A Game-Theoretic Approach to the Simple Coevolutionary Algorithm

Sevan G. Ficici and Jordan B. Pollack

DEMO Lab—Department of Computer Science
Brandeis University, Waltham Massachusetts 02454 USA
www.demo.cs.brandeis.edu

Abstract. The fundamental distinction between ordinary evolutionary algorithms (EA) and *co*-evolutionary algorithms lies in the *interaction* between coevolving entities. We believe that this property is essentially game-theoretic in nature. Using game theory, we describe extensions that allow familiar mixing-matrix and Markov-chain models of EAs to address coevolutionary algorithm dynamics. We then employ concepts from evolutionary game theory to examine design aspects of conventional coevolutionary algorithms that are poorly understood.

1 Introduction

While formal models of evolutionary algorithm (EA) dynamics burgeon in general [24, 26, 21], *co*-evolutionary algorithms, in particular, still have few formal tools for their analysis—investigations of coevolutionary algorithm dynamics are typically empirical in nature [7, 3, 1, 14, 16]. The reason for this divide stems from the need to formally account for the defining characteristic of coevolutionary algorithms: the *interaction* of coevolving entities. We believe that this property of coevolution is fundamentally *game-theoretic* in nature. Thus, we incorporate notions from *evolutionary game theory* (EGT) [15] and *replicator dynamics* [12] into the familiar mixing-matrix [26] and Markov-chain [24] models of (non-coevolutionary) EAs to arrive at a preliminary, yet principled framework for coevolutionary algorithm analysis.

While we are unaware of other formal coevolutionary algorithm models in the evolutionary computation (EC) community, we have discovered at least two precedents that originate, perhaps not surprisingly, from the economics community, which has come increasingly to study agent-based models of *economic evolution* [4, 19]. Both discuss Markov-chain models of genetic algorithms (GA) (i.e., bit-string encodings) where agent fitness is dependent upon population state. While the game structure assumed in the former study is not explicitly stated, the description is consistent with that of the latter study, which assumes an N-player game, where N is the size of the agent population. Such games typify economic domains, in contrast to the two-player games implied by the pair-wise contests frequently used by the EC community.

Coevolutionary dynamics have also been studied from the perspective of mathematical biology, for example in [10] and [5]. The "street-car" theory of the former study is concerned with reconciling the peculiar constraints of Mendelian genetics with game-theoretic notions of phenotypic stability. In contrast, the latter study presents a continuous-time dynamical systems approach to coevolution that operates strictly in the phenotype realm, and does not assume any constraints due to an underlying genotype structure. Nevertheless, both stress the existence of short-term and long-term time scales in the dynamics. Coming from an entirely different approach, computational learning theory (COLT) has been used to analyze machine learning of competitive (zero-sum) two-player games and construct a specialized form of coevolutionary algorithm [20].

Thus, we see that many sources exist for insight into coevolutionary dynamics. Nevertheless, neither the fields of economics nor biology ultimately concern themselves with modeling coevolutionary *algorithms*, and N-player and zero-sum games do not represent the entirety of coevolutionary domains. Many important aspects of coevolutionary algorithm dynamics remain to be explicated, for the gap between the hypothesized potential of coevolutionary algorithms and realized practice is substantial—the many successes of coevolution (e.g., [11, 22, 13]) are balanced by many irksome modes of failure that commonly recur (e.g., as discussed in [7, 18]). Investigation into these modes of failure has proven to be a challenge. We believe that an understanding of evolutionary game theory and replicator dynamics will help to meet this challenge.

Our fundamental assumption is that we are using coevolution as an optimization method. This assumption has important game-theoretic consequences in phenotype space that exist independently of any underlying genetic representation. These consequences must be appreciated in their own right before the dynamics of genetical constraints can be taken into account. After describing the game-theoretic extensions necessary to allow mixing-matrix and Markov-chain methods to model the "simple *co*evolutionary algorithm" (SCA), we move on to demonstrate how a principled game-theoretic approach can illuminate our understanding of coevolutionary algorithms.

