Ideal Evaluation from Coevolution

Edwin D. de Jong edwin@cs.brandeis.edu*
Jordan B. Pollack pollack@cs.brandeis.edu
DEMO Lab, Volen National Center for Complex Systems, Brandeis University MS018, 415 South street, Waltham MA 02454-9110, USA

Abstract

In many problems of interest, performance can be evaluated using tests, such as examples in concept learning, test points in function approximation, and opponents in game-playing. Evaluation on all tests is often infeasible. Identification of an accurate evaluation or fitness function is a difficult problem in itself, and approximations are likely to introduce human biases into the search process. Coevolution *evolves* the set of tests used for evaluation, but has so far often led to inaccurate evaluation.

We show that for any set of learners, a Complete Evaluation Set can be determined that provides ideal evaluation as specified by Evolutionary Multi-Objective Optimization. This provides a principled approach to evaluation in coevolution, and thereby brings *automatic* ideal evaluation within reach. The Complete Evaluation Set is of manageable size, and progress towards it can be accurately measured. Based on this observation, an algorithm named DELPHI is developed. The algorithm is tested on problems likely to permit progress on only a subset of the underlying objectives. Where all comparison methods result in overspecialization, the proposed method and a variant achieve sustained progress in all underlying objectives. These findings demonstrate that ideal evaluation may be approximated by practical algorithms, and that accurate evaluation for test-based problems is possible even when the underlying objectives of a problem are unknown.

Keywords

Coevolution, Pareto-Coevolution, accurate evaluation, Evolutionary Multi-Objective Optimization, underlying objectives, Pareto-hillclimber, over-specialization.

1 Introduction

1.1 Evaluation

For some of the most interesting problems that evolutionary computation might address, tests whose outcomes reflect some of the qualities of individuals can readily be performed, while the precise underlying objectives reflected by such tests are unknown. Examples include concept learning, function approximation, game-playing, as well as open-ended domains such as the evolution of complex behavior. Human designed fitness functions for such *test-based problems* will typically be inaccurate, and thereby limit the potential of evolutionary computation to address these problems. Selecting a fixed representative set of tests is often infeasible due to the large number of possible tests. An important question therefore is: how may *accurate evaluation* for test-based problems be achieved?

Coevolution (Barricelli, 1962, 1963; Axelrod, 1987; Hillis, 1990) evaluates individuals on an *evolving* set of tests, and may thereby in principle provide accurate evaluation.

*Current address: Utrecht University, Decision Support Systems Group. E-mail: dejong@cs.uu.nl

Coevolution has already led to several successful results (Hillis, 1990; Sims, 1994; Juillé & Pollack, 1996; Pollack & Blair, 1998). So far however, evaluation in coevolution has often been far from accurate, as indicated by problems such as over-specialization, Red Queen dynamics, and disengagement (Cliff & Miller, 1995; Watson & Pollack, 2001). In this article, we investigate how coevolution can be used to achieve accurate evaluation. This significant aim is approached by viewing test-based problems from the perspective of Evolutionary Multi-Objective Optimization (EMOO) (Deb, 2001), and considering what tests are required to determine who dominates who in a given population of learners.

To illustrate the difficulty of accurate evaluation, we consider the problem of selecting a move in a chess game. If the optimal Minimax (Von Neumann, 1928) values for this problem were available, it would be sufficient to compare the values of the few tens of board positions reachable via one of the currently available moves. This is in sharp contrast with the actual situation, where value estimates of the board evaluation function must be refined by means of extensive look-ahead search.[1] This substantial amount of required additional computation suggests that board evaluation functions used in current machine chess must be quite far removed from the optimal (Minimax) values.

In test-based problems, the quality of an individual is determined by its performance on a number of tests. Typically, the set of all possible tests for problems of this kind is very large, making it infeasible to evaluate individuals on all tests. As the above example illustrates, the definition of an accurate domain-specific fitness function is equally problematic, as an accurate specification of the quality of all possible individuals requires highly detailed knowledge of a problem. Moreover, the use of a scalar fitness function presumes that individuals can be ranked on a single dimension of performance. Such rankings cannot yield complete information when quality is governed by multiple objectives.

1.2 Underlying Objectives

A useful notion for test-based problems is that of the *underlying objectives* of a problem. The term *objective* as used in Evolutionary Multi-Objective Optimization refers to an indicator of quality returning an element from an ordered set of scalar values, such as a real number. For any test-based problem, a set of *underlying objectives* exists such that knowledge of the objective values of an individual is sufficient to determine the outcomes of all possible tests. The existence of a set of underlying objectives is guaranteed, as the set of all possible tests itself satisfies this property.

The notion of underlying objectives becomes more meaningful however when a *limited* set of objectives with this property can be identified; if a small set of underlying objectives exists, it represents important information about the structure of a problem. The smallest possible size for the set of underlying objectives can be viewed as the inherent dimensionality of a problem, and the nature of such a minimal set of underlying objectives may provide important information about the structure of a problem. Recent work by Bucci and Pollack (2003a, 2003b) provides a formal discussion of closely related notions based on *partial order decomposition*.

Samuel (1959) notes that terms for the evaluation polynomial of his learning checkers player should ideally be generated by the learning program itself, and mentions the idea of an orthogonal set of terms to be used in this evaluation polynomial. This idea is closely related to the notion of underlying objectives. When viewed in this way, it

[1] State-of-the-art chess programs currently consider from 2 million up to 200 million positions per second.

appears that many apparently single-objective problems may in fact be multi-objective problems. If this is so, then many problems currently addressed using one-dimensional fitness functions may benefit from taking a multi-objective approach. Another indication in this direction is given by recent research in which modeling single-objective problems as multi-objective problems is proposed as a technique to improve search performance (Knowles, Watson, & Corne, 2001). Here, it will be shown that coevolution offers a means to address the multi-objective problem that underlies a given test-based problem.

1.3 Coevolution

In order to discuss coevolution, some terminology will first be introduced. We will use the term *learner* for individuals whose performance we wish to optimize, and *evaluator* for tests that can be applied to learners. Learner and evaluator are roles individuals take on in the context of an algorithm, and a single individual may take on both roles, as occurs in self-play or single population coevolution. The learner and the evaluator can be of similar nature, as in two-player games, or the evaluator may be a test case, such as a test sequence for a sorting network or a test point in function approximation. The test represented by an evaluator E is applied to a learner A by means of an *interaction function* $G(A, E)$ which returns a scalar outcome, such as win/lose or a real number. We consider any setup where individuals are evaluated based on interactions with a (co-)evolving set of evaluators as coevolution. Most if not all instances of coevolution may be fruitfully viewed in this way if the notion of test is interpreted broadly as a procedure that reveals information about an individual. In coevolution problems, changes to the set of evaluators can affect the apparent ranking of the learners.

Samuel's checkers player (1959) repeatedly replaces the evaluator with the learner, and is the first known instance of what may be called co-learning, distinguished from coevolution by its use of directed changes. An early instance of coevolution in a computer simulation is by Barricelli (1962). Attention for coevolution only increased after the appearance of several later papers (Axelrod, 1987; Miller, 1989, 1996; Hillis, 1990; Koza, 1992; Lindgren, 1992; Kauffman & Johnsen, 1992a). Of these, the work by Hillis marked the first time a complex problem was addressed by coevolution, and was probably most influential in spurring further research into coevolution.

Coevolution has meanwhile led to a number of tantalizing results in domains including the evolution of complex behavior (Sims, 1994), pursuit and evasion (Koza, 1991; Reynolds, 1994; Miller & Cliff, 1994; Ficici & Pollack, 1998; Cliff & Miller, 1996), robot behavior (Floreano, Nolfi, & Mondada, 1998; Østergaard & Lund, 2003), exploration by identifying predictable novelty (Schmidhuber, 1999), communication development (Werner & Dyer, 1991; Ficici & Pollack, 1998), function approximation and classification (Pagie & Hogeweg, 1998; Juillé & Pollack, 1996; Paredis, 1996), density classification using cellular automata (Ficici & Pollack, 2001b; Juillé & Pollack, 1998; Pagie & Hogeweg, 2000; Paredis, 1997), job shop scheduling (Husbands & Mill, 1991), backgammon (Pollack & Blair, 1998), Tron (Funes, 2001), and Go (Rosin, 1997; Lubberts & Miikkulainen, 2001).

A factor that may explain successful results and interest in coevolution is the avoidance of biases resulting from the use of a human-designed fitness function; the dynamic evaluation in coevolution may in principle lead to 'arms-races' and open-ended evolution (Sims, 1994; Ficici & Pollack, 1998; Nolfi & Floreano, 1998; Stanley & Miikkulainen, 2002). Other factors proposed to explain positive results with coevolution include the ability to escape local optima due to the changing set of evaluators

(Hillis, 1990; Pagie & Hogeweg, 1998) and the use of resource-sharing methods (Werfel, Mitchell, & Crutchfield, 2000).

A particularly interesting application of coevolution is the idea of evolving modules, evaluated separately based on their ability to contribute to a larger whole. This results in dependencies between modules and complete individuals. Such setups are thus forms of coevolution, and many examples of this are available (Angeline & Pollack, 1994; Moriarty & Miikkulainen, 1998; Gomez & Miikkulainen, 1999; Potter & De Jong, 2000; Rosca & Ballard, 1996; Ahluwalia & Bull, 2001; Watson & Pollack, 2003; De Jong, 2003).

1.4 Issues in Coevolution

In spite of its successes and promise, coevolution has not yet developed into a reliable problem solving technique. Problems that have been reported include disengagement, over-specialization, and intransitivity.

Disengagement means that for one or more objectives, the performance levels represented by evaluators are outside the range of learner performance. This may occur when evaluators testing on some underlying objective disappear from the evaluator population, or pose too high or too low a threshold to distinguish between learners. Disengagement leads to a *loss of gradient* (Watson & Pollack, 2001). As a result, learners cannot be evaluated according to all underlying objectives.

Disengagement will generally lead to over-specialization. Over-specialization, also known as focusing (Watson & Pollack, 2001), refers to situations where learners progress on *some* of the underlying objectives, while not progressing on the remaining objectives.

Intransitivity signifies that the preference relation between learners is not transitive.[2] For example, first B appears preferable to A, then C appears preferable to B, but next A appears preferable to C. For a fixed set of evaluators, intransitive relations between learners cannot occur. This can be seen as follows: for any evaluator E, the interaction outcomes $G(A, E)$, $G(B, E)$ and $G(C, E)$ are fixed, scalar values (e.g. real numbers), and no assignment of values exists for which simultaneously $B > A$, $C > B$, and $A > C$. Similarly, when multiple evaluators are used, a fixed set of evaluators leads to fixed vectors of outcomes, and can therefore not lead to intransitive relationships. Accordingly, intransitivity is the result of a changing set of evaluators, or more specifically the loss of evaluators for some of the underlying objectives.

A changing set of evaluators, whether related to intransitivity or not, can lead evolution to test on only a subset of the underlying objectives, causing learners to focus on different objectives over time. Since the performance in untested objectives is likely to degrade, learners may continuously change without making overall progress, possibly resulting in cyclic behavior. This phenomenon is known as Red Queen dynamics (Cliff & Miller, 1995; Pagie & Hogeweg, 2000) or mediocre stable states (Angeline & Pollack, 1993).

