New Methods for Competitive Coevolution

Christopher D. Rosin   Richard K. Belew

{crosin,rik}@cs.ucsd.edu
Cognitive Computer Science Research Group
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0114

Technical Report #CS96-491

Abstract

We consider "competitive coevolution," in which fitness is based on direct competition among individuals selected from two independently evolving populations of "hosts" and "parasites." Competitive coevolution can lead to an "arms race," in which the two populations reciprocally drive one another to increasing levels of performance and complexity. We use the games of Nim and 3-D Tic-Tac-Toe as test problems to explore three new techniques in competitive coevolution. "Competitive fitness sharing" changes the way fitness is measured, "shared sampling" provides a method for selecting a strong, diverse set of parasites, and the "hall of fame" encourages arms races by saving good individuals from prior generations. We provide several different motivations for these methods, and mathematical insights into their use. Experimental comparisons are done, and a detailed analysis of these experiments is presented in terms of testing issues, diversity, extinction, arms race progress measurements, and drift.

1 Introduction

Coevolution refers to the simultaneous evolution of two or more populations with coupled fitness. In this paper we consider "competitive coevolution," in which the fitness of an individual in one population is based on direct competition with some individual(s) from another population. Such interactions have been hypothesized to occur in nature, and modelled by game-theoretic constructions such as Maynard Smith's "evolutionary stable strategies," (Maynard Smith, 1982) and the Prisoners' Dilemma (Axelrod, 1989). It also arises in the evolution of AI game-playing strategies where the range of potential opponents makes it difficult to establish a single, fixed, "exogenous" fitness function as is typically used in genetic algorithms (GAs) (Angeline & Pollack, 1993; Mitchell & Forrest, 1994). More generally, a number of situations arise in complex software engineering applications where the construction of appropriate test suites to demonstrate the reliability of software is considered almost as important as the development of the software itself. By viewing the software solutions as one population and the test suites as another, competitive coevolution allows the simultaneous search for both (Hillis, 1991).

Individuals from each of the coevolving populations must take turns testing and being tested. In this paper we use the term *host* to refer to the individual whose fitness is under consideration, and *parasites* to refer to the individuals that are testing the host (this terminology differs somewhat from Hillis'). In all experiments done here, two evolutionarily distinct populations (i.e. no exchange of genetic material between populations) are used, with competitions taking place between populations. Each population gets its turn as both hosts (when its fitness is being measured) and parasites (when it is being used to measure fitness).

Since the parasites are also evolving with a fitness based on a competition's outcome, the success of a host implies failure for its parasites. When the parasites evolve to overcome this failure, they create new challenges for the hosts; the continuation of this may lead to an evolutionary "arms race" (Dawkins & Krebs, 1979). New genotypes arise to defeat old ones. New parasite types drive further innovation, creating ever-greater levels of complexity and performance by forcing hosts to respond to a wider range of more challenging parasite test cases.

As is common in software engineering, the problem of testing candidate solutions can be considered separately from that of developing good solutions. A perfect solution to the testing problem is a minimal and complete set of extremely difficult test cases, and is typically very difficult to identify for realistic problems. Note, however, that such a test set may well make a poor parasitic population, even if we knew what it was. Hosts forced immediately to compete against such rigorous testing are all likely to fare poorly. Evolution requires information from fitness *variation* to guide search. Coevolution may provide this, too. In any generation, the parasites used will have varying degrees of evolutionary success, but none are likely to be far beyond the difficulty required to defeat the current hosts. As the generations progress, we can expect a drive for more and more difficult test cases to arise. The result should be a *pedagogical* series of parasites that gradually increase in difficulty as host solutions gradually increase in competence.

In this paper, new methods for competitive coevolution are explored in the context of game-learning. Nim and 3-D Tic-Tac-Toe are used as test problems. Three new techniques in competitive coevolution are explored. "Competitive fitness sharing" changes the way fitness is measured, "shared sampling" provides a method for selecting a strong, diverse set of parasites, and the "hall of fame" encourages arms races by saving good individuals from prior generations. We provide several different motivations for these methods, and mathematical insights into their use. Experimental comparisons are done, and a detailed analysis of these experiments is done in terms of testing issues, diversity, extinction, arms race progress measurements, and drift.

A preliminary version of this work was presented in an earlier conference paper (Rosin & Belew, 1995). There, competitive fitness sharing and shared sampling were introduced and tested on Nim and 2-D Tic-Tac-Toe, with runs analyzed in terms of diversity and extinction. In this paper, we expand on these results with a new technique (the hall of fame) and new methods of analysis. Since 2-D Tic-Tac-Toe was found to be a fairly easy problem, we do not present results on it here. Instead, we add runs on the more complex 3-D Tic-Tac-Toe problem. We also give new theoretical results.

2 Motivation

2.1 Goals of Competitive Coevolution

As described above, an implementation of competitive coevolution should provide for steady progress in the arms race. The genetic algorithm used must not only be strong in searching the particular domain, but also maintain adequate diverse sets of parasites for testing the opposing population. Some of our early experiments in competitive coevolution displayed repeated convergence on specialists that were poor test cases overall, with little arms race progress seen.

The use of diverse, strong opponents has been discussed in the context of game-playing (Epstein, 1995; Sun, Liao, Lu, & Zheng, 1994). Here, we want diverse parasites at every stage, appropriate to the current level of the hosts. The parasites need to be diverse in such a way that they provide appropriate testing for *all* hosts in the current population. The parasites must be neither so difficult that the hosts rarely win, nor so easy that the hosts almost always win.

To these ends, we will describe several methods that encourage progress in the arms race and select strong individuals and diverse test sets. First, we give two revealing viewpoints of competitive coevolution. Then, we will describe the extensions, and analyze them mathematically and experimentally.

2.2 The Infinite Population View

Consider an evolutionary model similar to one used in (Lindgren & Nordahl, 1994). Genotypes occupy a fraction of a population, so that these fractions sum to 1. Since these fractions can be any real number between 0 and 1, the population size is essentially infinite. At any point in time, however, only a finite number of distinct genotypes are present in the population. New genotypes may enter via crossover and mutation of the old genotypes. Reproduction still takes place in discrete generations, but because of the infinite size of the population, no genotype can ever become completely extinct.

This idealized model has several important features that differentiate it from a finite-population GA. First, no genotype will go extinct due to sampling errors. For example, innovations that exist in low numbers initially have a significant probability of being lost in a finite-population GA, but will not be lost in this model.

Second, consider a coevolutionary version of this model in which two distinct populations compete. Each genotype from one population competes with each genotype of the other, and receives fitness based on the fraction of the opposing population that it can defeat. Again, no genotypes ever completely disappear, so one that was not a good test earlier may rise again later if it is successful against a new opponent. A new genotype cannot be very successful (in terms of reproducing to the point of occupying a large fraction of the population) unless it is able to defeat most of the opponents ever seen in the opposing population.

The methods described below can be thought of as attempting to capture these features in a finite-population setting.

2.3 Theoretical Arguments

We have discussed elsewhere a theory of learning in competitive environments (Rosin & Belew, 1996). This theory depends on strong assumptions about the search algorithm used. It would be very difficult to prove that the GA satisfies these assumptions, so the theory cannot be directly applied to competitive coevolution. Nevertheless, it does shed light on desirable properties of competitive procedures.

It is important, for the sake of rapid convergence to perfect solutions, that new individuals (or sets of individuals) be found that are capable of defeating all of the prior individuals. Without such a guarantee, the system can get stuck on weak strategies that defeat each other in a cycle. If we have such a guarantee, we will be sure to progress towards optimality. In fact, theoretical randomization criteria can be described under which such a method will discover perfect solutions in polynomial time (Rosin & Belew, 1996). For our purposes here, we should seek methods that progress by producing new strategies that defeat older ones (an arms race).

Define an "optimal solution" as one that defeats all possible opponents. By definition, only one side may have an optimal solution. The other side, without an optimal strategy, can only hope to defeat non-optimal opponents. A *teaching set* is a set of individuals that work together by having, for each possible non-optimal opponent, at least one individual in the set capable of defeating that opponent (Goldman & Kearns, 1991; Rosin & Belew, 1996). If no reasonably-sized teaching set exists, then optimal individuals cannot be practically distinguished, and coevolution cannot be expected to find optimal individuals. In general, small teaching sets seem to exist for many problems (Goldman & Kearns, 1991; Rosin & Belew, 1996; Anthony, Brightwell, Cohen, & Shawe-Taylor, 1992).

For competitive coevolution, this suggests:

1. We should encourage some sort of niching, so that such *sets* of individuals may arise and be stable in a population.

2. The required size of samples, against which we test each individual, will be related to the size of teaching sets for the problem.

3. The method for choosing such samples should come as close to choosing a teaching set as is possible in the current evolutionary context.

3 Methods

We now describe three methods for competitive coevolution based on the general issues discussed in the introduction, the infinite-population model that we attempt to approximate, and the theoretical arguments above.

3.1 Competitive Fitness Sharing

The usual way to assign fitness based on success in a set of competitions is to sum the score (often a binary value indicating success/failure) across all competitions (Angeline & Pollack, 1993; Hillis, 1991; Axelrod, 1989). This measure will be called "simple fitness" here. In "fitness sharing" a sharing function is defined to take into account similarity (according to some metric) among individuals (Goldberg & Richardson, 1987). An individual's simple fitness is divided by the sum of its similarities with each other individual in the population, rewarding unusual individuals. Related to this is the "emergent fitness sharing" technique found to promote generalization in an immune system model (Smith, Forrest, & Perelson, 1993).