2 The Simple Coevolutionary Algorithm

Building on familiar mixing-matrix and Markov-chain models of the so-called Simple Genetic Algorithm (SGA) [26, 24], we describe extensions that utilize a game-theoretic approach to formalize coevolutionary algorithms. Using the notational convention of Vose [24], the SGA heuristic $\mathcal{G}$ is the composition of the function $\mathcal{M}$ (which includes the mixing matrices that implement the variational operators) with the function $\mathcal{F}$ (which includes the objective function, or evaluator), as shown in Equation 1. The input to $\mathcal{F}$ is a column vector $\mathbf{p}$ in the n-dimensional simplex Λ, where n is the number of possible phenotypes and $\mathbf{p}_i$ is the proportion of agents in the population that have phenotype i. The output of $\mathcal{F}$ is another n-dimensional vector on the simplex representing the fitness of each phenotype. Assuming fitness-proportionate selection, $\mathcal{M}$ performs the appropriate variational operations to produce the phenotype distribution of the next generation. If the population size is infinity, $\mathcal{G}$ represents a *mixing-matrix* model that is sufficient to study the dynamics of the EA. If the population is finite, however, one must distinguish between population states that are representable, given the population's size, and those that are not. This can be accomplished with a *Markov-chain* model. Each representable population state is a state of the Markov chain, and each transition probability is the likelihood of a finite-population approximation of the map $\mathcal{G}$ to produce the various representable population states.

$$\mathcal{G} = \mathcal{M} \circ \mathcal{F} \tag{1}$$

Since the distinguishing feature of coevolution is agent interaction, the necessary extensions revolve around the computation of fitness by the objective function. The modifications we describe below are compatible with mixing-matrix and Markov-chain models. In accord with the discrete-time equations of basic evolutionary game theory [15] and much of current EC practice, our modifications assume generational reproduction and domains where agents interact pair-wise (though, steady-state dynamics and N-player games are not excluded by our general approach). Coevolving agents may exist in a single population (e.g., [14, 1]), or two genetically isolated populations (e.g., [11]).

2.1 The Objective Function $\mathcal{F}$

Single-Population Domains We abstract the domain of agent interaction into an n by n *payoff matrix*, $\mathbf{G}$, where $\mathbf{G}_{i,j}$ gives the *expected* payoff for an agent of phenotype i when played against an agent of phenotype j. We interpret each distinct phenotype as a genetically determined *pure strategy* for the game (domain) in question. Note that a behavior that blends two or more phenotypes into a "mixed" strategy is not excluded—we simply require it to have a genetic basis and can therefore give it the status of a pure strategy, as well.

The *fitness* of strategy i represents the fitness value that each agent (and genotype) playing strategy i receives after *complete mixing*, i.e., pair-wise encounters with every agent in the population (including *itself*, in the case of a finite population [8]). As shown in Equation 3, we first calculate for each strategy i a weighted sum of the payoffs in row i of $\mathbf{G}$, the weights being determined by the phenotypic composition of the population, $\mathbf{p}$; this operation is accomplished by matrix multiplication. Since $\mathbf{G}$ may contain negative payoff values, we add a constant *baseline fitness*, w_0, to the result of the multiplication so that all fitness values are greater than zero; this is to allow normalization, as performed in Equation 2.

$$\mathcal{F}(\mathbf{p}) = \frac{F(\mathbf{p})}{\mathbf{1} \bullet F(\mathbf{p})} \tag{2}$$

$$F(\mathbf{p}) = \mathbf{G}\mathbf{p} + w_0\mathbf{1} \tag{3}$$

$$w_0 = 1 - \min(\mathbf{G}) \tag{4}$$

where '$\bullet$' is inner product.