In summary, the above observations point to the need to perform accurate evaluation. A lack of evaluators for some underlying objective is a form of inaccurate evaluation. Accurate evaluation thus requires evaluating learners on *all* underlying objectives. How this may be achieved is the central question of this article. We will now discuss previous approaches aimed at improving the accuracy of evaluation in coevolution.

[2] A relation R is transitive if and only if $aRb \wedge bRc \implies aRc$.

1.5 Towards Accurate Evaluation

Hillis (1990) used evaluators that consisted of input sequences and were called *parasites*. The parasites were used to test sorting networks on different aspects of their performance. Juillé (1999) describes an *ideal trainer* (Epstein, 1994) as an evaluation function that exposes learners to gradient, and presents problems of increasing difficulty. The ideal trainer is described in an abstract manner, and its instantiation requires a domain-specific function evaluating the difficulty of tests.

As a step towards *automatic* ideal evaluation, Rosin (1997) introduced the notion of a *teaching set*. For any imperfect learner from a set of learners, the teaching set contains a teacher defeating this learner, thus detecting any weaknesses in the learners. In the related *covering competitive algorithm* (Rosin, 1997), a new strategy must defeat *all* previous opposition. For problems with multiple underlying objectives, most improvements to the population do not meet this strict criterion, and the criterion is thus overly strict for problems with this property.

Evolutionary Multi-Objective Optimization (EMOO) offers a principled way to compare individuals in the presence of trade-offs, based on the criterion of Pareto-dominance. The earliest reference connecting coevolution and multi-objective optimization problems of which we are aware is in work by Husbands (1994). Juillé (1999) uses the term *objectives* to refer to test cases.

The specific and important idea of viewing the outcomes of a learner's interactions as objectives in the sense employed by EMOO is called Pareto-Coevolution (Ficici & Pollack, 2000; Watson & Pollack, 2000). Since its recent inception, Pareto-Coevolution has been used in several papers (Ficici & Pollack, 2001a, 2001b; Noble & Watson, 2001; Watson & Pollack, 2003; De Jong, 2003). The upcoming of the Pareto-Coevolution paradigm is another indication that problems that have been viewed as having one objective or fitness function may often be multi-objective problems, and thus benefit from being treated as such.

Apart from the multi-objective perspective of Pareto-Coevolution employed here, formalisms employed in the study of coevolution include evolutionary game theory (Maynard Smith, 1982; Ficici, Melnik, & Pollack, 2000; Ficici & Pollack, 2000, 2001a), order theory (Bucci & Pollack, 2002, 2003b), Markov models (Bull, 2001; Ficici & Pollack, 2000), and NK-landscapes (Olsson, 2001; Kauffman & Johnsen, 1992b). Finally, a crucial notion introduced by Ficici (2001b) is that of *distinctions*. Distinctions take into account the detectable strengths of learners in addition to their weaknesses. By making distinctions between learners, evaluators can provide a gradient for learning.

1.6 Ideal Evaluation

We will define ideal evaluation as evaluation according to all underlying objectives, and study the question of whether coevolution can evaluate according to this ideal evaluation function. We combine Ficici's notion of distinctions (2001b) with Rosin's idea of a complete set of tests to arrive at the concept of a *Complete Evaluation Set*. The resulting set of evaluators is sufficient to detect any differences between learners that may be relevant to selection under the EMOO framework. We show that by using distinctions as objectives for evaluators, automatic construction of the ideal evaluation function can in principle be achieved.

The Complete Evaluation Set, detecting all differences between learners relevant to learner selection, was first described in a technical report (De Jong & Pollack, 2002).[3]

[3] This technical report is an earlier version of the present article.

Other work from our lab, developed in parallel with the approach presented here, uses the mathematical formalism of pre-orders to model concepts from Pareto-Coevolution (Bucci & Pollack, 2002). Within this framework, a concept related to the Complete Evaluation Set defined here has been described; the *set of maximally informative* tests or evaluators (Bucci & Pollack, 2002, 2003b) also makes all distinctions relevant to learner selection. While closely related in concept, the two sets are distinct; the Complete Evaluation Set that will be used here is a *maximally informative set* of evaluators. This provides the property that its required size is bounded and small, which is needed for the approximation of the set by practical coevolutionary algorithms.

We employ the definition of the Complete Evaluation Set to show that ideal evaluation can be approximated by practical algorithms, and develop one such algorithm. The feasibility of the approach is demonstrated in experiments including comparisons with a range of other coevolutionary methods. On several abstract test problems, the proposed method is found to lead to stable progress on all underlying objectives, while other methods over-specialize and progress on only a subset of the underlying objectives. The experiments suggest that methods for coevolution have so far not evaluated according to all underlying objectives; addressing this issue may be vital in developing coevolution into a generally applicable problem solving technique.

1.7 Organization

This article is organized as follows. Section 2 defines an ideal evaluation function, and describes our model of coevolutionary methods, in which the evaluation of individuals is based on interactions with other evolving individuals. Section 3 shows how the ideal evaluation function can in principle be constructed from these interactions; the proof is given in appendix A. In Section 4, a class of algorithms is described where this principle is employed to approximate the ideal evaluation function. In Section 5, an experimental setup for testing algorithms of this kind is described. This setup is used in Section 6 to develop a specific algorithm for Pareto-Coevolution by combining the principle for evaluation with heuristics for search. Experimental results with the algorithm are presented in Section 7. Section 8 discusses the method in the light of known issues in coevolution, and Section 9 concludes the article. Following the conclusions section, a list of symbols and the URL for a MATLAB® implementation of the algorithm are provided.

2 Evaluation in Coevolution

In this section, we first describe what the ideal evaluation function for a problem would be when multiple objectives underlie performance. Coevolutionary algorithms do not have access to these underlying objectives, but must use the outcomes of interactions between evolving individuals to provide an evaluation function. We will demonstrate that based on this information, it is possible to construct an evaluation function that is precisely equivalent to the ideal evaluation function.

2.1 An Ideal Evaluation Function

We are concerned with test-based problems, i.e. problems where individuals are evaluated using tests. Individuals in such problems can be characterized by a number of underlying objectives, each of which represents some aspect of the quality of the individual. This description includes as a special case the situation where individuals are characterized by a single fitness value.

Evaluation functions for single-fitness problems calculate a single value that expresses the quality of an individual, which, after scaling, leads to a fitness value that indicates the probability of selection. When multiple underlying objectives are involved, the relative strengths and weaknesses of individuals cannot be expressed by a single value unless strong restrictions are imposed, such as the assumption that quality is a weighted combination of the objectives. To avoid such assumptions, we may employ an evaluation function F_{ideal} that determines for any pair of individuals a and b whether a is to be preferred over b or not.

An objective can be any function such that the quality of an individual increases monotonically in the objective, all other objectives being equal. Note that no specific relation is assumed between the underlying objectives and the genetic representation (genotype) of individuals; the approach is thus applicable to any problem in which performance can be evaluated based on tests. Furthermore, without loss of generality, we may assume that all objectives are to be maximized, as opposed to minimized.

The most general evaluation function makes no assumptions other than those made so far. The preferences following from these assumptions are that when an individual a has at least the same value as an individual b for every objective and higher values for one or more objectives, then a is preferred over b. Other than this, no preferences are specified; if a has a lower value for any objective, it will not be preferred over b. Thus, whenever a is preferable to b regarding some objective but b is preferable regarding others, the evaluation function does not attempt to estimate which individual provides a better trade-off between the objectives; rather, it signals no preference, so that a and b will both be maintained if possible. This description corresponds precisely to the Pareto-dominance relation which forms the basis of Evolutionary Multi-Objective Optimization (EMOO).

Definition 1 (Pareto-dominance) *An individual a* dominates *another individual b with respect to a set of objectives O if:*

$$\underset{O}{dom}(a,b) \iff \forall i : O(a,i) \geq O(b,i) \ \wedge \ \exists i : O(a,i) > O(b,i) \tag{1}$$

where $O(x,i)$ returns the value of the i^{th} objective of x, $1 \leq i \leq n$, and n is the number of objectives contained in O.

The ideal evaluation function is the evaluation function that would result from using the unknown underlying objectives U as the objectives in evaluation. Following the above principle of using Pareto-dominance as the evaluation function for multi-objective problems, this results in the following ideal evaluation function:

$$F_{\text{ideal}}(a,b) = \underset{U}{dom}(a,b) \tag{2}$$

If a global optimum exists, i.e. an individual that has the maximum possible value for each objective, then the ideal evaluation function will prefer it over any other individuals. If different individuals achieve high values for different underlying objectives, the function will discard all individuals except those for which no objective can be improved without reducing performance on another objective. This leads to a front of solutions that trade off the different underlying objectives in different ways. Thus, the solution concept employed here is that of the Pareto-optimal set with respect to the underlying objectives of the problem. For an introduction into Evolutionary Multi-Objective Optimization (EMOO), a number of sources are available (Fonseca & Fleming, 1993; Srinivas & Deb, 1994; Fonseca & Fleming, 1995; Van Veldhuizen, 1999; Coello, 2000; Deb, 2001).

2.2 Coevolution: Interactions as a Basis for Evaluation

Since the underlying objectives of a problem are generally unknown, selection in coevolution does not have direct access to the ideal evaluation function. In test-based problems, decisions must instead be based on the outcomes of tests, i.e. interactions between learners and evaluators, and coevolution evolves this set of evaluators. Here, we prove the existence of a small evaluator set that provides all information required to construct the ideal evaluation function.

We distinguish between learners and evaluators. The aim of learners is to maximize the outcomes of their interactions with evaluators. The aim of the evaluators is to provide a basis for learning, such that learner adaptation results in maximization of the underlying objectives. The set of all possible learners will be denoted as $\mathbb{L}$, and the set of all possible evaluators as $\mathbb{E}$. A particular set of learners will be denoted as L, and a particular set of evaluators as E.

All interactions are assumed to be pairwise. An interaction is a function $G : \mathbb{L}\text{x}\mathbb{E} \rightarrow \mathbb{O}$ which accepts a learner and an evaluator. It returns an outcome for the learner from some ordered set of values $\mathbb{O}$. Outcomes may for example be real valued numbers such as (inverted) errors or scores, or ordered labels such as pass/fail, or win/draw/lose. An interaction $G(a, e)$ may be thought of as a two-player game between a and e, or as a test or test-case that e poses to a. In either case, the interaction between a and e reveals *some* information about a's underlying objectives, while it is unknown what this information is, or what the underlying objectives are.

Clearly, in order for the interaction function G to be useful in evaluating individuals, it must bear some relation to the underlying objectives. Specifically, consider an individual a with a given set of underlying objectives. Then we will require that any increase in the value of an objective of a, marking an improvement in its skills, must be reflected in an increased outcome of its interaction with some player e. Conversely, G may not provide misleading information by indicating an improvement when there is none, or when the individual's underlying objectives have in fact decreased. This requirement is formally stated as follows:

Definition 2 (Interaction Requirement) *An interaction function G is consistent with underlying objectives U, with $U(x, i)$ written as x_i, if and only if for any pair of learners $a, b \in \mathbb{L}$:*

$$\exists i : a_i > b_i \quad \iff \quad \exists e \in \mathbb{E} : G(a, e) > G(b, e) \tag{3}$$

Since G is typically given, this requirement does not restrict the class of problems for which these results apply, but rather defines which functions may validly be seen as the underlying objectives of a problem. Specifically, we note that intransitivity is not excluded by the requirement, and the proof that will be presented equally applies to problems featuring intransitivity.