Our method is similar to both of these methods. In the "competitive fitness sharing" (henceforth, "fitness sharing" or "shared fitness") method we propose here, each parasite is treated as an independent resource to be shared by those hosts in the population that can defeat it. This is like having a separate sharing function for each parasite. The fitness assigned to a host defeating parasites with the set of indices X is $\sum_{j \in X} \frac{1}{N_j}$, where N_j is the total number of hosts in the population defeating parasite j. Fitnesses are most comparable when all hosts compete against the same set of parasites, and this is the case for all experiments here.

The effect of this method is to reward hosts that defeat parasites few others can, even though the rewarded host might not defeat *as many* parasites as others can. This is most relevant when there do not yet exist hosts that can defeat all of the parasites. In such a case, any host that defeats parasites that others cannot may well contain important genetic material.

As an example of the advantage that fitness sharing may provide, consider a situation in which a host type G defeats I parasites which no other host type can defeat. Assume that other host types defeat a greater number of parasites, on average. Thus, G is important in that it contains information about defeating parasites that no others can, but it defeats a below-average number of parasites overall. Assume G only had a small number of representatives in the population. This might be because it is a new innovation, or an old niche that recently became important. Or, it may simply be that random fluctuations carried its population to low numbers. Under simple fitness, G would have below-average fitness, and would stand a good chance of becoming extinct this generation (with exact probability depending on the type of selection). Under competitive fitness sharing, though, individuals of type G will receive substantially greater fitness, because they are rare: I fitness will be divided among a small number of individuals. Other, more prevalent types will have lower per-individual fitness, because their fitness is being divided among a larger number of host individuals. Therefore, the small number of individuals of type G stand a good chance of being selected, preventing the immediate extinction of G. A more detailed analysis along these lines is provided in Section 4.2.

Important genotypes such as G are never lost in the infinite population model; competitive fitness sharing helps ensure that they are not lost in the finite-population approximation.

Competitive fitness sharing is more able to allow the survival of important but poorly represented types. So, a greater number of diverse niches should be supportable under competitive fitness sharing. When many such niches exist they are necessarily small and subject to extremely low numbers due to sampling errors. Under simple fitness, there is a greater danger of extinction under these conditions than with competitive fitness sharing.

Recently, several authors have used a tournament format based on (Smith et al., 1993) with competitive coevolution. This format yields fitness with properties similar to those given by competitive fitness sharing. This has been successfully applied to the Prisoners' Dilemma (Darwen & Yao, 1996) and the game of Dots-and-Boxes (Weaver & Bossomaier, 1996).

3.2 Shared Sampling

It is desirable to reduce the computational effort expended in competition by testing each individual in the host population against only a limited sample of parasites from the other population. One way to get a representative sample of parasites is to pick a random subset of them. This is one technique explored below.

However, more information is available about individuals from the previous generation, and if parasites are selected from this set, we can obtain a more challenging sample. For example, parasites with the best fitness in the last generation might be preferentially selected for play in this generation. If there are several niches in the parasite population, however, this would tend to pick similar individuals from the niche that happened to have highest fitness, rather than selecting a diverse set of parasites from all niches.

It would be preferable to choose parasites from the previous generation that challenge all segments of the host population. Given current information, we want to come as close as possible to choosing a teaching set. A natural way to do this is to use competitive fitness sharing for sampling, as we did for selection. New sample members are chosen to have maximal competitive shared fitness within the current sample. In this way, each successive member of the sample is chosen to be one that competed well against individuals that the other members of the current sample did not compete well against. This technique will be called "shared sampling", and is specified in detail by the algorithm in Figure 1. Note that the "opponents" there are the individuals that were sampled in the previous generation to rate the fitness of the parasite population currently being sampled.

```
Initialize current sample to be the empty set
For each opponent i from previous generation
    beat[i]=0 (# in current sample beating i)
While the current sample is not yet full
    For each parasite j not yet in sample
        samp_fit[j]=0 (fitness within sample)
        For each opponent k beat by j last gen.
            samp_fit[j] += 1/(1+beat[k])
    Let j be such that samp_fit[j] is maximal
    Add individual j to current sample
    For each opponent i from previous generation
        If j beat opponent i last gen.
            Increment beat[i]
```

Figure 1: Shared Sampling

This sampling technique helps to address the concern mentioned above (Section 2.3) about providing teaching sets. In order to best distinguish opponents on the basis of competition with the sample, it is desirable to have the sample contain tests that defeat any possible imperfect opponent (this ideal should be better approximated as coevolution proceeds); otherwise, an imperfect opponent could look better than it actually is. Shared sampling selects a set of strong individuals from the previous generation that would have provided a relatively complete set of tests.

It is interesting to note that shared sampling can also provide some of the effects of competitive fitness sharing described above, even when competitive fitness sharing is not being used. In addition to choosing parasites that challenge all hosts, shared sampling will tend to choose more representatives of parasite types that beat a large number of hosts. So, parasite types that are defeated by a large segment of the host population will generally be represented less in the sample than those defeated by few members of the host population. Therefore, even under simple fitness, hosts receive greater fitness for defeating parasites that few other hosts can defeat. This effect - that simple fitness with shared sampling can provide performance comparable to that of competitive fitness sharing - will be seen in some of the results below.

3.3 Hall of Fame

In the infinite population model, new hosts can only be very successful if they are able to defeat most or all of the old parasites, since these old parasites are still present. If the genetic operators can continue to find new genotypes in both populations that defeat all previous opposition, progress towards optimal individuals is guaranteed. This is clearly an important feature.

A finite population, in contrast, has a very short memory. A genotype must be successful almost every generation to stay in the population. Once gone, it can be difficult to rediscover. In using the GA for static optimization problems, this is not very significant because old genotypes were strictly worse according to the fitness measure. In competitive coevolution, however, such old genotypes might have become successful again. And, losing them from the population allows the opposing population to be successful with new types that would lose against old, extinct genotypes. The result is that progress towards the optimum is no longer guaranteed.

So, in competitive coevolution, we have two distinct reasons to save individuals. One reason is to contribute genetic material to future generations; this is important in any evolutionary algorithm. Selection serves this purpose. Elitism serves this purpose directly by making complete copies of top individuals.

The second reason to save individuals is for purposes of testing. To ensure progress, we may want to save individuals for an arbitrarily long time and continue testing against them. To this end, we introduce the *hall of fame*, which extends elitism in time for purposes of testing. The best individual from every generation is retained for future testing. Hosts are tested against both current parasites, and a sample of the hall of fame. Successful new innovations cannot overspecialize; they are required to be robustly successful against old parasites.

We have experimented with performance-based methods of sampling the hall of fame. For example, we tried updating "fitness" of hall of fame members by playing them against the parasites for the current generation, and selecting a sample from the hall of fame based on this fitness. We have found the computational effort required to maintain performance information on members of the hall of fame is much greater than the benefit obtained over random sampling. Therefore, we use random sampling of the hall of fame in all experiments described here.

4 Analysis

4.1 Equivalence of Equilibria under Simple/Shared Fitness

We show here that the equilibria of coevolution are identical under simple and shared fitness. Equilibria are thus one element of long-term coevolutionary dynamics that is preserved when we move from simple fitness to shared fitness (though stability of these equilibria is a more difficult question that we do not consider here).

In the idealized evolutionary model considered here, the only operator is selection: it is assumed that no new genotypes are created. The specific dynamics associated with a specific type of selection are not considered. Instead, only the equilibria are discussed: for any common type of selection, a population will be at equilibrium when all of its members have equal fitness. This is only approximately true in a finite population, when selection has a stochastic component. So, an infinite population will be considered: each genotype occupies a certain real fraction of the entire population. Full round-robin competition is assumed.

4.1.1 Definitions and Equilibrium Conditions

Each genotype in each population occupies a certain fraction of that population. These fractions sum to 1 in each population. The vector of these fractions for the first population will be denoted X, with components x_i, and Y for the second population, with components y_j. A first-population genotype will be denoted i and a second-population genotype j.

The matrix which defines the binary competition outcomes is denoted B, with component b_{ij} equal to 1 if genotype i defeats j, 0 otherwise.

Simple fitness for a host is given by the fraction of the parasites defeated by that host. Since in equilibrium, all fitnesses will be equal, the conditions for simple equilibrium can be expressed as:

$$BY = C_1$$

$$X^T(1 - B) = C_2^T$$

All vectors are column vectors. The T denotes matrix transpose, and 1 will always refer to the all-1 matrix, column vector, or scalar, as appropriate. C_1 and C_2 are constant vectors with all components c_1 and c_2 respectively.

Shared fitness for a host is the sum, over all parasites defeated, of the fraction occupied by that parasite in its population over the the fraction of the host's population that defeats that parasite. Linear algebra notation can be inconvenient for expressing this. To help, adjusted vectors X' and Y' will be defined. Component i of X' is given by $\frac{x_i}{((1-B)Y)_i}$, and component j of Y' is given by $\frac{y_j}{(X^T B)_j}$, where the subscripts indicate a particular component of the vector. These adjusted vectors give the fitness contribution from each parasite genotype. So, the shared equilibrium conditions can be expressed as:

$$BY' = 1$$

$$X'^T(1 - B) = 1^T$$

These conditions express the fact that all fitnesses must be equal to 1 at shared equilibrium. No genotype can be undefeated (since it could not then be in equilibrium with other genotypes), so each parasite population contributes a total of 1 fitness to the opposing host population. The total of the fractions of the population that each host occupies is 1, so the average fitness must be 1. Since all fitnesses are equal in equilibrium, all fitnesses are 1. Note that, to show that the shared equilibrium conditions are satisfied, it is sufficient to show that all fitnesses are equal; equality with 1 follows by the condition on average fitness that is always satisfied (even away from equilibrium).

Note that since the elements of X sum to 1, $(1 - B)X = 1 - BX$, and $X^T(1 - B) = 1^T - X^T B$, and similarly for Y.

