, "Evolutionary computing in multi-agent environments: Partners," in *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA)*, T. Bäck, Ed. Morgan Kaufmann, 1997, pp. 370–377.

[17] R. Eriksson and B. Olsson, "Cooperative coevolution in inventory control optimisation," in *Proceedings of the Third International Conference on Artificial Neural Networks and Genetic Algorithms*, G. Smith, N. Steele, and R. Albrecht, Eds. University of East Anglia, Norwich, UK: Springer, 1997, pp. 583–587.

[18] M. Potter and K. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evolutionary Computation*, vol. 8, no. 1, pp. 1–29, 2000.

[19] L. Bull and T. C. Fogarty, "Evolving Cooperative Communicating Classifier Systems," in *Proceedings of the Third Annual Conference on Evolutionary Programming*, A. V. Sebald and L. J. Fogel, Eds., 1994, pp. 308–315.

[20] L. Bull, T. C. Fogarty, and M. Snaith, "Evolution in Multi-agent Systems: Evolving Communicating Classifier Systems for Gait in a Quadrupedal Robot," in *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA)*, L. Eshelman, Ed. Morgan Kaufmann, 1995, pp. 382–388.

[21] M. Potter, L. Meeden, and A. Schultz, "Heterogeneity in the coevolved behaviors of mobile robots: The emergence of specialists," in *Proceedings of The Seventeenth International Conference on Artificial Intelligence (IJCAI-2001)*, 2001.

[22] M. Potter and K. De Jong, "The coevolution of antibodies for concept learning," in *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature (PPSN V)*, A. E. *et al*. Eiben, Ed. Springer, 1998, pp. 530–539.

[23] M. A. Potter, K. A. De Jong, and J. J. Grefenstette, "A coevolutionary approach to learning sequential decision rules," in *Proceedings from the Sixth International Conference on Genetic Algorithms*. Morgan Kaufmann, 1995, pp. 366–372.

[24] R. P. Wiegand, "An analysis of cooperative coevolutionary algorithms," Ph.D. dissertation, George Mason University, Fairfax, Virginia, 2004.

[25] R. Axelrod, *The Evolution of Cooperation*. New York: Basic Books, 1984.

[26] R. P. Wiegand, W. Liles, and K. De Jong, "Modeling variation in cooperative coevolution using evolutionary game theory," in *Foundations of Genetic Algorithms (FOGA) VII*, R. *et al*. Poli, Ed. Morgan Kaufmann, 2002, pp. 231–248.

[27] T. Jansen and R. P. Wiegand, "Exploring the explorative advantage of the CC (1+1) EA," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2003*, E. *et al*. Cantú-Paz, Ed. Berlin, Germany: Springer, 2003, pp. 310–321.

[28] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[29] S. Kauffman, *The Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press, 1993.

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

[31] J. Hofbauer and K. Sigmund, *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.

[32] G. Fogel, P. Andrews, and D. Fogel, "On the instability of evolutionary stable strategies in small populations," *Ecological Modeling*, vol. 109, pp. 283–294, 1998.

[33] S. Ficici and J. Pollack, "A game-theoretic approach to the simple coevolutionary algorithm," in *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, M. Schoenauer, Ed. Springer, 2000, pp. 467–476.

[34] S. Ficici, O. Melnik, and J. Pollack, "A game-theoretic and dynamical-systems analysis of selection methods in coevolution," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, pp. 580–602, 2005.

[35] R. P. Wiegand, W. Liles, and K. De Jong, "Analyzing cooperative coevolution with evolutionary game theory," in *Proceedings of CEC 2002*, D. Fogel, Ed. IEEE Press, 2002, pp. 1600–1605.

[36] A. Liekens, "Evolution of finite populations in dynamic environment," Ph.D. dissertation, Technische Universiteit Eindhoven, Netherlands, 2005.

[37] L. M. Schmitt, "Theory of coevolutionary genetic algorithms," in *Parallel and Distributed Processing and Applications, International Symposium*, M. Guo and L. T. Yang, Eds. Berlin: Springer, 2003, pp. 285–293.