In coevolution, each learner is evaluated based on its outcomes against a current set of evaluators. Following Pareto-Coevolution, these outcomes are treated as objectives in the sense employed by Evolutionary Multi-Objective Optimization. A coevolutionary evaluation function will prefer a learner over another if the former learner dominates the latter, see definition 1. This results in the following *coevolutionary evaluation function* F_{coev} for learning individuals:

$$F_{\text{coev}} = \underset{O_G^E}{dom}(a, b) \tag{4}$$

where $a, b \in L$ are learners, and the k^{th} objective of a learner $L^i \in L$ is the outcome of its interaction with the k^{th} evaluator $E^k \in E$:

$$O_G^E(L^i, k) = G(L^i, E^k) \tag{5}$$

The coevolutionary evaluation function F_{coev} might be called a *subjective* evaluation function (Watson & Pollack, 2001), since it depends on the set of evaluators E and therefore does not necessarily reflect all underlying objectives. A general, objective evaluation function which does take all of the underlying objectives into account is given by the ideal evaluation function F_{ideal}. In the following, it will be shown that coevolutionary setups can in principle achieve a subjective evaluation that is equal to the objective evaluation.

3 Ideal Evaluation from Coevolution

Using Ficici's notion of a distinction (2001b), we will say that an evaluator $e \in \mathbb{E}$ *distinguishes* between two learners $a, b \in \mathbb{L}$ if and only if a's outcome against e is higher than b's outcome:

$$dist(e, a, b) \iff G(a, e) > G(b, e) \tag{6}$$

This definition is asymmetric, so that higher and lower objective values can be detected independently. We define a *Complete Evaluation Set* to be a set of evaluators E that make all distinctions that can be made between the learners in a set of learners L:

Definition 3 (Complete Evaluation Set) *An evaluation set $E \subseteq \mathbb{E}$ is* complete *for an interaction function G and a set of learners L if and only if:*

$$\forall a, b \in L : \quad [\exists e \in \mathbb{E} : G(a, e) > G(b, e) \quad \Longrightarrow \quad \exists e' \in E : G(a, e') > G(b, e')] \tag{7}$$

We will write E_L^* to denote such a Complete Evaluation Set for a set of learners L. Regarding the size of the Complete Evaluation Set, we first note that the definition is a condition that can be satisfied by multiple sets of evaluators. The number of distinctions can at most be equal to $n_l^2 - n_l$, where n_l is the number of learners. Since each distinction requires at most one evaluator, the number of evaluators required to construct a Complete Evaluation Set is bounded by this same number.

While this size is already quite reasonable, it may be reduced by selecting evaluators that make multiple distinctions. If desired, the set could be reduced still further by removing evaluators for distinctions that do not affect the dominance relations between learners. For algorithmic purposes however, minimizing the size of the evaluator set beyond the given limit is not necessarily beneficial, as this may decrease the diversity of the evaluator population. The primary observation therefore is that the given bound is sufficiently small to lead to feasible algorithms.

A central theoretical result of this article is that the use of a Complete Evaluation Set E_L^* as objectives for a set of learners L renders the coevolutionary evaluation function equivalent to the ideal evaluation function:

Theorem 1 (Equivalence with the ideal evaluation function) *Let $F_{\text{coev}}(a, b) = \underset{O_G^{E_L^*}}{dom}(a, b)$ be a coevolutionary evaluation function for L based on a Complete Evaluation Set E_L^*. Let $F_{\text{ideal}}(a, b) = \underset{U}{dom}(a, b)$ be the ideal evaluation function for L, based on the underlying objectives U. Furthermore, let G satisfy the interaction requirement for U. Then for any pair of learners $a, b \in L : F_{\text{coev}}(a, b) = F_{\text{ideal}}(a, b)$.*

Interaction outcomes

G(Li,Ek)	E1	E2	E3
L1	0	1	0
L2	0	0	1
L3	1	1	0

Resulting distinctions

dist(Li,Lj)	L1	L2	L3
L1	0	1	0
L2	1	0	1
L3	1	1	0

G(L2,E3)>G(L3,E3)

Figure 1: Matrix representation of the possible distinctions that can be made between a set of learners (example). A distinction between learners L^i and L^j can be made (shown as '1' in the right matrix) if the set of evaluators contains an individual E^k such that the outcome of L_i against E^k exceeds that of L_j.

This theorem is proved in Appendix A. It offers a principled approach to evaluation in coevolution, by guaranteeing ideal evaluation if the evaluator population is a Complete Evaluation Set and the learners use their outcomes against evaluators as objectives. The result thereby also provides a theoretical motivation for Pareto-Coevolution.

3.1 Approximating the Complete Evaluation Set

It was seen that Complete Evaluation Sets result in an evaluation function for learners that is identical to the ideal evaluation function. This leads to the question of whether a Complete Evaluation Set can be approximated by coevolutionary algorithms. It will be seen that this may be feasible in practice.

It was observed that the number of potential distinctions is bounded by the square of the number of learners. Thus, we can treat all potential distinctions between learners as objectives, resulting in a setup where evaluators strive to find all possible distinctions between learners:

$$O(E^k, n_l \cdot i + j) = \begin{cases} 1 & \text{if} \quad G(L^i, E^k) > G(L^j, E^k) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $O(E^k, n)$ is the n^{th} objective of an evaluator $E^k \in E$, L^i is a learner, $n_l = |L|$ is the number of learners, $1 \leq i, j \leq n_l$ and $G(l, e)$ is the interaction function. A convenient representation of the objectives of evaluators is as the entries in a square matrix, where the columns and rows represent the learners, and each non-diagonal entry represents whether a potential distinction between two learners is made by some evaluator, see Figure 1 and eq. 6.

In EMOO, this number of objectives is still large. An important observation however is that whereas generally in EMOO the goal is to combine high objective values in a single individual, here it is sufficient if each distinction is made by *some* evaluator. Thus, given a set of evaluators, we can select a subset that is sufficient for evaluation purposes by choosing one evaluator for each possible distinction.

4 A Class of Algorithms for Pareto-Coevolution

A Complete Evaluation Set for a given set of learners can be approximated by using the distinctions between learners as objectives. To translate this idea into an outline for algorithms, we first combine a population of learners and a set of offspring into

L_{pop}:=random_population()
E_{pop}:=random_population()
while $\neg$ learner-criterion
    $L_{\text{tot}} := L_{\text{pop}} \cup \text{generate}(L_{\text{pop}})$
    while $\neg$ evaluator-criterion
        $E_{\text{tot}} := E_{\text{pop}} \cup \text{generate}(E_{\text{pop}})$
        $\forall i, k : \mathrm{G}[i, k] := G(L^i, E^k)$
        $\forall k, i, j : \mathrm{d}[k, i, j] := (\mathrm{G}[i, k] \ > \ \mathrm{G}[j, k])$
        evaluate(E_{tot},d)
        $E_{\text{pop}} :=$ select (E_{tot})
    end
    evaluate(L_{tot},G)
    $L_{\text{pop}} :=$ select (L_{tot})
end

Figure 2: Algorithm outline for Pareto-Coevolution.

a single set. Next, we identify evaluators that find distinctions among the learners in this combined set. This provides a basis for evaluating the population of learners and its offspring. The population of learners and its offspring typically changes over time, and the identification of an appropriate evaluation set therefore needs to be repeated every generation.[4] To obtain an evaluation set for a given set of learners, we can invoke a secondary evolutionary process for evaluators within the primary cycle for learner evolution. This leads to the outline for coevolutionary algorithms shown in Figure 2.

The *generate* function creates a new generation of individuals given an existing population. The outcomes of the interactions between learners and evaluators are stored in the array G. From this array, the distinctions made by each evaluator are calculated and stored in the array d. *Evaluate* accepts a population of individuals and their objective values and determines which individuals are preferable over others based on their objective values. *Select* removes the non-preferred individuals, thus yielding the next population of individuals.

Specific instances of this class of algorithms can be generated by selecting stop criteria for the two loops. The stop criterion for learner evolution (outer loop) will typically be a domain-dependent performance threshold or a limit on the number of iterations.

Convergence to the ideal evaluation function can be guaranteed by repeating the following cycle. First, produce a set of evaluators chosen from the set of all possible evaluators, where each evaluator has a non-zero probability of being generated. This can be done by choosing randomly from the latter set. Next, store an evaluator for each new distinction found by the evaluators and repeat the cycle. In the limit, this process converges to a set representing all achievable distinctions for a set of n_l learners that contains at most $n_l^2 - n_l$ evaluators.

While this guarantee is of theoretical interest, two practical issues remain. First,

[4] While a generation-based approach is described here, the same principle can be applied in steady-state algorithms.

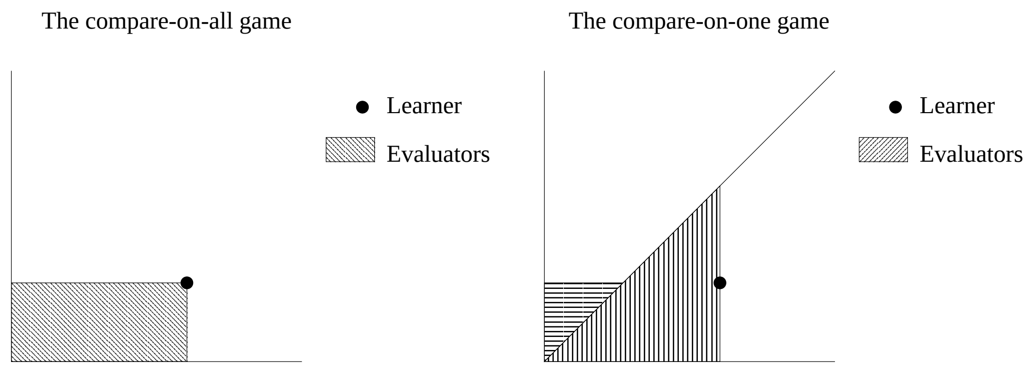

Figure 3: The gray areas show all evaluators that are solved by the learner in the figure. Left: the COMPARE-ON-ALL game. A learner receives a positive outcome if it is equal or greater than the evaluator in every dimension. Right: the COMPARE-ON-ONE game. A learner receives a positive outcome if it is equal or greater in the evaluator's highest dimension.

since it is generally unknown how many of the distinctions are achievable, a stop criterion must be chosen for the evaluator evolution loop. One possibility is to track statistics of the number of distinctions found over time, and to stop when the probability of finding additional evaluators drops below a given threshold. Another possibility, which will be employed here, is to iterate the inner loop for only one step at a time, so as to balance the computational effort spent on evolving learners and evaluators. Thus, we obtain an algorithm where a population of learners and a population of evaluators are both updated once each cycle, as in most two-population coevolutionary setups.

A second issue is search efficiency. The search for new evaluators making additional distinctions may be made much more efficient if it is based on the set of evaluators found so far. Indeed, the motivation for a population-based algorithm is that apart from distinguishing between current learners, evaluators may generate offspring that make *new* distinctions. The algorithm developed in Section 6.2 is motivated by this.

Finally, we consider the number of evaluators required. As seen above, $n_l^2 - n_l$ evaluators suffice to make all possible distinctions between n_l learners. When a population of evaluators is maintained however, we expect that the required number of evaluators will in practice be determined by the number of underlying objectives.