4.1.2 Equivalence of Equilibria

Lemma 1 *Any simple equilibrium is also a shared equilibrium.*

Proof: First, assume that a simple equilibrium exists, given by $X = V_1$ and $Y = V_2$ (so that $BV_2 = C_1$ and $V_1^T(1 - B) = C_2^T$). Then $(1 - B)V_2 = 1 - BV_2 = 1 - C_1$, another constant vector with all components equal to $1 - c_1$. And, $V_1^T B = 1^T - 1^T - V_1^T B = 1^T - V_1^T(1 - B) = 1^T - C_2^T$, a constant vector with all components equal to $1 - c_2$. So, taking $X = V_1$ and $Y = V_2$, $X' = \frac{V_1}{1-c_1}$, and $Y' = \frac{V_2}{1-c_2}$. Thus,

$$BY' = (BV_2)(\frac{1}{1 - c_2}) = C_1 \frac{1}{1 - c_2}$$

a constant vector, and

$$X'^T(1 - B) = V_1^T(1 - B)(\frac{1}{1 - c_1}) = C_2^T \frac{1}{1 - c_1}$$

another constant vector. So, the conditions for a shared equilibrium are also satisfied. □

Lemma 2 *A shared equilibrium is a simple equilibrium.*

Proof: Assume not: we have an assignment to X and Y that is a shared equilibrium but not a simple equilibrium. Consider this shared equilibrium. Let w refer to any genotype receiving lowest simple fitness in the first population, with w' defined similarly for the second population. Let b and b' be genotypes in the first and second populations receiving highest simple fitness.

Denote simple fitness of a genotype g by $si(g)$, and shared fitness by $sh(g)$. We have

$$sh(g) = \sum_j \frac{b_{gj} y_j}{\sum_k b_{kj} x_k}$$

for g in the first population, and

$$sh(h) = \sum_i \frac{(1 - b_{ih}) x_i}{\sum_k (1 - b_{ik}) y_k}$$

for h in the second population. These can be expressed as:

$$sh(g) = \sum_j \frac{b_{gj} y_j}{1 - si(j)}$$

$$sh(h) = \sum_i \frac{(1 - b_{ih}) x_i}{1 - si(i)}$$

Now, assume (towards a contradiction) that w defeats only genotypes with simple fitness smaller than $1 - si(w)$. Then

$$sh(w) < \sum_j \frac{b_{wj} y_j}{1 - (1 - si(w))}$$

$$sh(w) < \frac{1}{si(w)} \sum_j b_{wj} y_j$$

$$sh(w) < \frac{1}{si(w)} si(w)$$

$$sh(w) < 1$$

which is a contradiction, since we assumed that we have a shared equilibrium, in which it must be the case that $sh(x) = 1$ for all x. So, at least one genotype defeated by w must have simple fitness of at least $1 - si(w)$. Since at least one genotype in the population opposing w has fitness at least this large, $si(b') \geq 1 - si(w)$. Symmetrically, at least one genotype defeated by w' must have simple fitness at least $1 - si(w')$, so that $si(b) \geq 1 - si(w')$.

Proceeding similarly, assume that b defeats only genotypes with simple fitness larger than $1 - si(b)$. Then

$$sh(b) > \sum_j \frac{b_{bj} y_j}{1 - (1 - si(b))}$$

$$sh(b) > \frac{1}{si(b)} si(b)$$

which is again a contradiction. So, at least one genotype defeated by b must have simple fitness of at most $1 - si(b)$. This implies that $si(w') \leq 1 - si(b)$. Symmetrically, $si(w) \leq 1 - si(b')$.

We have $si(b) \geq 1 - si(w')$, which implies $si(w') \geq 1 - si(b)$. Together with $si(w') \leq 1 - si(b)$, this means that $si(w') = 1 - si(b)$. Symmetrically, $si(w) = 1 - si(b')$.

Now, w may only defeat genotypes with simple fitness equal to or less than $1 - si(w)$. If w defeats only genotypes with simple fitness equal to $1 - si(w)$, then:

$$sh(w) = \sum_j \frac{b_{wj} y_j}{si(w)}$$

$$sh(w) = \frac{1}{si(w)} si(w) = 1$$

If w defeats any genotypes with simple fitness less than $1-si(w)$, then at least one term in the above sum will decrease, and none will increase. This would give $sh(w) < 1$, a contradiction. So, w only defeats genotypes with simple fitness equal to $1 - si(w)$. Similarly, w' defeats only genotypes with simple fitness $1 - si(w')$.

Proceeding similarly, we know that b may only defeat genotypes with simple fitness equal to or greater than $1 - si(b)$. If b defeats only genotypes with simple fitness equal to $1 - si(b)$, then:

$$sh(b) = \sum_j \frac{b_{bj} y_j}{si(b)}$$

$$sh(b) = \frac{1}{si(b)} si(b) = 1$$

If b defeats any genotypes with simple fitness greater than $1 - si(b)$, then at least one term in the above sum will increase, and none will decrease. This would give $sh(b) > 1$, a contradiction. So, b only defeats genotypes with simple fitness equal to $1 - si(b)$. Similarly, b' defeats only genotypes with simple fitness $1 - si(b')$.

Since we are assuming that we do not have a simple equilibrium, assume without loss of generality that the first population does not have all equal simple fitness. Call the portion of the first population with simple fitness less than $si(b)$ the *remainder*. No genotype in the remainder can lose to any genotype with simple fitness $si(w')$ (since genotypes with simple fitness $si(w')$ can only defeat genotypes with fitness $1 - si(w') = si(b)$). So, each genotype in the remainder defeats every opponent with simple fitness $si(w')$. Since genotypes in the first population with simple fitness $si(b)$ can only defeat opponents with simple fitness $si(w')$, each genotype in the remainder defeats all of the opponents that any genotype with fitness $si(b)$ defeats. So, genotypes in the remainder receive simple fitness of at least $si(b)$, which contradicts the definition of the remainder.

Thus, there cannot exist a shared equilibrium that is not a simple equilibrium. □

From these two lemmas, we get:

Theorem 3 *Given particular sets of first and second population genotypes, the sets of simple equilibria and shared equilibria are the same.*

4.2 Extinction Probability

Assume that out of all parasites tested against in a particular generation, there were O parasites, each of which had at least one host capable of defeating it. The goal in this section is to find the probability that, for at least one of these O "defeatable parasites", selection does not actually choose any host capable of defeating it.

Probabilities will be derived here assuming that roulette wheel selection is used. The result should be comparable under tournament selection, but the analysis is more difficult. A similar, far more extensive analysis has been done for fixed fitness functions with conventional fitness sharing (Mahfoud, 1995).

First, assume that competitive fitness sharing is being used. The total fitness received by the host population is O, since each defeatable parasite contributes a total of 1 fitness to the host population, divided evenly among those hosts able to defeat it. So, a host with fitness f has probability $\frac{f}{O}$ of being selected during the roulette wheel selection of a single individual. If some set of hosts has total fitness F, the probability of selecting a member of this set is $\frac{F}{O}$.

Since each defeatable parasite contributes a total of 1 to those hosts able to defeat it, the hosts capable of defeating it must have a total fitness of at least 1. So, for each defeatable parasite, the probability of selection choosing a host capable of defeating that parasite is at least $\frac{1}{O}$. If N hosts are selected for a new population, the probability of not selecting any opposition for a particular parasite is at most $(1 - \frac{1}{O})^N$. So, the probability of not selecting any opposition for at least one parasite is at most $O(1 - \frac{1}{O})^N$. As an example, for $N = 1000$ and $O = 100$, the probability of such an extinction is less than 0.005.

Under simple fitness, the probability of not selecting any opposition for at least one defeatable parasite can be much higher. To take a worst case, consider a situation in which every defeatable parasite except for one, o_0, is defeated by every member of the host population except for one, p_0. p_0 defeats o_0; this is what makes o_0 defeatable. Now, if the host population has N members, $N-1$ of them receive fitness $O-1$, and 1 of them receives fitness 1. Each selection, host p_0 has a probability of $\frac{1}{(N-1)(O-1)+1}$ of being chosen. In selecting N hosts, p_0 has a probability of $(1-\frac{1}{(N-1)(O-1)+1})^N$ of not being chosen at all. Since N and O will generally be large, this probability can be very high. In our example with $N=1000$ and $O=100$, this probability is about 0.99.

This shows that loss of all opposition for a defeatable parasite is unlikely under competitive fitness sharing, but that no such guarantee is possible with simple fitness. Extinctions are examined empirically in Section 7.3.

5 Experimental Setup

Experiments were done to explore the effectiveness of the three extensions to competitive coevolution described above. These experiments were done on two games. The first of these was a small version of Nim, with about 600 nodes in its game graph. The small size of this game allows for a simple representation and understandable results. Additional experiments used a much larger game, 3-dimensional Tic-Tac-Toe (3DTTT) on a 3x3x3 board.

5.1 Representation of Game Strategies and Genetic Operators

5.1.1 Nim

In Nim, a position in the game consists of several piles of stones. Players alternate removing an arbitrary number of stones from a single pile. The player to take the last stone wins. The configuration used here starts with 4 piles, containing 3, 4, 5, and 4 stones. This configuration allows the first player to force a win with optimal play (Berlekamp, Conway, & Guy, 1982).

A simple one-gene-per-position representation is used for Nim. The positions are placed on the genome in lexicographic order: (0,0,0,0), (0,0,0,1), ..., (3,4,5,4), where each quadruple gives the number of stones in each of the four piles. Actually, (3,4,5,4) is not needed in this representation; since this is the initial position, it never needs to be evaluated (see below). This leaves a total of 599 positions on the genome.