Two-Population Domains In coevolution between two genetically isolated populations, agents from Population 1 only play against agents from Population 2. Let the column vector $\mathbf{q}$ represent the phenotype (strategy) distribution of the second population. The payoffs for agents from the second population are given by an additional payoff matrix, $\mathbf{H}$. A pair-wise interaction between an agent (from Population 1) playing strategy i, and a second agent (from Population 2) playing strategy j, will yield the payoff $\mathbf{G}_{i,j}$ for the first agent and $\mathbf{H}_{j,i}$ for the second. The calculation of fitness requires two equations, Equations 5 and 6, which assume that every agent in one population plays against every agent in the other population. Because the two populations can have entirely distinct genotype representations, a second set of mixing matrices is required to implement the variational operators. The state space of a two-population Markov chain is the Cartesian product of the two sets of representable population states, $P \times Q, \mathbf{p} \in P, \mathbf{q} \in Q$.

$$F_1(\mathbf{q}) = \mathbf{G}\mathbf{q} + w_0^G \mathbf{1} \tag{5}$$
$$F_2(\mathbf{p}) = \mathbf{H}\mathbf{p} + w_0^H \mathbf{1} \tag{6}$$

2.2 The Selection Operator $\mathcal{S}$

A common assumption of evolutionary game theory, as well as mixing-matrix and Markov-chain models of EAs, is that agents reproduce in proportion to fitness. Nevertheless, many popular alternative selection methods exist that are used both in ordinary EAs and coevolutionary algorithms. We wish our model to accommodate these alternatives. Thus, to Equation 1 we add a new function, $\mathcal{S}$, representing the selection operator, as shown in Equation 7. $\mathcal{S}$ accepts a vector of normalized fitness values and produces a new normalized distribution of agent proportions.

Fitness-proportionate selection is achieved by making $\mathcal{S}$ the identity operator. Linear rank selection, for example, is easily implemented by replacing an agent's fitness value f_i with a new value based upon the agent's fitness rank, r_i: $\mathcal{S}(\boldsymbol{f}) = \frac{R(\boldsymbol{f})}{\mathbf{1} \bullet R(\boldsymbol{f})}$, where R computes the rank of each agent based on fitness, $r_i = n \ldots 1$ (higher fitness results in higher rank values). Implementations of other selection methods for infinite populations are found in [6]. We discuss the effects of alternative selection methods below.

$$\mathcal{G} = \mathcal{M} \circ \mathcal{S} \circ \mathcal{F} \tag{7}$$

3 Evolutionary Game Theory

If we are using coevolutionary algorithms for optimization, and understand the coevolutionary domain to be representable by a payoff matrix, then the appropriate optimality concept is that of the *Nash equilibrium* [2]. This is a strategy that, when used by one player, offers no better alternative to the second player than to use the same strategy. If there exists another alternative that does no worse than the Nash strategy, then the Nash strategy is *weak*, otherwise it is *strict*. Thus, classical Nash *equilibrium* is achieved through the assumption of agent rationality.

In a biological context, however, we cannot appeal to rationality to achieve optima. The central contribution of evolutionary game theory [15] is the substitution of agent rationality with Darwinian selection. In EGT, *replicator dynamics* allow Nash equilibria to be achieved. But, we find that only certain Nash equilibria are attractors of the dynamical system, while others are not.

Evolutionary game theory operates exclusively in phenotype space; the replicator equations perform selection only, since no genotype space is assumed upon which to construct variational operators. This fact has been the focal point of debate between Darwinian adaptionists, who have used EGT with great success in analyzing empirical data, and population geneticists, whose mathematics clearly show that Mendelian genetics can easily contradict Darwinian reasoning. Hammerstein's "streetcar" theory of evolution [10] addresses this apparent paradox and shows how the long-term outcome of coevolution can, in fact, be adequately described on the phenotypic level. Of course, we do not presume this result to apply directly to coevolutionary algorithms, for the genotypic constraints imposed by representations used in EAs differ considerably from those found in Mendelian genetics. Nevertheless, the pressure to achieve optima is clearly found in the selection mechanism, which can be described in a game-theoretic framework. Thus, we suggest that game-theoretic properties must underlie the behavior of coevolutionary algorithms if they are to optimize successfully.