5 Test Problems and Experimental Setup

The previous section has outlined a class of algorithms that can be used for Pareto-Coevolution. With an eye on the practical development of principled algorithms for coevolution, we now investigate to what extent the algorithm is able to overcome problems encountered in coevolution.

It was seen that many of the difficulties with coevolution can be explained by a failure to evaluate on all underlying objectives. Thus, a central question is how evaluation and performance improvement on all underlying objectives can be achieved. To investigate this, test problems are designed that are likely to result in incomplete evaluation.

The problems that will be studied are variants of the Numbers Game (Watson &

Pollack, 2001). Each individual is specified by a number of real valued variables. These variables constitute the genome of the individual. The goal of the problem is to maximize the outcomes of tests posed by evaluators.

The first test-problem is called COMPARE-ON-ALL. In this problem, the learner and the evaluator are compared based on *all* of the evaluator's dimensions. The outcome of the interaction function for this problem is positive (1) if and only if the learner's values are all at least as high as those of its evaluator:

$$\text{compare} - \text{on} - \text{all}: \quad G_{all}(a,e) = \begin{cases} 1 & \text{if} \quad \forall i : a_i \geq e_i \\ -1 & \text{otherwise} \end{cases} \tag{9}$$

where a is a learner, e is an evaluator, and x_i denotes the value of individual x in dimension i.

The underlying objectives of this game simply correspond to its dimensions; this can be checked by verifying that the relation between G_{all} and this choice of underlying objectives satisfies the interaction requirement. Thus, the underlying objectives correspond precisely to the variables of the problem, and the aim is simply to maximize all variables. We may use the same notation x_i for both the variable values and the underlying objective values of an individual x. As we aim to model actual coevolution problems however, the selection mechanism may not use knowledge of these values. Instead, it must be based on the outcomes of interactions between individuals, thus turning the problem into a coevolutionary problem.

The second test-problem is called COMPARE-ON-ONE. In this problem, the learner and the evaluator are compared based on only *one* of the evaluator's dimensions, namely the evaluator's dimension with the highest value. The outcome of the interaction function for this problem is positive (1) if and only if the learner's value in the evaluator's highest dimension is at least as high as that of its evaluator:

$$m = \arg\max_i e_i \tag{10}$$

$$\text{compare} - \text{on} - \text{one}: \quad G_{one}(a,e) = \begin{cases} 1 & \text{if} \quad a_m \geq e_m \\ -1 & \text{otherwise} \end{cases} \tag{11}$$

where a is a learner, e is an evaluator, and x_i denotes the value of individual x in dimension i. Again, it may be checked that the underlying objectives for this problem correspond to its dimensions.

The games are illustrated in Figure 3. For the compare-on-all game (left), a learner solves the tests posed by evaluators that do not exceed its own values. For the compare-on-one game, there are different classes of evaluators, determined by the highest dimensions of the evaluators. Evaluators that have their highest value in the first dimension, i.e. those below the diagonal, can only compare learners based on their horizontal position, while those above the diagonal test learners on their vertical position only.

Figure 4 illustrates the evaluators that can make distinctions between two learners for the two problems. While evaluators in the compare-on-all game can compare learners based on all of their dimensions, this is not possible in the compare-on-one game. Therefore, evaluators in different regions must be maintained.

Although evaluators in distinct regions are required to address the problem, a coevolutionary setup can easily lead to convergence such that evaluators in only some regions are maintained, thus leading to incomplete evaluation. Therefore, the problem is likely to result in a set of evaluators that tests on only some of the underlying objectives. This difficulty makes the problem suitable for the study of methods that aim to prevent incomplete evaluation.

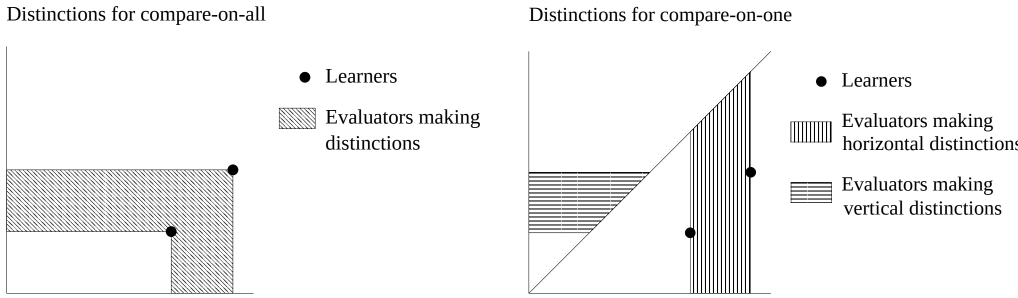

Figure 4: Left: distinctions for the COMPARE-ON-ALL game. Evaluators provide information about all underlying objectives. Right: distinctions for the COMPARE-ON-ONE game. Evaluators in different regions test on different objectives. To make progress on all underlying objectives, evaluators must be maintained in each region.

To detect whether learners progress on only some dimensions, the performance criterion for both test problems is the individual's *minimum* value, rather than for example the average value of its variables. Since individuals start with low values in all dimensions, an increasing performance implies that progress is being made in all dimensions.

We now discuss the experimental setup. Different methods for evaluation and selection will be studied in the following section. A new generation of individuals is created using mutation only. Mutation adds a random value x chosen uniformly from $[-d-b, d-b]$ to a dimension i, where d is called the *mutation distance* and b is the *mutation bias*. The mutation bias reflects the notion that in most problems of practical interest, mutation is more likely to result in deterioration than in improvement. Mutation is applied to two randomly chosen dimensions. Mutating one dimension at a time would be unrealistic, given that we aim to study a model of actual problems; in this case, a higher interaction outcome would precisely indicate that the performance over all objectives has improved. By mutating in two dimensions at a time, an increase in one dimension will often be accompanied by a decrease in another. This renders the problem qualitatively more difficult. Further increases in the number of mutated dimensions would make progress slower, but would not make a qualitative difference. To keep versions of the problem with different dimensionalities comparable, the number is therefore fixed, and unrelated to the total number of dimensions.

6 Design of an Algorithm for Pareto-Coevolution: DELPHI

Our aim is to develop an algorithm for Pareto-Coevolution based on the outline for such algorithms described in Section 4. To this end, specific methods for evaluation and selection have to be chosen. A difficulty is that learner selection and evaluator selection are interdependent. However, since the target for evaluator selection is defined by the Complete Evaluation Set E^*_L, we can study learner selection in isolation by temporarily assuming that evaluator selection identifies E^*_L and thus provides ideal evaluation. By retaining the learner selection method found in this way, we can subsequently study evaluator selection.

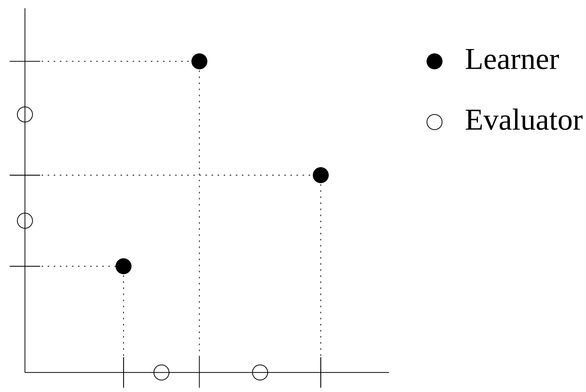

Figure 5: Schematic representation of a Complete Evaluation Set described in the text. Evaluators are selected by projecting the learners onto the axes and placing an evaluator between each pair of subsequent points.

6.1 Learner Selection under Ideal Evaluation

To construct a Complete Evaluation Set, we may choose an evaluator for each learner that has a higher value in some dimension than some other learner, such that the evaluator is in between the learners in this dimension. The evaluator thus makes a distinction between the two learners. To ensure that the evaluator tests on this dimension, all of its other dimensions are given a value of zero, see Figure 5. Any such set satisfies the definition of a Complete Evaluation Set for both test problems, see eq. 7. As stated by the equivalence theorem (1), the Complete Evaluation Set provides the same information as the objectives themselves; whenever two learners have a different value for some objective, an evaluator exists that assigns a higher outcome to the learner with the higher objective value. Thus, evaluation based on E_L^* can simply be implemented by the Pareto-dominance relation using the values of an individual's variables as objectives.

We first use a basic EMOO method for learner selection which evaluates an individual based on whether it is non-dominated, meaning that no currently existing individuals dominate it. Specifically, learners are sorted on the number of individuals by which they are dominated, and selected with a probability that is proportional to the resulting rank. Figure 6 shows the resulting minimum value, averaged over the individuals in each population, for the four-dimensional problem[5] with mutation bias ($b = 0.05$). The graph (bottom line) shows that the method fares poorly even when given access to the ideal evaluation function. The cause of this failure is that the non-dominance criterion, when based on a population rather than the set of all possible individuals, yields only coarse information about the objectives.

Intuitively, it should be possible to ensure progress when given access to the ideal evaluation function. This can indeed be achieved, by using a stricter selection criterion. If individuals are only replaced by individuals that *dominate* it, rather than merely being non-dominated, then replacements can only result in improvement, and regress can be

[5]Since the two test problems share the same set of underlying objectives, they become identical under ideal evaluation.

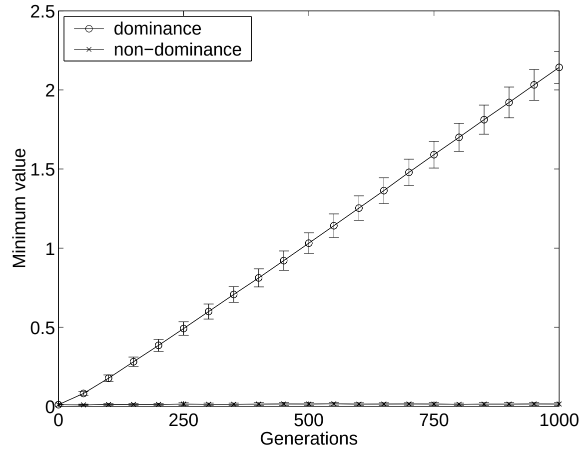

Figure 6: Results for evolving learners under a Complete Evaluation Set E_L^*, which results in evaluation according to the ideal evaluation function. A non-dominance-based selection fails even under these ideal circumstances, and is therefore not expected to function when opponents are evolving. The stricter selection criterion of dominance avoids regress, and is thereby able to improve.

avoided. We therefore use a second selection method that accepts new individuals into the population if and only if they dominate some existing member, which is then replaced. As Figure 6 shows, this method, called dominance selection, indeed achieves progress given a Complete Evaluation Set E_L^* as its objectives.

From the first experiment, we can conclude that non-dominance is insufficiently strict as a criterion for learner replacement. The criterion of dominance guarantees that given access to the ideal evaluation function, any change to the population must result in progress. Accordingly, learner selection will employ the dominance criterion.

6.2 Evaluator Selection

The idea behind the use of a population-based method is that evaluators making useful distinctions are a good basis for finding evaluators that make new distinctions. This idea is most clear when evaluators are assumed to test on a single underlying objective. Experimental evidence supporting this expectation is presented in the next section.