[38] M. Chang, K. Ohkura, K. Ueda, and M. Sugiyama, "Modeling coevolutionary genetic algorithms on two-bit landscapes: Random partnering," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2004*, R. *et al*. Poli, Ed. Berlin, Germany: Springer, 2004, pp. 513–524.

[39] R. Watson and J. Pollack, "Coevolutionary dynamics in a minimal substrate," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2001*, L. *et al*. Spector, Ed. Morgan Kaufmann, 2001, pp. 702–709.

[40] A. Bucci and J. B. Pollack, "Focusing versus intransitivity geometrical aspects of co-evolution," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2003*, E. *et al*. Cantú-Paz, Ed. Berlin, Germany: Springer, 2003, pp. 250–261.

[41] C. Rosin and R. Belew, "New methods for competitive coevolution," *Evolutionary Computation*, vol. 5, no. 1, pp. 1–29, 1997.

[42] S. Ficici and J. Pollack, "A game-theoretic memory mechanism for coevolution," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2003*, E. *et al*. Cantú-Paz, Ed. Berlin, Germany: Springer, 2003, pp. 286–297.

[43] E. de Jong and J. Pollack, "Ideal evaluation from coevolution," *Evolutionary Computation*, vol. 12, no. 2, pp. 159–192, 2004.

[44] A. Bucci and J. Pollack, "On identifying global optima in cooperative coevolution," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2005*, H. *et al*. Beyer, Ed. New York, New York: ACM Press, 2004, pp. 539–544.

[45] J. Sarma, "An analysis of decentralized and spatially distributed genetic algorithms," Ph.D. dissertation, George Mason University, Fairfax, Virginia, 1998.

[46] M. Giacobini, E. Alba, and M. Tomassini, “Selection intensity in asynchronous cellular evolutionary algorithms,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2003*, E. *et al*. Cantú-Paz, Ed. Berlin, Germany: Springer, 2003, pp. 955–966.

[47] B. Dorronsoro, E. Alba, M. Giacobini, and M. Tomassini, “The influence of grid shape and asynchronicity on cellular evolutionary algorithms,” in *Proceedings of CEC 2004*, G. *et al*. Greenwood, Ed. IEEE Press, 2004, pp. 2152–2158.

[48] D. Cliff and G. F. Miller, “Tracking the red queen: Measurements of adaptive progress in co–evolutionary sumulations,” in *Proceedings of the Third European Conference on Artificial Life*. Springer–Verlag, 1995, pp. 200–218.

[49] D. Hillis, “Co-evolving parasites improve simulated evolution as an optimization procedure,” *Artificial Life II, SFI Studies in the Sciences of Complexity*, vol. 10, pp. 313–324, 1991.

[50] P. Husbands, “Distributed coevolutionary genetic algorithms for multi–criteria and multi-constraint optimisation,” in *Evolutionary Computing, AISB Workshop for Selected Papers*. Springer–Verlag, 1994, pp. 150–165.

[51] A. Ronge and M. G. Nordahl, “Genetic programs and co-evolution developing robust general purpose controllers using local mating in two dimensional populations,” in *Parallel Problem Solving from Nature IV, Proceedings of the International Conference on Evolutionary Computation*, ser. LNCS, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds., vol. 1141. Berlin, Germany: Springer Verlag, 1996, pp. 81–90.

[52] L. Pagie, “Information integration in evolutionary processes,” Ph.D. dissertation, Universiteit Utrecht, Netherlands, 1999.

[53] R. P. Wiegand and J. Sarma, “Spatial embedding and loss of gradient in cooperative coevolutionary algorithms,” in *Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, H. P. S. *et al*. Schwefel, Ed. Berlin, Germany: Springer, 2004, pp. 912–922.

[54] N. Williams and M. Mitchell, “Investigating the success of spatial co-evolution,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2005*, H. *et al*. Beyer, Ed. New York, New York: ACM Press, 2004, pp. 523–530.