3.1 Polymorphic Attractors

Polymorphic attractors [15] can occur in single-population coevolution when the game is not constant-sum. A polymorphic attractor is a point attractor of the replicator dynamical system where the population contains more than one phenotype: $\hat{\mathbf{p}}_i > 0$ for ≥ 2 values of i. All phenotypes present in a polymorphic attractor $\hat{\mathbf{p}}$ (and hence all agents) receive the same fitness: $\forall \hat{\mathbf{p}}_i > 0 : \mathbf{f}_i = c$. The proportions with which the strategies appear in the population can be interpreted as a mixed Nash equilibrium. Surprisingly, a pure strategy with the same behavior as the polymorphism need not be an attractor of replicator dynamics [15]. This is because replicator dynamics act upon individual fitness rather than population-wide fitness.

Thus, we see that the simple coevolutionary algorithm, as with the simple genetic algorithm, operates with an *agent-centric* view of performance. This makes recognizing (as opposed to achieving) polymorphic attractors problematic. For example, the Hawk-Dove game is a simple two-strategy, non-constant sum game that has a single, and polymorphic, attractor where $\hat{\mathbf{p}}_{Hawk} = \frac{7}{12}$ and $\hat{\mathbf{p}}_{Dove} = \frac{5}{12}$. How is this equilibrium to be comprehended by an agent-centric view? To begin, sorting the population of agents by fitness has an ill-defined result due to ties—all agents receive the same fitness, and the "best" agent, with respect to sorting, could be either a Hawk or a Dove. Of course, neither strategy alone constitutes an optimal solution. Not only are both strategies needed, but they are needed in a particular proportion. The "solution" to the Hawk-Dove game is not a single strategy, but an *ecology*.

We may argue, then, that a coevolutionary algorithm should report the population's phenotypic proportions once fitness converges. This is straight-forward if the genotype-phenotype mapping is known to be one-to-one. But, with real-world representations, a particular phenotype is often achievable through a variety genotypes. Since we only have knowledge of an agent's genotype, the question of phenotypic equivalence between two agents is generally undecidable. How, then, are we to distinguish between fitness convergence with phenotypic polymorphism, on the one hand, and genuine phenotypic convergence, on the other? This distinction is not important in the Simple Genetic Algorithm—any optimal phenotype will do. But, in coevolution, polymorphic optima are achieved through specific phenotype proportions, which must remain intact.

3.2 Time Scales of Selection and Search

Given an initial population state, $\mathbf{p}^0$, what will iteration of the selection operator on $\mathbf{p}^0$ (without variation) do in the limit? Because variation operators are not applied, we know that phenotypes that are absent in $\mathbf{p}^0$ cannot appear later. Also, we know that some phenotypes present in $\mathbf{p}^0$ may be eliminated by selection. In ordinary evolution, the limit behavior is the point attractor $\hat{\mathbf{p}}$, which is composed exclusively of the most fit phenotype in $\mathbf{p}^0$ (assuming the absence of neutrality). In *co*evolution, however, the limit behavior can be very different: assuming fitness-proportionate selection, the attractor may be a limit cyclic rather than a fixed point. Further, if a point attractor $\hat{\mathbf{p}}$ exists and the game is not constant-sum, then $\hat{\mathbf{p}}$ may easily be composed of a phenotype *other* than the most fit phenotype in $\mathbf{p}^0$.

Linear stability analysis [23] tells us that the strategies *in support* of a coevolutionary point attractor—that is, all strategies i where $\hat{\mathbf{p}}_i > 0$—must be the highest scoring strategies in all population states within some epsilon of the attractor. Otherwise, the attractor could not be locally stable. But, the region within which this is true need not be the attractor's entire basin of attraction. Indeed, population states that are outside of the epsilon, $|\hat{\mathbf{p}} - \mathbf{p}| > \epsilon$, but still within the basin of attraction, may contain very many phenotypes that out-score those in support of $\hat{\mathbf{p}}$, but which eventually drive each other to extinction, leaving only the strategies in $\hat{\mathbf{p}}$. Fitness values in this region of population-state space are "deceptive" in the sense that they favor phenotypes that will eventually be selected against.