If evaluators test on a single underlying objective, then successful evolution of learners will at some point lead to a state where all learners pass the threshold value posed by an evaluator. Once this happens, the evaluator makes no more distinctions, and a higher threshold is required. If there is any relation between an evaluator and its offspring, the evaluator is a likely genetic source for evaluators representing this higher threshold, and conversely the evaluator's offspring is likely to test on the same objective as its parent. By only replacing an evaluator when it is dominated by its offspring, valuable evaluators can be maintained while making no distinctions, and replaced once a higher threshold for the same objective is found. This replacement criterion is based on the diversity maintenance technique of *Deterministic Crowding* (Mahfoud, 1995). By virtue of this principle, the ability to test on an objective can be preserved even when it is not detectable in the evaluator's objective values. We refer to this method as a *Pareto-hillclimber*, as it performs hill-climbing based on the Pareto-dominance criterion.

Another example of a Pareto-hillclimber is the PAES algorithm (Knowles et al., 2001), which additionally features an archive to estimate dominance rankings.

In summary, we have arrived at a setup where a learner replaces an existing individual if it dominates that individual, while for evaluators we require in addition to this that the replacer must be the offspring of the individual that is being replaced. Thus, given a population of learners L and a population of evaluators E, new learners are evaluated based on the evaluators in E and can replace any learner they dominate, while evaluators are Pareto-hillclimbers that use the distinctions between the learners in L as their objectives. This method will be called DELPHI, which stands for Dominance-based Evaluation of Learners on Pareto-Hillclimbing Individuals.

7 Experimental Results

We examine the DELPHI method on the two test problems that have been described, and first compare its performance to several competitive coevolution methods. Most existing methods for coevolution are forms of competitive coevolution (Hillis, 1990; Rosin & Belew, 1997; Rosin, 1997), although several cooperative alternatives have been suggested (Potter & De Jong, 2000; Moriarty & Miikkulainen, 1998). In competitive coevolution, all individuals try to maximize the outcome of their interactions. A typical approach to dealing with the outcomes of multiple interactions is to calculate an average score. Thus, in the first comparison setup (AVG E, AVG L), learners use the average score against evaluators as a fitness value, while evaluators use the average score against learners as their fitness. Removal selection is done probabilistically by selecting individuals with a probability proportional to their fitness.

A second method (PROB E, PROB L) views the outcomes as objectives, and employs a basic EMOO method (Fonseca & Fleming, 1993). The method performs probabilistic selection based on non-dominance by sorting individuals based on the number of individuals they are dominated by and using the normalized rank as the probability of selection. To provide a stricter selection criterion, we also use a method that selects the best half of the sorted population (HALF E, HALF L). Moving from non-dominance to dominance as a selection criterion, we arrive at a method that replaces an existing individual by any new individual that dominates it (DOM E, DOM L). Finally, we can place the additional condition that the replacee must be the parent of the new individual. For this competitive setup (P-HC E, P-HC L), learners are Pareto-hillclimbers evaluated on the evaluators, and evaluators are Pareto-hillclimbers evaluated on the learners.

The parameters used are as follows. The initial value in each dimension is determined by uniform random selection from $[0, 0.05]$. The mutation distance $d = 0.1$. No mutation bias is used ($b = 0$), except where otherwise indicated. The size of learner and evaluator populations is 50, and so is that of new generations, so that the total learner and evaluator populations before selection both contain 100 individuals. All experiments are averaged over 100 runs, except for trajectory graphs, which display single runs.

Figure 7 shows the performance (average minimum value) of the above methods on the two-dimensional COMPARE-ON-ALL problem. All competitive methods are able to achieve non-trivial progress; the increases in the average minimum values are substantial compared to the mutation distance. For the more difficult COMPARE-ON-ONE game, some progress was still made on the two-dimensional version, but performance on the five-dimensional variant stalls for most methods, see Figure 8. The most difficult problem is obtained by employing a mutation bias ($b = 0.05$) in the five-dimensional COMPARE-ON-ONE problem, so that values from $[-0.15, 0.05]$ are added to two ran-

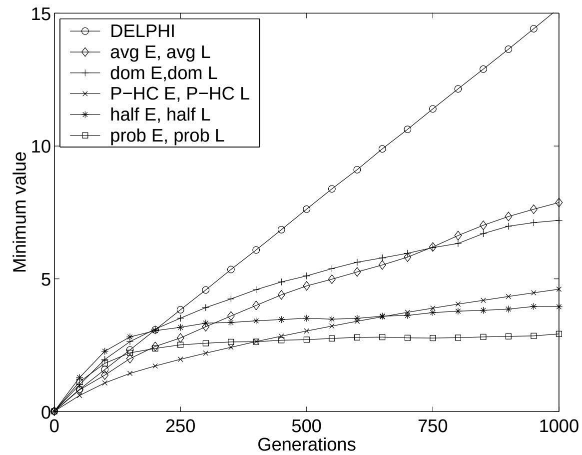

Figure 7: Performance of DELPHI and a number of competitive methods on the 2-dimensional COMPARE-ON-ALL problem. Performance is measured as the average minimum value of an individual's variables. All methods achieve some progress on this problem.

domly selected variables. Using this mutation bias, more than half of the mutations are purely deleterious, and the majority (86%)[6] of the mutations that produce an increase in some dimension cause a (typically larger) decrease in some other dimension. For this difficult problem, no competitive method produces substantial progress, see Figure 9.

In contrast with the competitive methods, DELPHI displays consistent and considerable progress in all underlying dimensions across all problems. The steady increase of the minimum value of learners implies that evaluator selection produces stable evaluation on all underlying objectives. The experiment thereby demonstrates the practical relevance of the theoretical result, which stated that identification of all possible distinctions between learners renders coevolutionary evaluation equivalent to the ideal evaluation function.

To test whether the main choices made in developing DELPHI are necessary, we have furthermore performed the following control experiments. All of these employ the basic setup of using the outcomes of interactions with evaluators as the objectives for learners, and using the distinctions between learners as objectives for evaluators.

First, to see whether Pareto-hillclimbers are useful in maintaining a diverse set of distinctions, we can attempt explicitly to make evaluators spread over the possible distinctions (DOM E, SPREAD DIST L). We do this by assigning to each distinction a fitness contribution of one over the number of evaluators that make the distinction. This method for sharing fitnesses is known as *competitive fitness sharing*. It was introduced by Rosin (1997), and was the most successful of several methods when applied to distinctions (Ficici & Pollack, 2001b), as it is here. Next, we test whether the learners may also benefit from the restriction that new individuals can only replace their parents, so that both learners and evaluators are Pareto-hillclimbers (P-HC E, P-HC DIST L). Finally, we can remove the parent criterion in evaluator selection, so that both learners and evaluators are selected based on dominance (DOM E, DOM DIST L).

[6]Probability of decrease (dec) given an increase (inc): $P(dec|inc) = \frac{p(dec+inc)}{p(inc)} = \frac{p(dec+inc)}{1-p(dec+dec)} = \frac{6}{7}$

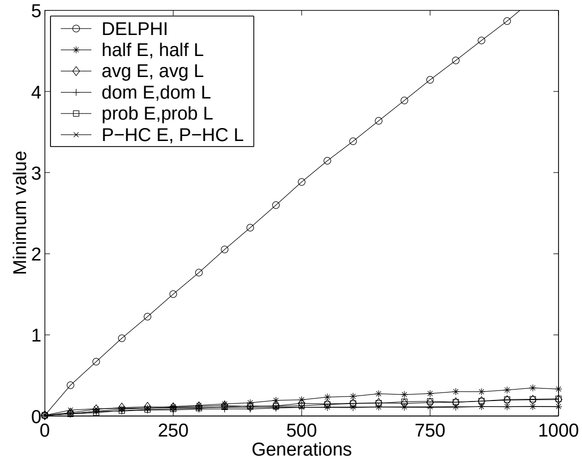

Figure 8: Performance of DELPHI and a number of competitive methods on the 5-dimensional COMPARE-ON-ONE problem. Most competitive methods stall.

While (DOM E, SPREAD DIST L) performs very well for the two-dimensional version of COMPARE-ON-ALL, it is surpassed by DELPHI on the more difficult five-dimensional version, as are the other two methods, see Figure 10. On the COMPARE-ON-ONE problem, only the two methods employing Pareto-hillclimbers display sustained progress (Figure 11). Both methods still result in sustained progress when mutation bias is added (Figure 12), while the remaining two methods fail on this final problem.

We conclude that the method that has been developed (DELPHI) provides sustained progress on all versions of the problem, and surpasses all of the other methods that have been investigated. The results show that using Pareto-hillclimbers on distinctions as evaluators can result in an evaluation function that continues to test on all underlying objectives. As a result, learners can achieve sustained progress in all underlying objectives. Disengagement and over-specialization are thereby successfully avoided.

The variant that uses Pareto-hillclimbers on distinctions for evaluators and on outcomes for learners (P-HC E, P-HC DIST L) also displays sustained progress for all problems, but at a lower rate. While the current experiments do not demonstrate this, we believe that this variant may be of relevance when the evaluation provided by evaluators is unstable, for instance in problems with a large number of underlying objectives. For such problems, Pareto-hillclimbers may be able to retain valuable characteristics that are not discerned by the current evaluator population. The lower rate of progress may be explained from the reduced number of opportunities for new individuals to replace existing ones.

Finally, we investigate the behavior of the method by plotting the trajectories of learners and evaluators in the COMPARE-ON-ONE problem. Since only mutation is used, each individual has a single chain of ancestors that leads back to the initial population. We plot this chain for all individuals present in the final population, after running the method for a hundred generations. Figure 13 shows the resulting trajectories. To avoid influencing the trajectories, we do not cut off variable values at zero as in

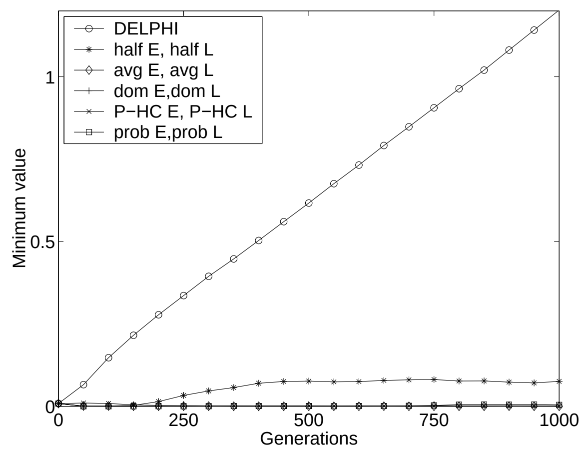

Figure 9: Performance of DELPHI and a number of competitive methods on the five-dimensional COMPARE-ON-ONE problem with mutation bias. None of the competitive methods provides sustained progress, DELPHI does.

the normal experiments, and use no mutation bias.

The most striking observation is that the evaluators follow the axes, while learners progress in all dimensions simultaneously. This provides support for the earlier suggestion that each underlying objective is identified by evaluators that focus on a single objective.

As a comparison, we plotted the same information for a competitive experiment (AVG E, AVG L), see Figure 14. While learners have made some initial progress in the vertical dimension, they have over-specialized by progressing mainly in one dimension, the horizontal one. Since no evaluators making vertical distinctions persist and all current evaluators are far away from the diagonal, it is unlikely that the potential for making vertical progress will be regained.