A binary *evaluation function* is used: each gene can take allele values 0 and 1. To make a move, legal moves are tested using the evaluation function, and the first position evaluated as 1 is selected as the move. Positions are tested in the following order. Piles are selected in order from first through fourth. For each pile containing $k>0$ stones, moves leaving the pile with 0 up to $k-1$ stones are tested in order. If none of these positions is evaluated as 1, the first move in this ordering (not generally a good move) is selected by default.

Traditional crossover and mutation are used, with a 0.004 probability of crossover at each gene, and 0.0022 probability of mutation at each gene. These were chosen to give averages a bit over 2 crossovers and 1 mutation per generated individual.

Though this problem is easily solved by game-tree search, this particular form of it is somewhat more difficult because the GA has no knowledge about the game tree and cannot directly search it. Also, the GA needs to find a strong set of opposing strategies to use as test cases (a teaching set) in order to solve the game. Random sampling in the space of test cases is inadequate for this game (see Section 7.2).

5.1.2 3DTTT

In this game, players alternate placing markers ("X" and "O") on a 3x3x3 grid. The first player to obtain 3-in-a-row on any row, column, or diagonal (face diagonal or cube diagonal) wins. The game is a very easy first-player win if it is not restricted, so we disallow the first player from making the initial move at the center location. The game is then more interesting, and becomes a second-player win.

There are far too many distinct positions in this game, even with symmetries accounted for, to use as straightforward a representation as was used for Nim. Instead, strategies are represented as an ordered list of condition/action pairs. A condition is a set of simple constraints on the current position. Each constraint specifies one of the 27 locations in the cube, and what must be at that position (friendly marker, enemy marker, or empty). The action then specifies a move.

To make a move with such a strategy during play, the rules in the list are checked in order starting from the first rule in the list. The first matching rule with a currently legal action (unoccupied point in the cube) is allowed to move. Conditions are checked in all possible symmetric configurations (rotations and reflections of the cube; 48 total). Only one symmetry needs to match; if it does, the same symmetry is applied to the action.

The genetic operators are applied directly to this representation. Crossover takes place between rules only. Since rule lists may be variable length, the length of the child is chosen randomly from the interval between the parent list lengths (inclusive). Then, uniform crossover is done: child rule i is set to the first parent's rule i with probability $\frac{1}{2}$, and to the second parent's rule i with probability $\frac{1}{2}$. If only one parent has a rule at position i (because the other parent has length less than i), the child's rule is taken from that parent.

Since crossover does not change rules, a variety of mutation types are used to get a reasonable range of rule variation. A single rule's condition can have a single random constraint added or deleted. Similarly, a single entire rule can be deleted or inserted at a randomly chosen place in the list. A single action can be mutated, and a single pair of rules can be swapped.

Operators were applied with uniform probability. In particular, a mutation operator was performed on each child with 90% probability. When a mutation operator was to be applied, one of the six operators described above was chosen randomly. Offspring were created via crossover with 50% probability, and via copying with 50% probability, before applying any mutation.

Rules were initialized with the number of constraints chosen randomly from the range $[0-3]$. The number of rules in each individual in the initial population was chosen randomly from the range $[1-8]$. The maximum number of rules was set at 30, and the maximum number of constraints in a rule was set at 8.

5.2 Genetic Algorithm Details

For both Nim and 3DTTT, the first population contains first-player strategies, and the second contains second-player strategies. Though the two populations use the same representation, the requirements of the populations are different since the strategies play different roles in the game.

We have found tournament selection to work well in competitive coevolution; parents are selected in independent 2-tournaments. Elitism is important in these problems. If it is not used, an optimal individual may be found and then lost. Additionally, elitism may work well in conjunction with fitness sharing to protect important innovations. An elite size of 1 was used in Nim, and 20 in 3DTTT. Each population had 500 members in Nim, and each had 1000 members in 3DTTT.

Sampling was done without replacement (though multiple copies of an individual may appear in a sample if multiple copies appear in the population or hall of fame being sampled). In initial generations, the hall of fame does not contain enough distinct individuals to obtain a large enough sample, so the hall of fame sample is filled out with random individuals as necessary for these first few generations.

The use of standard fitness sharing with tournament selection can lead to fluctuations in niche sizes. The method of *continuously updated sharing* has been developed to overcome this problem (Oei, Goldberg, & Chang, 1991), This method involves calculating shared fitness within the next generation's population, currently being constructed, rather than in the original population. This resembles the method of shared sampling used here. Continuously updated sharing was tried here with competitive fitness sharing, on Nim. It was not found to be helpful. The fluctuations discussed in (Oei et al., 1991) may be less of a problem due to the presence of a large number of niches, and the fact that competitive fitness sharing sums fitness over many components. These factors would tend to reduce the problem that leads to fluctuations: one large niche winning all selection tournaments against another large niche.

The combination of extensions used, and the sample sizes used for testing, will be given for each of the particular experiments described below.

6 Comparative Results

6.1 Nim

For Nim, simple fitness and shared fitness were tested:

1. without sampling (using entire parasite population);
2. with random sampling of 100 parasites; and
3. with shared sampling of 100 parasites.

The hall of fame was tried without the other extensions, and with both of the other extensions. Details are given in Table 1. Performance is measured as the number of games required to obtain a perfect first player strategy. That is, the genetic algorithm is run until a perfect strategy is found, and a count is made of the total number of games played during fitness evaluations.

Most of the runs that used sampling used the same total number of parasites (100). Smaller numbers of parasites sometimes worked, but performance was not as consistent when this was done. The exception to this was the set of runs that used all three extensions. In this case, a solution may be consistently found with a smaller number of parasites, so a total of only 50 were used.

Table 1 shows results for the variants studied. The numbers in parentheses under **Sampling** and **Hall of Fame** show how many current and hall of fame parasites were used, respectively. **Reps.** gives the number of runs of that type, and **% Complete** gives the percentage of these runs that found optimal solutions. When sampling is not used, each game is used twice: once with the population 1 individual playing host and once with the population 2 individual playing host; the results given in this case reflect the fact that each such game is only played and counted once, with the outcome used for both purposes.

No.	Fitness	Sampling	Hall of Fame	Reps.	% Complete	Avg	Std Dev
(i)	Simple	None (500)	no	5	0%	($\gg$ 3000)	-
(ii)	Simple	Random (100)	no	5	0%	($\gg$ 3000)	-
(iii)	Simple	Shared (100)	no	10	100%	201	60.6
(iv)	Shared	None (500)	no	10	100%	806	330
(v)	Shared	Random (100)	no	10	100%	353	116
(vi)	Shared	Shared (100)	no	10	100%	145	27.5
(vii)	Simple	Random (50)	yes (50)	10	100%	1040	714
(viii)	Shared	Shared (25)	yes (25)	10	100%	73.0	15.6

Table 1: Number of Games to Solve Nim (in Millions)

Runs using simple fitness, without sampling and with random sampling, never completed. These were run for 3 billion games (12000 generations and 30000 generations, respectively), and though there was initial progress, these runs showed no hope of eventual success.

Significance tests are done at the 5% level with a nondirectional Mann-Whitney rank test. Since this test works with ranks, we can sensibly include runs that did not complete; these are all ranked identically as the worst (since no run that completed took longer than these were run). Significant differences are shown in Table 2. A "*" indicates a significant difference. The Mann-Whitney test is used for all significance tests in this paper.

The game cannot be solved without the use of at least one of the three new methods. Using any one of the three methods is adequate to eventually solve the game. Using methods in combination allows significantly faster solutions, with the use of all three methods being fastest overall.

	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
(i)	X	*	*	*	*	*	*	*
(ii)		X	*	*	*	*	*	*
(iii)			X	*	*	*	*	*
(iv)				X	*	*	-	*
(v)					X	*	*	*
(vi)						X	*	*
(vii)							X	*
(viii)								X

Table 2: Significant Differences ($p < 0.05$) in Number of Games to Solve Nim

6.2 3DTTT

For 3DTTT, each of the two populations contained 1000 members[1]. Runs were stopped when a perfect second-player strategy was found. Since these runs were computationally expensive, we only compared coevolution with all three extensions to coevolution without any of the extensions.

No.	Fitness	Sampling	Hall of Fame	Reps.	% Complete	Avg	Std Dev
(i)	Simple	Random (100)	no	5	60%	(> 1210)	(473)
(ii)	Shared	Shared (50)	yes (50)	5	100%	285	71.2

Table 3: Number of Games to Solve 3DTTT (in Millions)

Table 3 gives results. Some of the runs in (i) did not complete, though they were run for 2 billion games (10000 generations). The runs that did not complete were not included in the average and standard deviation; this is indicated in the table with parentheses.

Though coevolution without the extensions was sometimes able to solve the game, the fastest run without extensions was slower than the slowest run with the extensions. The difference is significant ($p < 0.01$). Coevolution with the three extensions was able to solve this large game in a number of games that was on the order of the number of games required to solve Nim.

Since coevolution without extensions was unable to solve Nim, it is somewhat surprising that it was sometimes able to solve the apparently more complex 3DTTT problem. This is discussed further in Section 7.7.

As a speed comparison, we attempted to solve 3DTTT with a naive conventional depth-first minimax search. Over 7 billion nodes were searched in 1 day of CPU time on a 132 MHz PPC 604. At this point, the search was still exploring variations that begin with the first possible first-player move and the first possible second-player response to this first-player move. A much larger amount of time would have been required to complete the search.

There are many early wins in 3DTTT. Search can be greatly improved by utilizing this knowledge: we modified search to check for two-in-a-row configurations that result in immediate wins for the moving player. Doing this allowed a rapid solution to the game (under a minute of CPU time). The final coevolved solution still has the advantage of being very compact, fast, and comprehensible.

7 Interpretation

7.1 Typical Runs

[1] Substantially smaller populations did not work well, probably due to the large fraction of offspring ruined by the complex genetic operators.