For example, in randomly generated non-constant sum games of 100 strategies, we can easily find examples of an initial population state, $\mathbf{p}^0$, that leads to an attractor composed of a single strategy, $i = Win : \hat{\mathbf{p}}_i = 1.0, \forall i \neq Win : \hat{\mathbf{p}}_i = 0$, but that exhibits fitness deception at the beginning of its orbit towards the attractor. In one instance of $\mathbf{p}^0$ that we have found, the eventual "winner" strategy $\hat{\mathbf{p}}_{Win}$ is actually out-scored by 54 of the 100 game strategies and over 50% of the population in $\mathbf{p}^0$, as well as in the next ten generations (iterations of selection), $\mathbf{p}^1 \ldots \mathbf{p}^{10}$. Indeed, some 39 generations are required before $\hat{\mathbf{p}}_{Win}$ becomes the most fit in the population. Other initial conditions in the neighborhood of $\mathbf{p}^0$ yield similar behavior. This result is in stark contrast to the dynamics of selection in ordinary evolution, where the winning strategy of the attractor $\hat{\mathbf{p}}_{Win}$ is made apparent after a single round of evaluation.

This example raises a concern about the efficiency with which conventional coevolutionary algorithms may perform search in non-constant sum games. In each generation, decisions are made regarding which strategies should be kept and form the basis for future search; these decisions can easily be misled by fitness values. Note that this form of "deception" does not involve the genotype space at all (as it does in GA trap functions, for example)—it exists purely within the game-theoretic relationships of phenotype proportions. Strategies that exhibit superior performance only in transient population states, $\mathbf{p}^{0 \ldots n}$, may be less useful guideposts for search than the strategies in $\hat{\mathbf{p}}$, which at least have some stability properties with respect to $\mathbf{p}^{0 \ldots n}$. Should the application of variation operators be delayed until fitness values are "believable"? If so, how should the coevolutionary algorithm be modified to improve efficiency yet avoid premature convergence?

3.3 Dynamics of Alternative Selection Methods in Coevolution

The constraint of finite populations creates at least three problems for the canonical fitness-proportionate roulette-wheel selection method used in EAs. First, the multinomial distribution of roulette-wheel operation introduces a great deal of noise into the selection process. Second, this noise makes resolving small differences in genotype fitness impossible for realistic population sizes. Third, an excessive difference in fitness between the best individuals and the rest of the population can cause premature convergence. These problems have lead to the development of many alternative selection methods for evolutionary algorithms, such as *truncation selection*, (μ, λ)-ES selection, and *ranking* [17]. We have used EGT to test these selection methods and have shown that none of them are able to attain polymorphic attractors, *even if the population is infinite* [6]. Instead, these selection methods exhibit attractors (sometimes chaotic) that are unrelated to the game. Thus, implementation decisions that may be sound for ordinary evolutionary algorithms can be pathological in an idealized coevolutionary context (where the search problem is solved, leaving only the problem of selection), and therefore dubious in a real-world algorithm. Indeed, we have shown [8] that the use of truncation selection is one reason why the simulations by Fogel, et al [9] lead them to claim that EGT loses predictive power in finite populations.

3.4 Population Structures

Coevolutionary algorithms that use two genetically isolated populations typically involve asymmetric games, where agents from one population can only play against agents from the other [11, 7]. Evolutionary game theory tells us that asymmetric games never result in polymorphic equilibria [12]. This feature is actually not a property of the games themselves, but rather a property of the two-population structure, which asymmetric games happen to require. Thus, if we use a two-population structure on a symmetric game that has a polymorphic attractor, then the polymorphic attractor will disappear. In the case of the Hawk-Dove game, a single population of agents yields the polymorphic attractor of $\frac{7}{12}$ Hawks and $\frac{5}{12}$ Doves. But, when the game is played between two separate populations, one population will converge to all Hawks and the other to all Doves.