The trajectories in Figure 13 provide some experimental evidence for the suggestion that evaluators can potentially identify the underlying objectives. However, part of the observed behavior may be a result of the direct correspondence between variables and objectives. We therefore test whether evaluators still identify the underlying objectives when these do not correspond directly to properties of the problem other than the outcomes of interactions. To this end, we rotate the underlying objectives. Both learners and evaluators are rotated clockwise 30 degrees with respect to the origin before determining the outcomes of interactions. As a result, the underlying objectives of the problem are rotated 30 degrees anti-clockwise; by following these rotated directions, the evaluators will end up at the optimal directions of progress when evaluated. The variables of the problem and the operators of variation remain unchanged. We run the experiment longer (a thousand generations) in order to determine whether the directions in which the evaluators move are stable. As Figure 15 shows, the evaluators approximately identify the new underlying objectives of the problem. This indicates that the discovery of the underlying objectives by the algorithm was not dependent on a correspondence between the variables and objectives of the problem.

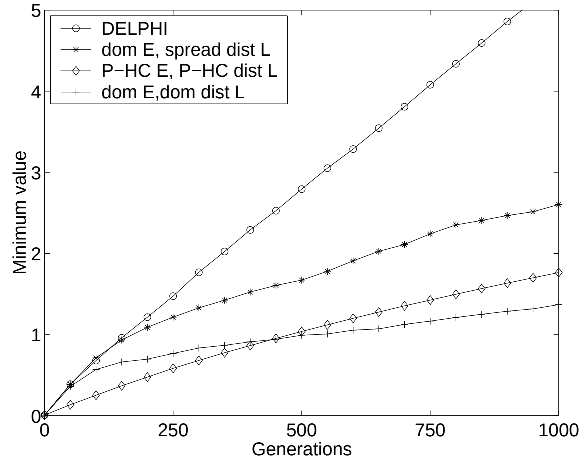

Figure 10: Performance of DELPHI and other distinction-based methods on the 5-dimensional COMPARE-ON-ALL problem. All methods achieve substantial progress.

8 Discussion

To assess the applicability of the approach that has been described, we examine to what extent the assumptions that have been made hold in problems of practical interest. A first question is whether the interaction function corresponds to the intended underlying objectives. We have identified three issues that could be thought to threaten this: the outcomes of interactions may be influenced by structural influences or by noise, and interactions may be non-transitive. Secondly, we expect that the most important issues with respect to practical application are general search issues that will be detailed. These four topics will be discussed in turn below.

8.1 Structural Influences

Since interactions are modeled as functions, we implicitly assume deterministic interactions. In stochastic problems, different encounters between the same pair of individuals can lead to different outcomes. If the factors governing the outcomes are structural, e.g. whether a checkers player plays first or not, then the different circumstances under which the opponent is faced should be viewed as distinct interactions. In the example, the test posed by playing first against said checkers player should be viewed as distinct from the test posed by letting the player go first, and these tests would therefore be distinct evaluators.

8.2 Noise

Apart from structural influences, interactions may be affected by noise, resulting from dice rolls for example. Such influences may be addressed by sampling over a number of interactions, and using the average outcome as an indication of the relation between the skills of the individuals. Given enough samples, any non-structural influences can be averaged out. As a result, the averaged outcome of the interactions between one pair of individuals will come to bear a consistent relation (lower, equal, or higher) to that of any other pair, resulting in equivalence classes of averaged outcomes. When viewed in terms of these equivalence classes, rather than of the particular averaged outcome, the

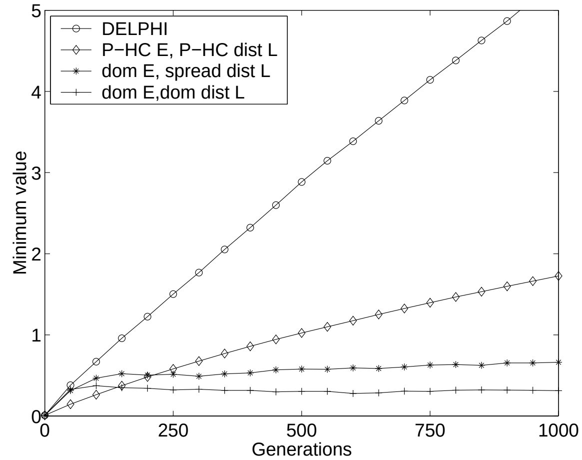

Figure 11: Performance of DELPHI and other distinction-based methods on the 5-dimensional COMPARE-ON-ONE problem. The methods employing Pareto-hillclimbers achieve sustained progress, while the other two methods stall.

resulting evaluation is deterministic. Such equivalence classes are obtained when the ranges of average outcomes of individuals become non-overlapping or identical. Since the outcomes must in the limit converge to the expected value, such a sample size is guaranteed to exist, though it may be very large. The downside of this approach is the computational cost, which must be weighed against the benefits of accurate evaluation.

8.3 Intransitive Superiority

A third known problem is that of *intransitive superiority* relationships (Cliff & Miller, 1995; Watson & Pollack, 2001). This refers to problems where an individual $b > a$, $c > b$, and $a > c$, such as the Rock, Paper, Scissors game. As in other problems, a non-minimal set of underlying objectives can be obtained by considering the outcomes against all possible evaluators as objectives. Thus, intransitive problems can be viewed as multi-objective problems, where the ability to address each opponent is a separate objective. In the example, a would have a high value for the objective represented by c, and a low value for the objective represented by b. The method will strive to find individuals that maximize or trade off these objectives.

Individuals in an intransitive cycle such as Rock, Paper, Scissors are mutually non-dominated. If no other individuals exist, the solution to the resulting EMOO problem contains all individuals in the cycle. Interestingly, such cycles can also become irrelevant. When other individuals are identified that dominate all of the individuals in the cycle, these are preferable, and the cycle need no longer be maintained.

A question to be verified is whether the constraint on the interaction function G (eq. (3)) holds in the presence of intransitivity. This is the case, since by choosing the evaluators themselves as objectives, a distinction between two learners always corresponds to a higher value for some objective, and vice versa. We therefore see no obstructions in using the presented approach for problems featuring intransitive superiority.

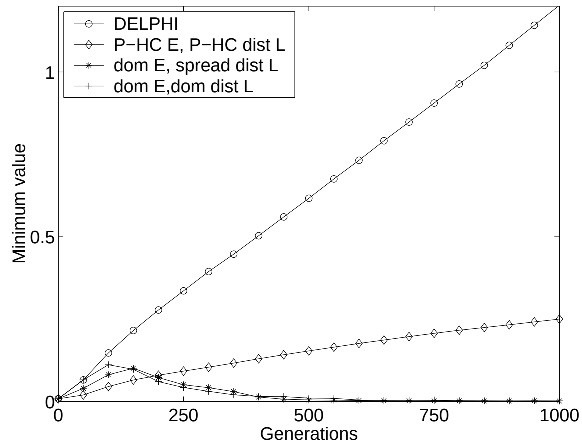

Figure 12: Performance of DELPHI and other distinction-based methods on the 5-dimensional COMPARE-ON-ONE problem with mutation bias. The methods employing Pareto-hillclimbers still achieve sustained progress, while the other two methods now fail entirely on this most difficult problem in the set.

8.4 Search and Exploration

The selection criterion for both learners and evaluators is strict, as it is based on dominance. However, it may not always be possible to find dominating individuals within a single step of mutation (or other variation operators) from the current population. We believe this limited degree of exploration is the most important limitation of the algorithm that has been presented, and that it will have to be addressed before the principles demonstrated here can be applied in practice.

We expect that accurate evaluation and exploration can be achieved by combining an archive that is updated using strict replacement criteria with a population in which exploration is allowed. The archive achieves a form of *elitism* for multi-objective optimization, while the population provides exploration. For the explorative population, a number of methods for diversity maintenance in evolutionary multi-objective optimization in general are available (Zitzler & Thiele, 1999; Knowles & Corne, 1999; Corne, Knowles, & Oates, 2000; Deb, Agrawal, Pratab, & Meyarivan, 2000).

The particular algorithm that has been investigated, DELPHI, replaces existing learners only once they are dominated. For the class of test problems that have been investigated, this works effectively, as higher performance for a learner objective is always possible. For general multi-objective problems however, trade-offs between the different objectives typically exist, and this criterion may be too strict. Suitable methods for this situation have recently been developed within Evolutionary Multi-Objective Optimization (Laumanns, Thiele, Deb, & Zitzler, 2002), and may readily be applied within the current framework.

The class of algorithms that has been defined requires evaluators to find distinctions among learners as part of the evolutionary cycle. Since this is not required in conventional coevolutionary algorithms, one question is how difficult it is to find such distinctions. The feasibility of this can only become clear through practical applications, but promising results with the use of distinctions for the evaluation of evaluators

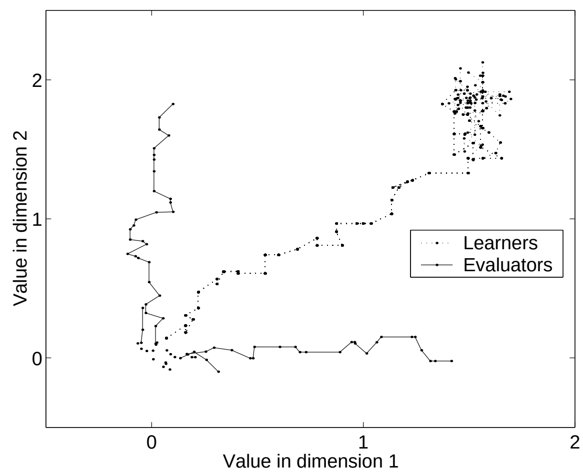

Figure 13: Trajectories of the individuals in the final population of a run of DELPHI on the two-dimensional COMPARE-ON-ONE problem. Each of the underlying objectives is identified by some evaluator.

have already been obtained (Ficici & Pollack, 2001b).

For symmetric problems, one way to find an evaluator distinguishing between learners a and b may be to take a and apply a small mutation to it. This will result in an individual that typically performs less well than a and that, if the mutation is appropriate, still performs better than b. Thus, evaluators may be identified based on the learners, rather than by an independent search process.

A question is whether evaluators can be found for all underlying objectives. First, we note that this question is relevant to any approach, and that mechanisms to promote more complete evaluation can only be helpful. If evaluators for certain objectives are not present in the initial population, the algorithm that has been described contains no specific mechanisms to search for such missing objectives, although it will identify them when found. For many problems, a sufficiently large population will display at least some variance in the performance achieved by learners and tested by evaluators. If this is not the case, e.g. because objectives only start to play a role for high-quality individuals, the addition of an explicit exploration mechanism may be necessary.

Evaluation is based on Pareto-dominance, and thus discards any scaling information in interaction outcomes. Since multiple objectives can provide more information than a single value and since fitness ranking methods are used with success even in scalar fitness genetic algorithms, we do not expect this difference to be of importance, but it is possible that scaling information may be useful for certain problems.

The solution concept we have employed is the Pareto-front. For certain problems, it may not be feasible to maintain a representative subset of this front. Even then, a multi-objective approach may still be useful as a means to circumvent local optima. While single-objective methods search in one direction, EMOO methods explore several directions simultaneously. This may help to identify a promising direction.