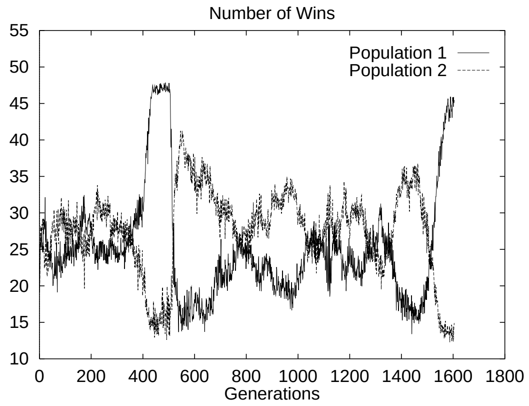


Figure 2: Number of Wins for Both Populations in Nim

In typical runs on Nim, populations take turns being ahead in the arms race, with fitness measured against current opponents fluctuating for both populations. Figure 2 shows average number of wins during a typical successful run, using fitness sharing, a shared sample of 25, and a hall of fame sample of 25. The competition seems balanced, with neither population dominating for too long.

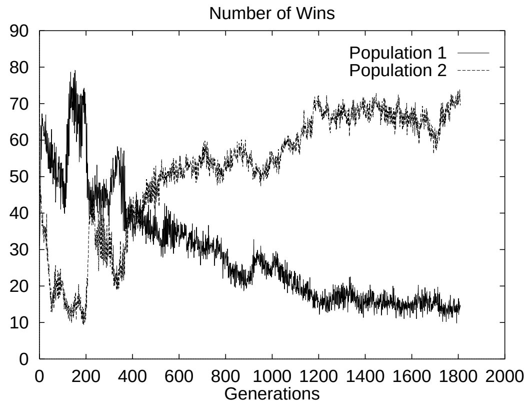


Figure 3: Number of Wins for Both Populations in 3DTTT

For 3DTTT, there is less fluctuation. Figure 3 shows a typical plot of average fitness for a successful run using fitness sharing, a shared sample of 50, and a hall of fame sample of 50. The first population typically dominates early in the run (there is an advantage in having an extra piece on the board when playing against weak opponents). Later in the run, the second population begins to dominate. At this point, the first population is still managing to cover the second population; very rarely is a particular second-population genotype able to defeat *all* first-population parasites for more than one generation. But, the first population falls behind in ability to defeat a large number of parasites with a single strategy. Larger sets of first-population hosts are required to cover all second-population parasites, and first-population fitness falls.

7.2 Testing

An important basic question is whether coevolution was necessary at all. If a large, randomly chosen set of parasites was also sufficient to produce a perfect strategy, then search over examples was unnecessary. Several experiments were done in which a fixed set of test cases was chosen randomly (corresponding to a fixed second population in the experiments above), and strategies were tested against these. No sampling was used; all parasites were used with every individual tested.

For Nim, both with and without fitness sharing, no strategy could be found that defeated all test cases (with a variety of test case set sizes used, from 2000 up to 100000), though runs with fitness sharing came quite close (for example, failing to defeat only 12 cases out of 100000). Population sizes of up to 10000 were used, so it appears that coevolution is really necessary to find perfect strategies for Nim.

Random tests were also used on a smaller version of Nim (3,4,5). Here, strategies were found that could beat all tests, but even when 200000 test cases were used, these strategies were not perfect. This suggests that even if the GA had succeeded in finding strategies that could beat all test cases for large Nim, these strategies would likely be imperfect.

Comparing these results using random testing with those of competitive coevolution brings to mind issues discussed in the introduction. The first is that coevolution is indeed able to find a set of test cases adequate for the evolution of optimal individuals. The results on the small version of Nim indicate that finding such adequate sets can be a difficult problem, requiring much more search than random sampling can be expected to accomplish.

The second point concerns finding *pedagogical* sequences of test cases. For the large version of Nim, random tests apparently provided a rough fitness landscape that was too difficult for the GA to successfully search. There existed some test cases which were a sort of "needle in the haystack"; defeating other test cases did not provide enough information to get close to defeating these difficult cases. This is not as much of a problem with competitive coevolution: it seems that one population never gets so far ahead that the other population cannot catch up (until a perfect individual is found). The difficulty of test cases stays relatively appropriate at each step of evolution.

7.3 Extinction

As described in Section 4.2, fitness sharing and shared sampling should theoretically reduce the extinction of host types that defeat parasites others cannot. Here, we provide empirical support. Each generation, for each population, a count is made of the number of sampled parasites P meeting two conditions:

1. There existed hosts H this generation which defeated P.

2. No such host H was selected to contribute genetic material to the next generation.

This describes the sort of extinction events considered in Section 4.2. Note that the loss of a single host may be counted as several extinction events (one for each parasite which it beat, but which no other host could beat). For purposes of comparison, the average number of extinction events per generation is calculated, and the average of this quantity across multiple runs is presented. Extinction events are typically fairly rare during a successful run.

Table 4 shows average extinction events per generation for the eight variants, on Nim. For variants which could not find perfect strategies, the first 2500 generations were used. P1 refers to extinctions among second-population hosts tested against first-population parasites, and P2 refers to extinctions among first-population hosts tested against second-population parasites.

Note that the total number of extinction events depends on the total number of parasites used; for example, there are potentially far more extinction events when the entire parasite population is used for testing. To normalize for this effect and make averages comparable, the number of extinctions is presented as a percentage of the total number of parasites used. This figure can range from 0% (no extinction events) to 100% (every parasite had some host which defeated it but was not selected).

Table 5 shows significant differences in extinction events. Fitness sharing (and shared sampling, by its similar effects) greatly reduce the number of extinction events. Important genotypes are protected to a greater degree. It is interesting to see that the hall of fame does not by itself substantially reduce extinction events. Although the use of the hall of fame allows runs to succeed, it is not by itself sufficient to protect important genotypes. It must provide some other advantage; we return to this question below.

Table 6 shows extinction events for 3DTTT. Extinction events were present in all runs that did not use the extensions, and in none of the runs that did use the extensions. The large population size probably helped to reduce extinction events. The difference between the two sets of runs is significant ($p < 0.01$).

No.	Fitness	Sampling	Hall of Fame	P1	P2
(i)	Simple	None (500)	no	0.559%	0.384%
(ii)	Simple	Random (100)	no	0.148%	0.0334%
(iii)	Simple	Shared (100)	no	0.0476%	0.00781%
(iv)	Shared	None (500)	no	0.000578%	0.000028%
(v)	Shared	Random (100)	no	0.00025%	0.00015%
(vi)	Shared	Shared (100)	no	0.00796%	0.0%
(vii)	Simple	Random (50)	yes (50)	0.431%	4.51%
(viii)	Shared	Shared (25)	yes (25)	0.00118%	0.0%

Table 4: Average Extinctions Per Generation for Nim (expressed as a percentage of the total number of parasites)

	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
(i)	X	*,-	*,*	*,*	*,*	*,*	-,*	*,*
(ii)		X	-,*	*,*	*,*	*,*	*,*	*,*
(iii)			X	*,-	*,-	*,*	*,*	*,*
(iv)				X	-,-	-,-	*,*	-,-
(v)					X	-,-	*,*	-,-
(vi)						X	*,*	-,-
(vii)							X	*,*
(viii)								X

Table 5: Significant Differences ($p < 0.05$) in Extinctions for Nim (P1,P2)

7.4 Useful Diversity

The simple nature of Nim and our representation of strategies for it allows a close look at the usage of positions in the game. Each time a move is made from a position, during any game in a single generation, a counter for that position is incremented. These counts reveal important positions that often occur as a part of real games. The number of positions with usage above a "significant usage" threshold is measured each generation, for both populations. This threshold is set fairly high (see below), with the goal being to identify positions whose usage has been strongly selected for by the GA, rather than those that happen to be used a few times due simply to a chance mutation. A larger number of significantly used positions should therefore indicate a greater diversity of games played by selected individuals. Such diversity is important to test hosts more fully each generation and provide them with richer fitness information. For purposes of comparison, significant usage is averaged over all generations in a run, and the average of this quantity across multiple runs is presented. Because of the close correspondence between the set of positions and the genes in our representation, this measure is similar to measures of gene usage (Bedau & Packard, 1991).

Table 7 shows significant usage for the eight variants, on Nim. The significant usage threshold was set at

No.	Fitness	Sampling	Hall of Fame	P1	P2
(i)	Simple	Random (100)	no	0.0149%	0.375%
(ii)	Shared	Shared (50)	yes (50)	0.0%	0.0%

Table 6: Average Extinctions Per Generation for 3DTTT (expressed as a percentage of the total number of parasites)

the number of games that 4 hosts play against all their parasites. This is 2000 for the runs without sampling, 200 for the runs using all three extensions (since these used a total of only 50 parasites), and 400 for the remaining runs. For variants which could not find perfect strategies, the first 2500 generations were used.

No.	**Fitness**	**Sampling**	**Hall of Fame**	**P1**	**P2**
(i)	Simple	None (500)	no	44.29	46.92
(ii)	Simple	Random (100)	no	58.33	64.43
(iii)	Simple	Shared (100)	no	91.75	85.86
(iv)	Shared	None (500)	no	90.42	94.45
(v)	Shared	Random (100)	no	91.86	97.18
(vi)	Shared	Shared (100)	no	97.70	97.15
(vii)	Simple	Random (50)	yes (50)	48.22	70.75
(viii)	Shared	Shared (25)	yes (25)	102.5	104.6

Table 7: Significant Usage for Nim

	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
(i)	X	*,*	*,*	*,*	*,*	*,*	-,*	*,*
(ii)		X	*,*	*,*	*,*	*,*	*,-	*,*
(iii)			X	-,*	-,*	*,*	*,*	*,*
(iv)				X	-,-	*,-	*,*	*,*
(v)					X	*,-	*,*	*,*
(vi)						X	*,*	-,-
(vii)							X	*,*
(viii)								X

Table 8: Significant Differences ($p < 0.05$) in Significant Usage for Nim (P1,P2)

Table 8 shows significant differences in usage. The use of fitness sharing (and shared sampling, by its similar effects) yield a significant increase in usage. These methods allow survival of a more diverse set of genotypes, resulting in greater diversity in usage during games played. As with extinction events, the hall of fame by itself does not seem to substantially increase significant usage. Such diversity does not explain the advantage provided by the hall of fame.