Multiple-population structures are often used in evolutionary algorithms (including coevolution) to maintain genetic diversity (e.g., [22]). In a coevolutionary context, however, the influence of population structure can transcend the process of genotypic search to exert an independent (and unintentional) effect on the process of phenotype selection. Evolutionary game theory shows that, even when the search problem is solved *a priori*, different population structures may lead to different outcomes, even though the game remains the same.

3.5 Local Optima

When variational operators are unable to reach new phenotypes that are sufficiently good to survive selection with respect to the current population (and thereby allow search to continue), the state of the evolutionary system is at a *local optimum*. In stationary fitness environments, local optima are easy to conceptualize. In coevolutionary domains, however, fitness landscapes are dynamic. In this case, game theory provides a natural way of describing a coevolutionary local optimum: it is a *search-constrained attractive Nash equilibrium*. That is, the coevolutionary state is a "best reply" to itself only to the extent that all *locally* reachable population states are inferior to the Nash state—all paths to superior population states are blocked by the convergent force of selection.

3.6 Dominating Strategies

An important concept in game theory is that of *dominance* [2]. Strategy s_a (pure or mixed) is said to dominate strategy s_b if s_a does as well or better than s_b against all strategies in the game and strictly better against at least one: $s_a \succ s_b \Leftrightarrow \forall i: \; G(s_a, s_i) \geq G(s_b, s_i) \;\wedge\; \exists j: \; G(s_a, s_j) > G(s_b, s_j)$. If $s_a \succ s_b$, but there exists some strategy against which s_a does no better than s_b, then s_a *weakly dominates* s_b. Otherwise, s_a *strictly dominates* s_b. If s_a is a pure strategy and strictly dominates s_b, then the replicator dynamics will remove s_b from the population. This is not necessarily true for weak dominance [12].

That replicator dynamics removes strategies that are strictly dominated (by pure strategies, in the case of non-overlapping generations [25]) suggests that it is performing an operation akin to multi-objective optimization, where every strategy represents an objective of the coevolutionary problem. That is, in comparing two strategies, s_a and s_b, the salient question is not whether s_a beats s_b in the game, or out-scores s_b when played against a series of opponents, but rather whether s_a *Pareto dominates* s_b with respect to a given set of opponent strategies (including themselves). (The notion of Pareto dominance, as used in multi-objective optimization, is not to be confused with Pareto dominance with respect to game payoffs, as used in game theory.)

This observation leads to the realization that multi-objective optimization techniques may be explicitly applied to coevolutionary domains *without* the use of replicators, such that both strictly as well as weakly dominated strategies are rejected in favor of Pareto-optimal strategies. Though such an approach may no longer qualify as a coevolutionary algorithm, we suspect that it may nevertheless provide a powerful method of optimizing coevolutionary domains.

4 Conclusion

We have argued for the relevance of evolutionary game theory to the study of coevolutionary algorithms. We have presented game-theoretic extensions that allow mixing-matrix and Markov-chain models of EAs to address coevolutionary algorithm dynamics. Additionally, we have shown that a game-theoretic view of coevolutionary algorithm design and operation reveals that the simple coevolutionary algorithm 1) cannot recognize polymorphic equilibria, 2) can be made inefficient by a newly-recognized form of fitness deception, 3) exhibits pathological behaviors if certain commonly-used selection methods are employed, and 4) can have different attractors depending upon population structure. Finally, we have used game-theoretic concepts to formulate a notion of coevolutionary local optima, and to understand coevolution as a form of multi-objective optimization. We believe that EGT can help with the design and validation of new search methods for coevolutionary domains. This is the subject of our current work.

5 Acknowledgments

The authors thank members of the DEMO Lab, particularly Richard Watson (for fruitful discussions regarding connections between coevolution, multi-objective optimization, and Pareto optimality), and Ofer Melnik.