Finally, a note on computational complexity. DELPHI employs the outcomes of interactions between all learners and all evaluators. For large populations, this may be costly. To improve efficiency, multiple explorative pairs of learner and evalua-

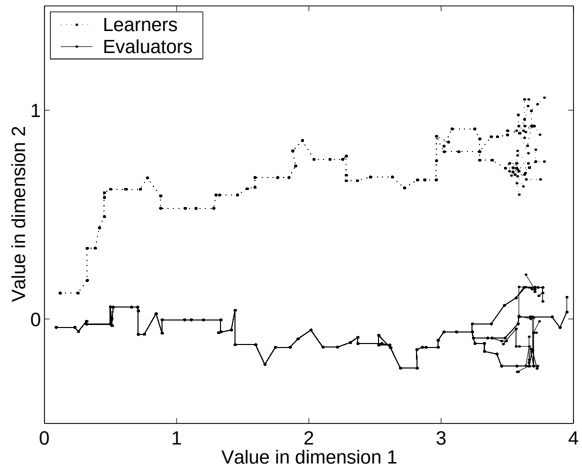

Figure 14: Trajectories for the competitive method on the same problem. Only one of the underlying objectives is identified. No progress is made in the vertical dimension as result.

tor subpopulations can be used in combination with a learner and evaluator archive that preserves overall improvements and provides complete evaluation. Such multi-population setups also have a clear potential for parallel implementations.

8.5 Application

One question is what problems may benefit most from the coevolutionary approach that has been described. A minimal condition is the availability of tests, i.e. ways to extract information from learners. It has been shown that a set of evaluators that is squared in the number of learners is sufficient for ideal evaluation. In practice, the number of evaluators may depend more on the number of underlying objectives; if each evaluator tests on one objective, then existing evaluators can be used to generate new evaluators for the same objective. If this is the case, the approach will be suited most for problems with a limited number of objectives. We expect coevolution to be most valuable in problems where *multiple* objectives underlie performance, while the nature of these objectives is unknown.

9 Conclusions

In the past, co-evolution has been set up as a process where the unary fitness of evolving individuals is judged not with an absolute yardstick but by a sum or average of values which are calculated relative to a current population. Next to a limited number of successes at getting co-evolution to proceed, researchers have found a variety of self-limiting behaviors such as disengagement, the Red Queen effect, and mediocre stability.

Following a line of research aimed at accurate automatic evaluation, which arose out of work by Juillé, Rosin, and Ficici and takes into account the possibility that *multiple* objectives may underlie a problem, we have now discovered that it is possible to determine a *complete* evaluation set. This theoretical set of evaluations is the perfect teacher which can precisely determine whether one individual dominates another. It

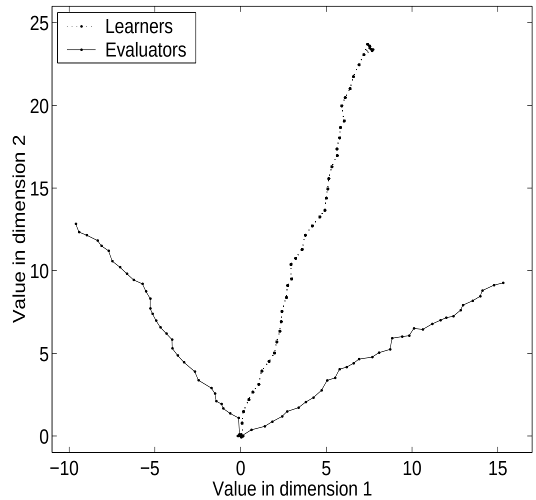

Figure 15: Trajectories in a version of the COMPARE-ON-ONE problem where the underlying objectives have been rotated 30 degrees. The evaluators still identify the underlying objectives when they do not correspond to the variables of the problem.

allows an evolutionary system to tease apart those individuals which should be maintained and those which can be safely discarded.

Moreover, we found that for a given population this set is of manageable size, namely quadratic in the number of individuals. By virtue of this, the set can not only be identified in the theoretical limit, but also be approximated in practice. Using our algorithm, DELPHI, we demonstrate that with appropriate heuristics, a dynamic set of evaluators can be maintained which drive a population forward without reference to a fixed external fitness function, which is the goal of coevolutionary learning.

The Complete Evaluation Set evaluates according to the *underlying objectives* of a problem. In experiments, the evaluators were observed to identify the underlying objectives. For most problems of interest, the question of what the underlying objectives are has not yet been asked. An intriguing consequence of Pareto-Coevolution therefore is that it may help uncover the hidden structure of problems.

Acknowledgements

The authors would like to thank Bram Bakker, Marcus Hutter, Christian Igel, Jürgen Schmidhuber, Kenneth Stanley, Pieter Jan ’t Hoen, Marc Toussaint, Steven van Dijk, and the anonymous reviewers for helpful comments and suggestions, and the members of the DEMO-lab, in particular Anthony Bucci, Sevan Ficici and Richard Watson, for this and for discussions that were vital in the development of this work. EdJ gratefully acknowledges a TALENT fellowship from the Netherlands Organization for Scientific Research (NWO).

Online Appendix: MATLAB® implementation of DELPHI

Detailed pseudo-code, movies of two experiments, and a MATLAB® implementation of the DELPHI algorithm are available online from the homepage of the first author.[7]

Symbols

Objectives and dominance:

O	a set of objectives
$O(x,i)$	value of the i^{th} objective in O for individual x
x_i	i^{th} underlying objective of individual x
U	set of underlying objectives $U(x,i) = x_i$
O_G^E	objectives given by the outcomes against E using interaction function G
$\underset{O}{dom}(a,b)$	a Pareto-dominates b with respect to objectives O

Populations and individuals:

	Small letters, e.g. x, a, b, denote individuals.
$\mathbb{E}$	set of all possible evaluators
E	a particular set of evaluators
E^i	i^{th} individual in the set of evaluators E
E_L^*	Complete Evaluation Set for a set of learners L
$\mathbb{L}$	set of all possible learners
L	a particular set of learners
L^i	i^{th} individual in the set of learners L

Function:

$F(a,b)$:	evaluation function, specifying whether a is to be preferred over b
$F_{\text{coev}}(a,b)$:	coevolutionary evaluation function, resulting from interactions of a and b with evaluators
$F_{\text{ideal}}(a,b)$:	ideal evaluation function
$G(a,b)$:	interaction function, yields the outcome of an interaction between a learner a and an evaluator b

[7]URL: `http://www.cs.uu.nl/~dejong`

Appendix A

Proof 1 (Equivalence with the ideal evaluation function) *To prove theorem 1, we show that given the assumption (eq.* (3)*) on the interaction function G, the evaluation function F_{coev} that results from using E_L^* as objectives for learners equals the ideal evaluation F_{ideal}.*

$$F_{\text{coev}}(a,b) \iff F_{\text{ideal}}(a,b) \tag{12}$$

$$\underset{O_G^{E_L^*}}{dom}(a,b) \iff \underset{U}{dom}(a,b) \text{ (by (4) and (2))} \tag{13}$$

$$[\forall e \in E_L^* : G(a,e) \geq G(b,e) \land \exists e \in E_L^* : G(a,e) > G(b,e)] \tag{14}$$

$$\iff [\forall i : a_i \geq b_i \land \exists i : a_i > b_i] \text{ (by (5) and (1))} \tag{15}$$

First we show the implication:

$$\text{Assume: } \forall e \in E_L^* : G(a,e) \geq G(b,e) \land \exists e \in E_L^* : G(a,e) > G(b,e) \tag{16}$$

$$\text{Assume: } \exists i : b_i > a_i \tag{17}$$

$$\Rightarrow \exists e \in \mathbb{E} : G(b,e) > G(a,e) \text{ (by (3))} \tag{18}$$

$$\Rightarrow \exists e \in E_L^* : G(b,e) > G(a,e) \text{ (by (7))} \tag{19}$$

$$\text{This contradicts (16). Therefore (17) cannot hold, so:} \tag{20}$$

$$\nexists i : b_i > a_i \tag{21}$$

$$\Rightarrow \forall i : a_i \geq b_i \tag{22}$$

$$\text{Furthermore: } \exists i : a_i > b_i \text{ (by (16, right) and (3))} \tag{23}$$

Combining (22) *and* (23) *proves the implication. To show the reverse implication:*

$$\text{Assume: } \forall i : a_i \geq b_i \land \exists i : a_i > b_i \tag{24}$$

$$\text{Assume: } \exists e \in \mathbb{E} : G(b,e) > G(a,e) \tag{25}$$

$$\exists i : b_i > a_i \text{ (by (3))} \tag{26}$$

$$\text{This contradicts (24). Therefore (25) cannot hold, so:} \tag{27}$$

$$\nexists e \in \mathbb{E} : G(b,e) > G(a,e) \tag{28}$$

$$\Rightarrow \forall e \in \mathbb{E} : G(a,e) \geq G(b,e) \tag{29}$$

$$\text{And since } E_L^* \text{ is a subset of } \mathbb{E}\text{:} \tag{30}$$

$$\Rightarrow \forall e \in E_L^* : G(a,e) \geq G(b,e) \tag{31}$$

$$\exists e \in \mathbb{E} : G(a,e) > G(b,e) \text{ (by (24, right) and (3))} \tag{32}$$

$$\exists e \in E_L^* : G(a,e) > G(b,e) \text{ (by (32) and (7))} \tag{33}$$

Combining (31) *and* (33) *proves the reverse implication, and completes the proof.* ■

References

Ahluwalia, M., & Bull, L. (2001). Coevolving functions in genetic programming. *Journal of Systems Architecture*, *47*(7), 573–585.

Angeline, P. J., & Pollack, J. B. (1993). Competitive environments evolve better solutions for complex tasks. In S. Forrest (Ed.), *Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93* (pp. 264–270). San Francisco, CA: Morgan Kaufmann.

Angeline, P. J., & Pollack, J. B. (1994). Coevolving high-level representations. In C. G. Langton (Ed.), *Artificial life iii* (Vol. XVII, pp. 55–71). Redwood City, CA: Addison-Wesley.

Axelrod, R. (1987). The evolution of strategies in the iterated prisoner's dilemma. In L. Davis (Ed.), *Genetic algorithms and simulated annealing* (pp. 32–41). London: Pitman Publishing.

Barricelli, N. A. (1962). Numerical testing of evolution theories. Part I: Theoretical introduction and basic tests. *Acta Biotheoretica*, *16*(1–2), 69–98.

Barricelli, N. A. (1963). Numerical testing of evolution theories. Part II: Preliminary tests of performance, symbiogenesis and terrestrial life. *Acta Biotheoretica*, *16*(3–4), 99–126.

Bucci, A., & Pollack, J. B. (2002). Order-theoretic analysis of coevolution problems: Coevolutionary statics. In *Proceedings of the GECCO-02 workshop on coevolution: Understanding coevolution.*

Bucci, A., & Pollack, J. B. (2003a). Focusing versus intransitivity. Geometrical aspects of coevolution. In E. Cantú-Paz et al. (Ed.), *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-03.* Berlin: Springer.

Bucci, A., & Pollack, J. B. (2003b). A mathematical framework for the study of coevolution. In *Foundations of genetic algorithms (FOGA-2002).* San Francisco, CA: Morgan Kaufmann.

Bull, L. (2001). On coevolutionary genetic algorithms. *Soft Computing*, *5*(3), 201–207.

Cliff, D., & Miller, G. F. (1995). Tracking the Red Queen: Measurements of adaptive progress in co-evolutionary simulations. In F. Morán, A. Moreno, J. J. Merelo, & P. Chacón (Eds.), *Proceedings of the Third European Conference on Artificial Life: Advances in Artificial Life* (Vol. 929, pp. 200–218). Berlin: Springer.

Cliff, D., & Miller, G. F. (1996). Co-evolution of pursuit and evasion II: Simulation methods and results. In P. Maes, M. J. Mataric, J.-A. Meyer, J. B. Pollack, & S. W. Wilson (Eds.), *From Animals to Animats. proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, SAB-96* (Vol. 4, pp. 506–515). Cambridge, MA: The MIT Press.