During a run, significant usage displays revealing trends. Figure 4 shows average wins and significant usage for the first population, during a run in which fitness sharing and shared sampling of 100 parasites were used. The second population's average wins is a near-mirror image (though not quite, due to sampling), so it is not shown.

Significant usage displays an initial rise, followed by fluctuations which are due in part to temporary convergence and failure in the population. When the population begins to converge to a temporarily successful type (fitness spikes at 670 and 1260 generations), diversity loss can be seen in the drop in significant usage. During periods of relative failure (for example at 450 generations), few types are strongly selected for, and significant usage also drops.

7.5 Arms Race Progress

7.5.1 Domain Dependent Measure

In an ideal arms race, individuals in both populations become strictly stronger, continuously. In general, we will not have an efficiently computable, absolute measure of strength to use in checking this condition. The simplicity of Nim, however, allows an attempt to find such a measure.

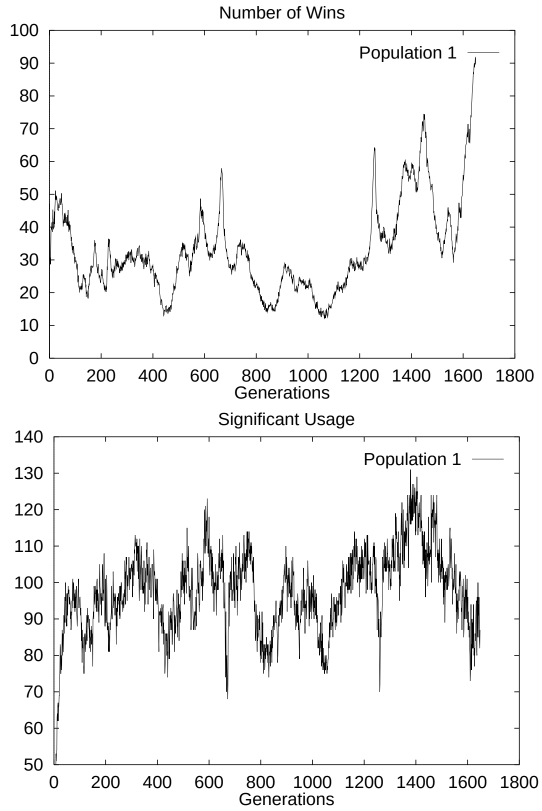


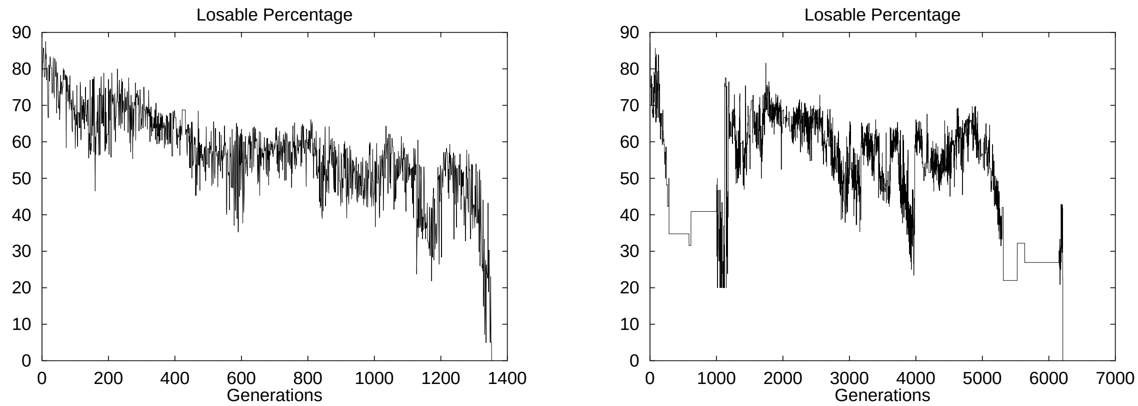
Figure 4: Number of Wins and Significant Usage for Nim

One way to measure progress is what we will call the "losable percentage": the percentage of reachable positions from which a loss may be forced through some line of play, for the best member of the first population in each generation. Roughly speaking, each such losable node is an opportunity that the second population may exploit by leading the first-player strategy to that node, then forcing a win. As these opportunities are exploited and the first population corrects these defects, the losable percentage should decrease, reflecting the arrival of strategies with fewer flaws. A perfect strategy would have a losable percentage of zero.

Figure 5a shows the losable percentage during a successful Nim run with fitness sharing, shared sampling of 25 parasites, and a hall of fame sample of 25 parasites. As in most runs, there is an initial decrease as gross flaws are weeded out. In this run, there is a noisy decrease afterwards to perfection.

Progress is not always clear from these graphs. Figures 5b shows the losable percentage during a successful Nim run with simple fitness, random sampling of 50 parasites, and a hall of fame sample of 50 parasites. Here, the losable percentage fluctuates with little trend. There are periods where the first population is dominating and a single top strategy remains the best for several generations. Eventually, a perfect strategy is found.

Either coevolutionary progress is not present for large portions of the run shown in Figure 5b, or losable percentage is an incomplete measure of progress in the arms race. In either case, it would be preferable to have a domain-independent view of the arms race. This will be discussed in the following section, and this particular run will be reconsidered.



(a) Shared Fit./Shared Samp./Hall of Fame **(b)** Simple Fit./Random Samp./No Hall

Figure 5: Losable Percentage for Nim

7.5.2 Domain Independent View

A different notion of progress is that newly generated hosts should defeat parasites that prior hosts were able to defeat, as well as new parasites. Coevolutionary cycles arise when newly generated hosts are specialized to defeat new parasites, but cannot defeat old parasites that prior hosts were able to defeat.

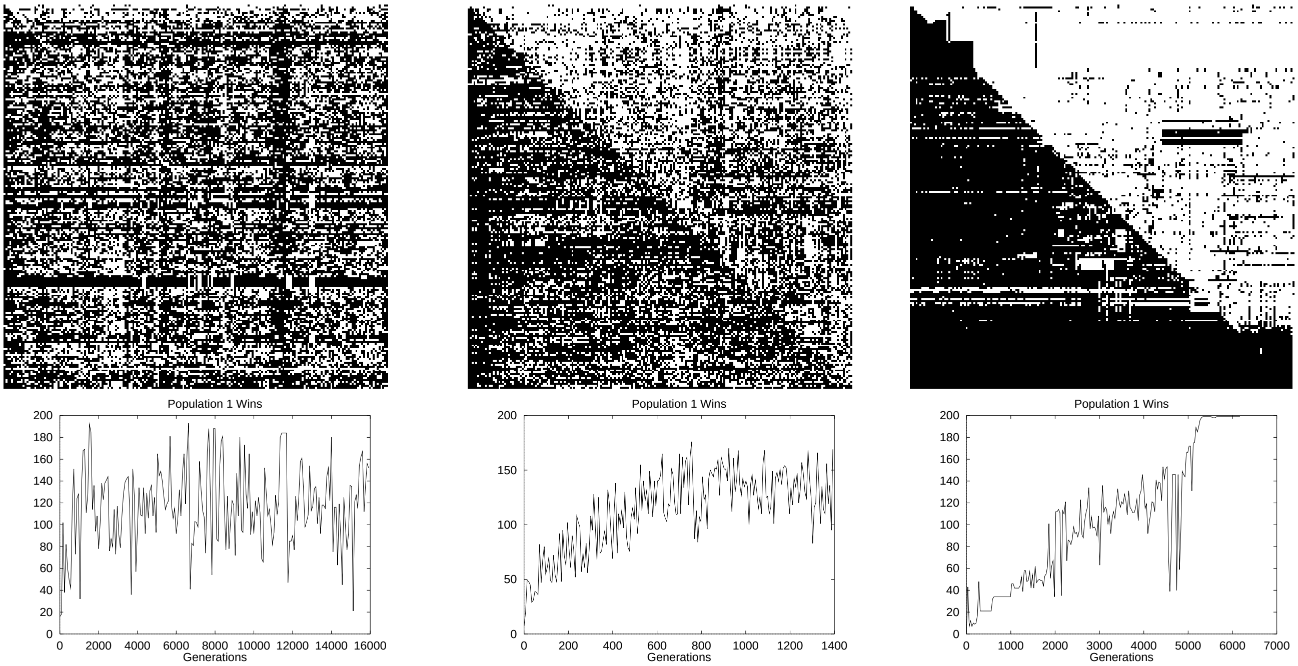
We may attempt to measure progress according to this idea by examining new individuals and testing them against new and old opponents. The best individual from each generation, according to number of wins, is saved (as in the hall of fame). At the end of the run, a full round-robin tournament is played among these individuals. The outcome of this tournament is displayed on a grid, with each row corresponding to a first-population individual and each column corresponding to a second-population individual. Generations go down and left to right, starting from the first generation in the upper left-hand corner. A black dot indicates that the particular first-population strategy won the game between the row and column individuals. For display purposes, we sample generations at a fixed interval to obtain a 200 by 200 bitmap. Similar diagrams have been used to show the outcome of pursuer-evader coevolution (Cliff & Miller, 1995).