References

1. P. J. Angeline and J. B. Pollack. Competitive environments evolve better solutions for complex tasks. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 264–270. Morgan Kaufmann, 1994.
2. K. Binmore. *Fun and Games: A Text on Game Theory*. D.C. Heath, 1992.
3. D. Cliff and G. F. Miller. Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations. In F. Moran et al., editors, *Third European Conference on Artificial Life*, pages 200–218. Springer Verlag, 1995.
4. H. Dawid. A markov chain analysis of genetic algorithms with a state dependant fitness function. *Complex Systems*, 8:407–417, 1994.

5. U. Dieckmann and R. Law. The dynamical theory of coevolution: A derivation from stochastic ecological processes. *Journal of Mathematical Biology*, 34:579–612, 1996.
6. S. G. Ficici, O. Melnik, and J. B. Pollack. A game-theoretic investigation of selection methods used in evolutionary algorithms. In A. Zalzala et al., editors, *Proceedings of the 2000 Congress on Evolutionary Computation*. IEEE Press, 2000.
7. S. G. Ficici and J. B. Pollack. Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states. In C. Adami et al., editors, *Artificial Life VI (1998)*, pages 238–247. MIT Press, 1998.
8. S. G. Ficici and J. B. Pollack. Effects of finite populations on evolutionary stable strategies. In L. D. Whitley, editor, *Proceedings of the 2000 Genetic and Evolutionary Computation Conference*. Morgan-Kaufmann, 2000.
9. D. B. Fogel, G. B. Fogel, and P. C. Andrews. On the instability of evolutionary stable strategies. *BioSystems*, 44:135–152, 1997.
10. P. Hammerstein. Darwinian adaptation, population genetics and the streetcar theory of evolution. *Journal of Mathematical Biology*, 34:511–532, 1996.
11. D. Hillis. Co-evolving parasites improves simulated evolution as an optimization procedure. In C. Langton et al., editors, *Artificial Life II*. Addison-Wesley, 1991.
12. J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.
13. H. Juillé and J. B. Pollack. Coevolving the "ideal" trainer: Application to the discovery of cellular automata rules. In J. R. Koza et al., editors, *Proceedings of the Third Annual Genetic Programming Conference*, pages 519–527. Morgan Kaufmann, 1998.
14. K. Lindgren. Evolutionary phenomena in simple dynamics. In C. Langton et al., editors, *Artificial Life II*, pages 295–312. Addison-Wesley, 1991.
15. J. Maynard-Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.
16. N. Meuleau and C. Lattaud. The artificial evolution of cooperation. In J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and S. D., editors, *Artificial Evolution (AE 95)*, pages 159–180. Springer-Verlag, 1995.
17. M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1996.
18. J. B. Pollack and A. D. Blair. Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, 32(3):225–240, 1998.
19. T. Riechmann. Learning and behavioral stability: An economic interpretation of genetic algorithms. *Journal of Evolutionary Economics*, 9:225–242, 1999.
20. C. D. Rosin. *Coevolutionary Search Among Adversaries*. PhD thesis, University of California, San Diego, 1997.
21. J. Shapiro, A. Prügel-Bennet, and M. Rattray. A statistical mechanical formulation of the dynamics of genetic algorithms. In T. C. Fogarty, editor, *Evolutionary Computing (AISB Workshop)*, pages 17–27. Springer-Verlag, 1994.
22. K. Sims. Evolving 3d morphology and behavior by competition. In *Artificial Life IV*, pages 28–39. MIT Press, 1994.
23. S. H. Strogatz. *Nonlinear Dynamics and Chaos*. Addison Wesley, 1994.
24. M. D. Vose. *The Simple Genetic Algorithm*. MIT Press, 1999.
25. J. Weibull. *Evolutionary Game Theory*. MIT Press, 1995.
26. L. D. Whitley. A review of models for simple genetic algorithms and cellular genetic algorithms. In V. Rayward-Smith, editor, *Applications of Modern Heuristic Methods*, pages 55–67. Alfred Waller, 1995.