[55] L. Panait, R. P. Wiegand, and S. Luke, “A visual demonstration of convergence properties of cooperative coevolution,” in *Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, H. P. S. *et al*. Schwefel, Ed. Berlin, Germany: Springer, 2004, pp. 892–901.

[56] S. Luke, ECJ 9: A Java EC research system. http://www.cs.umd.edu/projects/plus/ec/ecj/, 2002.

[57] D. Goldberg and K. Deb, “A comparative analysis of selection schemes used in genetic algorithms,” in *Foundations of Genetic Algorithms (FOGA)*, G. Rawlins, Ed. Morgan Kaufmann, 1990, pp. 69–93.

[58] D. Whitley, “The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best,” in *Proceedings of the Third International Conference on Genetic Algorithms*, J. D. Schaffer, Ed. San Mateo, CA: Morgan Kaufman, 1989, pp. 116–121.

[59] H. Mühlenbein and D. Schlierkamp-Voosen, “Predictive models for the breeder genetic algorithm,” *Evolutionary Computation*, vol. 1, no. 1, pp. 25–49, 1993.

[60] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[61] Y. Sawaragi, H. Nakayama, and T. Tanino, “Theory of multiobjective optimization,” *Mathematics in Science and Engineering*, vol. 176, 1985.

[62] L. Panait, R. P. Wiegand, and S. Luke, “A sensitivity analysis of a cooperative coevolutionary algorithm biased for optimization,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2004*, R. *et al*. Poli, Ed. Berlin, Germany: Springer, 2004, p. (to appear).

[63] R. *et al*. Poli, Ed., *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2004*. Berlin, Germany: Springer, 2004.

[64] H. *et al*. Beyer, Ed., *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2005*. New York, New York: ACM Press, 2004.

[65] L. *et al*. Spector, Ed., *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2001*. Morgan Kaufmann, 2001.

[66] E. *et al*. Cantú-Paz, Ed., *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2003*. Berlin, Germany: Springer, 2003.

[67] H. P. S. *et al*. Schwefel, Ed., *Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature (PPSN VIII)*. Berlin, Germany: Springer, 2004.

Liviu Panait received an M.Sc. degree in Computer Science from George Mason University in 2002, and is currently working toward his Ph.D. His research interests include evolutionary computation, reinforcement learning, multiagent systems, computer games, and artificial life.

Liviu Panait co-chaired the AAMAS 2006 workshop on Adaptation and Learning in Autonomous Agents and Multiagent Systems, co-organized the AAAI 2005 Fall Symposium on Coevolutionary and Coadaptive Systems, served as a program committee member or as an invited reviewer for several international conferences and journals, and he is a member of the IEEE Task Force on Coevolution. He is a co-author of the ECJ evolutionary computation library and the MASON multi-agent simulation toolkit.

Sean Luke is an assistant professor in the Department of Computer Science, George Mason University. He received his Ph.D. in Computer Science from the University of Maryland in 2000. Sean Luke also served as a visiting research scientist at Sony Computer Science Laboratories, Tokyo, Japan. Sean Luke is the author of the ECJ evolutionary computation system and the MASON multi-agent simulation toolkit. His research interests include evolutionary computation, genetic programming, machine learning of neural networks and finite-state automata, coevolution, multiagent simulation, and swarm robotics.

R. Paul Wiegand received his Ph.D. from George Mason University in 2004, on the topic of *An Analysis of Cooperative Coevolutionary Algorithms*. He currently works for ITT Industries Advanced Engineering & Sciences and conducts research at the Naval Center for Applied Research in Artificial Intelligence, part of the Naval Research Laboratory in Washington, DC. His general research interests are in the areas of theory of coevolutionary computation and application of coevolution for multiagent learning.

Paul Wiegand helped organize the coevolution workshop at the 2002 Genetic and Evolutionary Computation Conference and served as an invited researcher at Dortmund University as part of the Collaborative Research Center “Computational Intelligence” (SFB 531) in 2003. He has also served as co-moderator of EC-Digest, a research-oriented email list that serves the evolutionary computation community; and as a member of the IEEE Task Force on Coevolution.