In the ideal arms race, each individual would defeat the opponents from all prior generations. This would correspond to a black triangle in the lower left, and a white triangle in the upper right. Short of perfection, we would like to see long-term progress in the form of a dark left edge and a light upper edge. Since we are only taking a single individual from each generation, the diagram may appear noisy if a larger *set* of individuals was required to cover all opposition in most generations. In this case, the particular best individual chosen will be inadequate to cover all previous generations by itself. This effect sometimes shows up as a solid block of black in the white triangle (or vice versa) when it extends over several generations.

To help visualize progress in the arms race, we add up the number of first-population victories (black squares) in each generation (row). The graph of this quantity should display a linear rise to show coevolutionary progress, as first-population hosts are able to defeat second-population parasites from a larger number of generations. For the 3DTTT runs, we add up second-population victories instead, since the game is a second player win.

Figure 6 shows the arms race diagrams for several combinations of extensions, on Nim. Figure 6a shows the breakdown of the arms race for a run in which none of the extensions were used. This plot shows 16000 generations; the game was never solved during this run. Initial generations looked promising (though this is somewhat difficult to see in this figure), but by 16000 generations there is no long-term progress visible. The prominent relevant feature is the diagonal: newly generated types are specialized to defeat current opposition. The graph of first-population victories shows no long-term trend. Other runs that fail to solve the game are qualitatively similar.

Figure 6b shows the outcome of a run with fitness sharing and shared sampling, in which the game was solved after 1514 generations. Here, the two triangles may be clearly seen, showing strong progress. The



(a) Simp. Fit./Rand. Samp./No Hall
16000 generations

(b) Shar. Fit./Shar. Samp./No Hall
1514 generations

(c) Simp. Fit./Rand. Samp./Hall
6212 generations

Figure 6: Arms Race for Nim

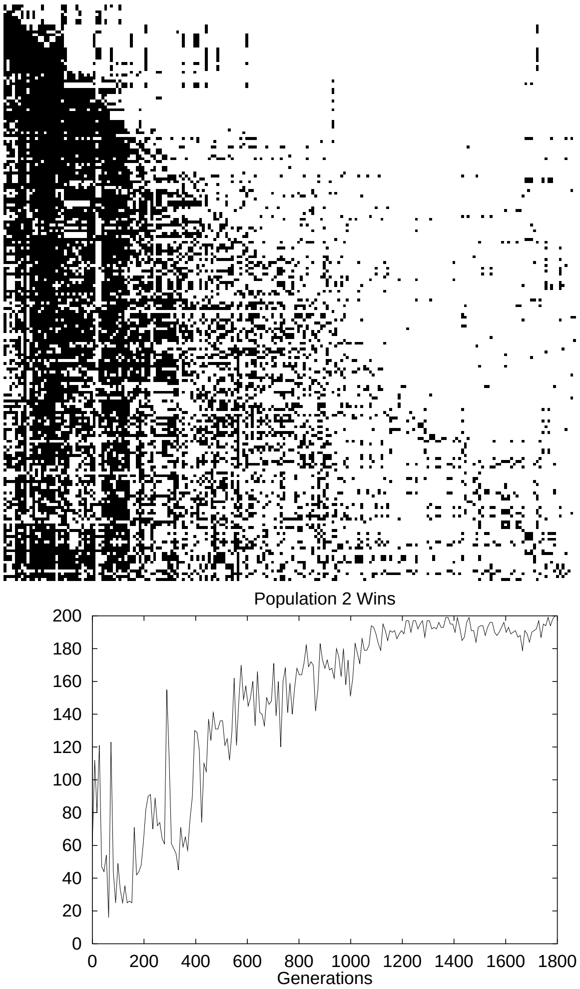
graph of first-population victories shows a noisy rise. This is typical in successful runs that don't use the hall of fame.

Figure 6c shows a run with the hall of fame, but with simple fitness and random sampling. This run completed after 6212 generations. The two triangles are sharply defined. The hall of fame makes a big difference in the arms race, since it causes direct selection for hosts that defeat the best parasites from prior generations. This appears to be the source of the performance advantage provided by the hall of fame.

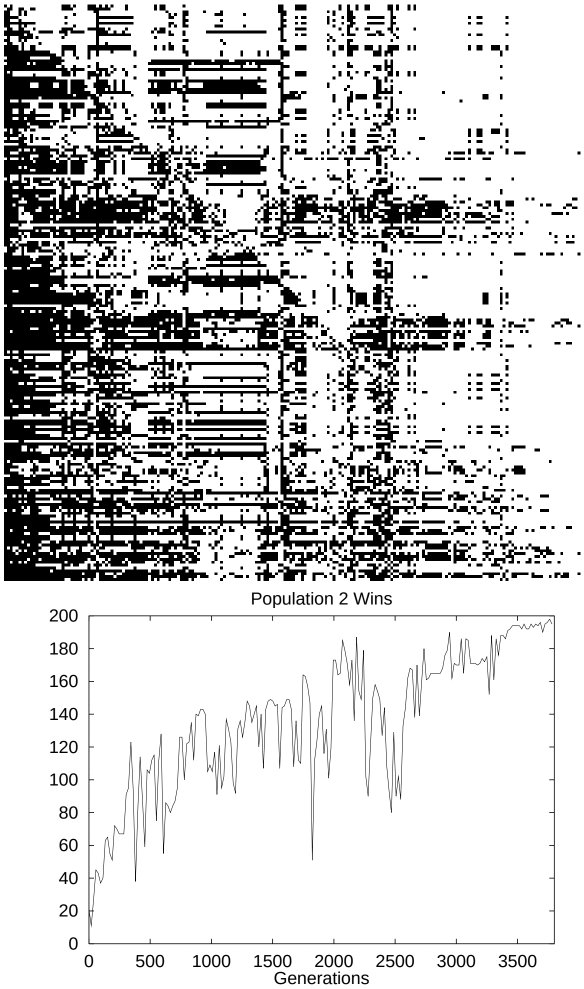
The run in Figure 6c is the same as the run for which losable percentage is shown in Figure 5b. For long periods during this run, the arms race diagram shows clear progress that cannot be seen in losable percentage. As each population exposes flaws in opposing strategies, these flaws are weeded out. Due to the hall of fame, populations remember how to exploit these flaws so that they have a difficult time reappearing. This progress is difficult to see with losable percentage. Losable percentage is a particular domain-dependent global measure of solution quality, but it is inadequate for revealing all coevolutionary progress.

Figure 7a shows the arms race in a successful run on 3DTTT using fitness sharing, a shared sample of 50 parasites, and a sample of 50 parasites from the hall of fame. This run solved the game after 1800 generations. During the early part of the run, the arms race is fairly balanced. The first population is eventually able to defeat most early second population opponents, and largely retains this ability. Later in the run, single first-population hosts are not adequate to defeat all prior parasites. First-population strategies do exist that defeat the second-player strategies from later generations, but larger sets of first-population strategies are required to cover all of these opponents. As a result, the representative first-player strategy chosen for the arms race diagram defeats sparser subsets of these opponents. This feature contributes to the clear rise in second-player victories during the run.

Figure 7b shows the arms race in a successful run on 3DTTT using simple fitness, no hall of fame, and a random sample of 100 parasites. This run solved the game in 3875 generations. There appear to be longer stretches where later strategies are failing to beat earlier opponents; this is to be expected since the hall of fame is not used. There is still an increasing trend in second-player victories during the run. This feature



(a) Shared Fit./Shared Samp./Hall
1800 generations



(b) Simple Fit./Random Samp./No Hall
3875 generations

Figure 7: Arms Race for 3DTTT

stands in contrast to Nim, where no long-term progress in the arms race was visible when the extensions to coevolution were not used. This difference is discussed in Section 7.7.

7.6 Timing and Drift

For pedagogical purposes, the competition between populations should be fairly balanced at all points during coevolution. This way, fitness is most informative and neither side is in danger of losing track of the opposition. Competitive fitness sharing and shared sampling prevent the loss of important genotypes and increase useful diversity. This may help prevent timing problems, since it encourages selection of opposition for all parasites. This way, it is difficult for a parasite to become dominant, with no hosts able to defeat it.

When one side in competitive coevolution gets very far ahead, both sides lose fitness information because the outcome of most competitions is the same. For the losing side, this implies that there will be little selection in the direction it needs to go. For the winning side, this may allow genetic drift[2] if mutation can overcome the now-minimal selection pressure.

The effect of this in competitive coevolution is a potentially beneficial one. Drift will provide degraded versions of the particular parasites that the losing hosts are finding so difficult to defeat. Drift will continue to the point where these degraded parasites may be defeated; this begins to give the losing population some direction. Ideally, the drifting parasites could provide a smooth hill in difficulty, up to the current undefeatable parasite type.

It is difficult to view this effect during a run, because it cannot be easily separated from everything else that is happening. In an attempt to see whether this effect could be significant, a separate experiment was done. The first population was seeded entirely with copies of a first player strategy (for the game of Nim) that was very good but not quite perfect. This difficult first player was chosen from a point near the end of a successful competitive coevolution run. The second population was seeded randomly, and competed against the first (with full crossbar testing, 500 individuals per population, competitive fitness sharing).

As a first test, if the first population is not allowed to drift, there is no fitness variation to guide the second population (they all lose every game), and the search fails (40 million randomly generated 2nd player strategies were all unable to defeat the seed player). So, next, drift is allowed, by using the usual Nim GA parameters. Elitism is such that the seed individual is not displaced until it is defeated.

To track the second population, a comparison was made between the moves that must be made to defeat the seed individual, and the moves that were actually made in games won by the second population against drifting opponents. Figure 8 was obtained as follows. Every game won by a second player was checked to see how many of the moves made (1st and 2nd player) during that game also arise in one of the games by which the seed player can be defeated. Such moves are rare, because there are only a handful of games by which the seed player can be defeated; these games will be called "winning paths". The graph shows this quantity as a fraction of total moves played; if this fraction becomes 1, the participating 2nd player is able to defeat the seed player (this happens at generation 49 in this run).

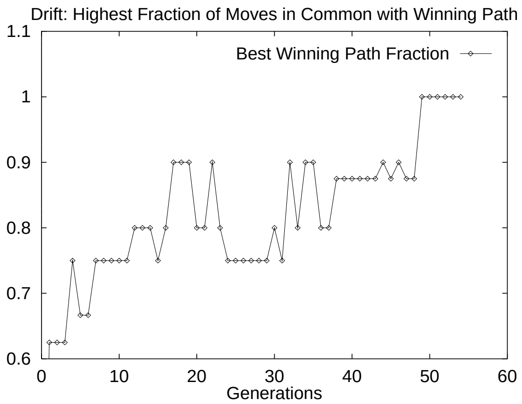


Figure 8: Drift provides a hill to climb

A closer look at the games played by the second-population individuals shown in the graph shows how this is working. Since the seed player is not perfect, there is a particular position from which the seed player makes a mistake. Very few games' paths lead it to this mistake, however, and perfect play is required to capitalize on it after it is made. Drifting versions of the seed player relax both of these conditions. Over the drifting population, there exist more paths to reach the mistake position. Also, exactly perfect play is not

[2] We use the term "drift" to refer to fluctuations due to random forces (as opposed to selection). The term is often used when discussing random convergence of a gene to a particular allele. With our fairly high mutation rates, such convergence does not seem to be an important issue.

necessarily required to capitalize on this mistake against degraded first-population individuals. This allows the second population to find this mistake and make use of it, then improve the play both before and after reaching the mistake until the seed player can be defeated.

Drift in the first population creates degraded individuals which provide sufficient information to guide second population towards defeating difficult first-population opponents. This is a form of self-correction to timing that may occur during coevolution. It resembles the way in which the Baldwin effect can exploit individuals' plasticity (eg. to learn during their lifetimes) to convert difficult evolutionary problems into tractable ones (Belew & Mitchell, 1996).

Drift is likely to be an important factor in tuning parameters. Mutation rates must be high enough to permit significant random variation in a successful population, and sample size must be large enough to include degraded individuals. Other factors may enter into the control of drift as well. For example, in an early set of experiments on 2-dimensional Tic-Tac-Toe, random sampling of opponents was found to require *fewer generations* to solve the game than full crossbar testing (Rosin & Belew, 1995). This is surprising: random sampling may save work overall, but it yields noisier fitness information that should lead to a larger number of generations required. It appears now that full crossbar testing was less effective because it did not permit drift to as great an extent.

7.7 Search Difficulty and Coevolutionary Difficulty

Of the two problems studied, 3DTTT seems more difficult in some ways. The game, representation, and genetic operators are all more complex. For consistent success, a larger population and larger sample were required, and a larger total number of games were required to solve the game. It is therefore surprising (see Section 6) that simple coevolution, without any extensions, was able to solve 3DTTT some of the time, since it was completely unable to solve Nim.

We believe that the complex representation and operators used in 3DTTT make search difficult. Operators will often be very destructive, and improved individuals are fairly rare. But these factors would make evolution difficult even in a non-coevolutionary setting, against a fixed set of opponents. In general, 3DTTT is probably a difficult *search* problem.

This stands in contrast to Nim, with its simple, crossover-friendly representation. Operators will rarely be very destructive. Two individuals that play well in different parts of the game graph will often be easy to combine via crossover into a child that plays well in the union of these parts. These suggest that Nim should be an easier *search* problem.

So, there must be other sources of difficulty in Nim that cause coevolution without extensions to fail. For example, the coevolution-specific problems of arms race progress and pedagogy may be particularly severe in Nim. The paths from initial random strategies to perfect strategies might be longer and more difficult to traverse (in a pedagogical sense) than in 3DTTT, making Nim a more difficult *coevolutionary* problem.

We do not yet have a good objective way to separate search difficulty from coevolutionary difficulty, or to measure the coevolutionary difficulty of a problem. This is an important topic for future research.

8 Conclusion

We have identified several factors important in competitive coevolution, and have examined the performance of several new methods in light of these factors. Competitive fitness sharing, shared sampling, and the hall of fame significantly improve coevolutionary performance. Solutions are particularly efficient when these methods are used in conjunction.

Competitive fitness sharing modifies short-term dynamics by selecting hosts that defeat parasites few other hosts defeat. Selection of individuals on this basis creates a diverse population that covers the opposition. Important genotypes that defeat strong parasites are unlikely to be lost. This is important for both testing (when the population serves as parasites) and for future evolution. The use of competitive fitness sharing preserves the coevolutionary equilibria that exist under traditional simple fitness.

Shared sampling selects diverse sets of parasites that test all segments of a host population. In addition, the performance-based nature of shared sampling gives it some of the effects of competitive fitness sharing:

important parasites are sampled more frequently, so that hosts able to defeat them are strongly rewarded. This allows successful coevolution with shared sampling, even when fitness sharing is not used.

The hall of fame preserves individuals from previous generations for later use as parasites. This helps enforce progress in an arms race: successful newly evolved hosts must defeat old parasites as well as current ones. This is important to convergence on optimal solutions. The use of the hall of fame is found to serve this purpose well in practice; it plays a role in coevolution that is distinct from the roles played by shared sampling and fitness sharing.

Timing and pedagogy are important in coevolution. If one side gets far ahead of the other, both sides lose informative fitness and evolution slows. The diversity provided by fitness sharing can help prevent such timing problems, because each population maintains a cover of the opposing population. Additionally, drift may play an important role in correcting timing problems. When a population is far ahead in the arms race, it wins almost all competitions and has little fitness information to guide it, so that it begins to drift. Drifting individuals are less difficult to defeat and can provide a hill for opponents to climb towards defeating individuals that are currently out of reach.

Acknowledgements

Thanks to Apple Computer for equipment donations.

References

Angeline, P.J., & Pollack, J.B. (1993). Competitive environments evolve better solutions for complex tasks. *Proceedings of the Fifth International Conference on Genetic Algorithms.* Morgan Kaufmann.

Anthony, M., Brightwell, G., Cohen, D., & Shawe-Taylor, J. (1992) On exact specification by examples. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory.* ACM.

Axelrod, R. (1989). Evolution of strategies in the iterated prisoner's dilemma. *Genetic Algorithms and Simulated Annealing.* Morgan Kaufmann.

Bedau, M.A., & Packard, N.H. (1991). Measurement of evolutionary activity, teleology, and life. *Artificial Life II.* Addison-Wesley.

Belew, R.K., & Mitchell, M. (1996). *Adaptive Individuals in Evolving Populations: Models and Algorithms.* Addison-Wesley.

Berlekamp, E.R., Conway, J.H., & Guy, R.K. (1982). *Winning Ways for Your Mathematical Plays.* Academic Press.

Cliff, D. & Miller, G. (1995). Tracking the red queen: measurements of adaptive progress in co-evolutionary simulations. *Third European Conference on Artificial Life.* Springer-Verlag.

Darwen, P., & Yao, X. (1996). Automatic modularisation by speciation. *Proceedings of the Third IEEE International Conference on Evolutionary Computation (ICEC'96).* IEEE.

Dawkins, R., & Krebs, J.R. (1979). Arms races between and within species. *Proceedings of the Royal Society of London B* 205:1161.

Epstein, S.L. (1995). Toward an ideal trainer. *Machine Learning* 15:3.

Goldberg, D.E., & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. *Proceedings of the Second International Conference on Genetic Algorithms.* L. Erlbaum Assoc.

Goldman, S.A., & Kearns, M.J. (1991). On the complexity of teaching. *Proceedings of the Fourth Annual Workshop on Computational Learning Theory.* Morgan Kaufmann.

Hillis, W.D. (1991). Co-evolving parasites improve simulated evolution as an optimization procedure. *Artificial Life II*. Addison-Wesley.

Lindgren, K., & Nordahl, M.G. (1994). Artificial food webs. *Artificial Life III*. Addison-Wesley.

Mahfoud, S.W. (1995) Population size and genetic drift in fitness sharing. *Foundations of Genetic Algorithms 3*. Morgan Kaufmann.

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

Mitchell, M., & Forrest, S. (1994). Genetic algorithms and artificial life. *Artificial Life* 1:3.

Oei, C.K., Goldberg, D.E., & Chang, S.J. (1991) *Tournament selection, niching, and the preservation of diversity* (IlliGAL Report 91011). Urbana: University of Illinois, Illinois Genetic Algorithms Laboratory.

Rosin, C., & Belew, R. (1996) A competitive approach to game learning. *Proceedings of the Ninth Annual ACM Workshop on Computational Learning Theory*. ACM.

Rosin, C., & Belew, R. (1995) Finding opponents worth beating: methods for competitive co-evolution. *Proceedings of the Sixth International Conference on Genetic Algorithms*. Morgan Kaufmann.

Smith, R.E., Forrest, S., & Perelson, A.S. (1993). Searching for diverse, cooperative populations with genetic algorithms. *Evolutionary Computation* 1:2.

Sun, Chuen-Tsai, Ying-Hong Liao, Jing-Yi Lu, & Fu-May Zheng. (1994) Genetic algorithm learning in game playing with multiple coaches. *Proceedings of the First IEEE Conference on Evolutionary Computation*. IEEE.

Weaver, L. & Bossomaier, T. (1996, May). *Evolution of neural networks to play the game of Dots-and-Boxes*. Poster session presented at Artificial Life V. Available online at

`http://cs.anu.edu.au:80/people/Lex.Weaver/pub+sem/publications/alife-pa125.ps`