Coello, C. A. C. (2000). An Updated Survey of GA-Based Multiobjective Optimization Techniques. *ACM Computing Surveys*, *32*(2), 109–143.

Corne, D. W., Knowles, J. D., & Oates, M. J. (2000). The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature, PPSN-VI* (Vol. 1917, pp. 839–848). Berlin: Springer.

De Jong, E. D. (2003). Representation development from Pareto-coevolution. In E. Cantú-Paz et al. (Ed.), *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-03* (p. 262-273). Berlin: Springer.

De Jong, E. D., & Pollack, J. B. (2002). *Principled Evaluation in Coevolution* (Tech. Rep. No. CS-02-225; May 31 2002). Brandeis University.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms.* New York, NY: Wiley & Sons.

Deb, K., Agrawal, S., Pratab, A., & Meyarivan, T. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature, PPSN-VI* (Vol. 1917, pp. 849–858). Berlin: Springer.

Epstein, S. L. (1994). Toward an ideal trainer. *Machine Learning*, *15*(3), 251–277.

Ficici, S. G., Melnik, O., & Pollack, J. B. (2000). A game-theoretic investigation of selection methods used in evolutionary algorithms. In *Proceedings of the 2000 congress on evolutionary computation, CEC-00.* Piscataway, NJ: IEEE Press.

Ficici, S. G., & Pollack, J. B. (1998). Coevolving communicative behavior in a linear pursuer-evader game. In R. Pfeifer, B. Blumberg, J.-A. Meyer, & S. Wilson (Eds.), *From Animals to Animats. Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior, SAB-98* (pp. 263–269). Cambridge, MA: The MIT Press.

Ficici, S. G., & Pollack, J. B. (2000). A game-theoretic approach to the simple coevolutionary algorithm. In M. Schoenauer et al. (Ed.), *Parallel Problem Solving from Nature, PPSN-VI* (Vol. 1917). Berlin: Springer.

Ficici, S. G., & Pollack, J. B. (2001a). Game theory and the simple coevolutionary algorithm: Some preliminary results on fitness sharing. In R. K. Belew & H. Juillé (Eds.), *Proceedings of the GECCO-01 Workshop on Coevolution: Turning Adaptive Algorithms upon Themselves.*

Ficici, S. G., & Pollack, J. B. (2001b). Pareto optimality in coevolutionary learning. In J. Kelemen (Ed.), *Sixth European Conference on Artificial Life.* Berlin: Springer.

Floreano, D., Nolfi, S., & Mondada, F. (1998). Competitive coevolutionary robotics: From theory to practice. In R. Pfeifer, B. Blumberg, J.-A. Meyer, & S. Wilson (Eds.), *From Animals to Animats. Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior, SAB-98* (pp. 512–524). Cambridge, MA: The MIT Press.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms, ICGA-93* (pp. 416–423). San Francisco, CA: Morgan Kaufmann.

Fonseca, C. M., & Fleming, P. J. (1995). An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, *3*(1), 1–16.

Funes, P. (2001). *Evolution of complexity in real-world domains.* Unpublished doctoral dissertation, Brandeis University, Waltham, MA.

Gomez, F. J., & Miikkulainen, R. (1999). Solving non-markovian control tasks with neuro-evolution. In T. Dean (Ed.), *Proceedings of the 16th International Joint Conference on Artificial Intelligence, IJCAI-99* (pp. 1356–1361). San Francisco, CA: Morgan Kaufmann.

Hillis, D. W. (1990). Co-evolving parasites improve simulated evolution in an optimization procedure. *Physica D*, *42*, 228–234.

Husbands, P. (1994). Distributed Coevolutionary Genetic Algorithms for Multi-Criteria and Multi-Constraint Optimisation. In T. Fogarty (Ed.), *Evolutionary Computing* (Vol. 865, pp. 150–165). Berlin: Springer.

Husbands, P., & Mill, F. (1991). Simulated co-evolution as the mechanism for emergent planning and scheduling. In R. Belew & L. Booker (Eds.), *Proceedings of the 4th international conference on genetic algorithms* (pp. 264–270). San Francisco, CA: Morgan Kaufmann.

Juillé, H. (1999). *Methods for Statistical Inference: Extending the Evolutionary Computation Paradigm.* Unpublished doctoral dissertation, Brandeis University.

Juillé, H., & Pollack, J. B. (1996). Co-evolving intertwined spirals. In L. Fogel, P. Angeline, & T. Baeck (Eds.), *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming* (pp. 461–467). Cambridge, MA: The MIT Press.

Juillé, H., & Pollack, J. B. (1998). Coevolutionary learning: a case study. In *Proceedings of the 15th International Conference on Machine Learning* (pp. 251–259). San Francisco, CA: Morgan Kaufmann.

Kauffman, S. A., & Johnsen, S. (1992a). Co-evolution to the edge of chaos: Coupled fitness landscapes, poised states, and co-evolutionary avalanches. In C. Langton, C. Taylor, J. Farmer, & S. Rasmussen (Eds.), *Artificial life II* (Vol. X, pp. 325–369). Redwood City, CA: Addison-Wesley.

Kauffman, S. A., & Johnsen, S. (1992b). Co-evolution to the edge of chaos: Coupled fitness landscapes, poised states, and co-evolutionary avalanches. In C. Langton, C. Taylor, J. Farmer, & S. Rasmussen (Eds.), *Artificial Life II* (pp. 325–369). Redwood City, CA: Addison-Wesley.

Knowles, J., & Corne, D. (1999). The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. In P. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, & A. Zalzala (Eds.), *Proceedings of the Congress on Evolutionary Computation, CEC-99* (Vol. 1, pp. 98–105). Piscataway, NJ: IEEE Press.

Knowles, J. D., Watson, R. A., & Corne, D. W. (2001). Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In E. Zitzler, K. Deb, L. Thiele, C. C. Coello, & D. Corne (Eds.), *First international conference on evolutionary multi-criterion optimization* (Vol. 1993, pp. 268–282). Berlin: Springer.

Koza, J. R. (1991). Evolution and co-evolution of computer programs to control independent-acting agents. In J.-A. Meyer & S. W. Wilson (Eds.), *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior, SAB-91* (pp. 366–375). Cambridge, MA: The MIT Press.

Koza, J. R. (1992). Genetic evolution and co-evolution of computer programs. In C. Langton, C. Taylor, J. Farmer, & S. Rasmussen (Eds.), *Artificial life II* (Vol. X, pp. 603–629). Redwood City, CA: Addison-Wesley.

Laumanns, M., Thiele, L., Deb, K., & Zitzler, E. (2002). Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, *10*(3), 263–282.

Lindgren, K. (1992). Evolutionary phenomena in simple dynamics. In C. Langton, C. Taylor, J. Farmer, & S. Rasmussen (Eds.), *Artificial life II* (Vol. X, pp. 295–312). Redwood City, CA: Addison-Wesley.

Lubberts, A., & Miikkulainen, R. (2001). Co-evolving a go-playing neural network. In R. K. Belew & H. Juillé (Eds.), *Proceedings of the GECCO-01 Workshop on Coevolution: Turning Adaptive Algorithms upon Themselves* (pp. 14–19).

Mahfoud, S. W. (1995). *Niching Methods for Genetic Algorithms.* Unpublished doctoral dissertation, University of Illinois at Urbana-Champaign. (IlliGAL Report 95001)

Maynard Smith, J. (1982). *Evolution and the theory of games.* Cambridge, UK: Cambridge University Press.

Miller, G., & Cliff, D. (1994). Protean behavior in dynamic games: Arguments for the co-evolution of pursuit-evasion tactics. In D. Cliff, P. Husbands, J.-A. Meyer, & S. W. Wilson (Eds.), *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior, SAB-94* (pp. 411–420). Cambridge, MA: The MIT Press.

Miller, J. (1989). The coevolution of automata in the repeated prisoner's dilemma. *Santa Fe Institute working paper 89-003*. (Later appeared as (Miller, 1996))

Miller, J. (1996). The coevolution of automata in the repeated prisoner's dilemma. *Journal of Economic Behavior and Organization*, *29*(1), 87–112.

Moriarty, D. E., & Miikkulainen, R. (1998). Forming neural networks through efficient and adaptive coevolution. *Evolutionary Computation*, *5*(4), 373–399.

Noble, J., & Watson, R. A. (2001). Pareto coevolution: Using performance against coevolved opponents in a game as dimensions for Pareto selection. In L. Spector, E. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, & E. Burke (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference* (p. 493-500). San Francisco, CA: Morgan Kaufmann.

Nolfi, S., & Floreano, D. (1998). Co-evolving predator and prey robots: Do 'arms races' arise in artificial evolution? *Artificial Life*, *4*(4).

Olsson, B. (2001). Co-evolutionary search in asymmetric spaces. *Information Sciences*, *133*(3-4), 103–125.

Østergaard, E. H., & Lund, H. H. (2003). Co-evolving complex robot behavior. In A. M. Tyrrell, P. C. Haddow, & J. Torresen (Eds.), *Evolvable systems: From biology to hardware, fifth international conference, ICES-2003* (Vol. 2606, pp. 308–319). Springer-Verlag.

Pagie, L., & Hogeweg, P. (1998). Evolutionary consequences of coevolving targets. *Evolutionary Computation*, *5*(4), 401–418.

Pagie, L., & Hogeweg, P. (2000). Information integration and Red Queen dynamics in coevolutionary optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation, CEC-00* (pp. 1260–1267). Piscataway, NJ: IEEE Press.

Paredis, J. (1996). Coevolutionary computation. *Artificial Life*, *2*(4).

Paredis, J. (1997). Coevolving cellular automata: Be aware of the Red Queen. In T. Bäck (Ed.), *Proceedings of the Seventh International Conference on Genetic Algorithms, ICGA-97.* San Francisco, CA: Morgan Kaufmann.

Pollack, J. B., & Blair, A. D. (1998). Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, *32*(1), 225–240.

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

Reynolds, C. W. (1994). Competition, coevolution and the game of tag. In R. A. Brooks & P. Maes (Eds.), *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems* (pp. 59–69). Cambridge, MA: The MIT Press.

Rosca, J. P., & Ballard, D. H. (1996). Discovery of subroutines in genetic programming. In P. Angeline & K. E. Kinnear, Jr. (Eds.), *Advances in Genetic Programming 2* (pp. 177–202). Cambridge, MA: The MIT Press.

Rosin, C. D. (1997). *Coevolutionary Search among Adversaries.* Unpublished doctoral dissertation, University of California, San Diego, CA.

Rosin, C. D., & Belew, R. K. (1997). New methods for competitive coevolution. *Evolutionary Computation*, *5*(1), 1–29.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *3*(2), 210–229. (Reprinted in E. A. Feigenbaum and J. Feldman (Eds.) 1963, *Computers and Thought*, McGraw-Hill, New York)

Schmidhuber, J. (1999). Artificial curiosity based on discovering novel algorithmic predictability through coevolution. In P. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, & A. Zalzala (Eds.), *Proceedings of the Congress on Evolutionary Computation, CEC-99* (Vol. 3, pp. 1612–1618). Piscataway, NJ: IEEE Press.

Sims, K. (1994). Evolving 3D morphology and behavior by competition. In R. Brooks & P. Maes (Eds.), *Artificial Life IV* (pp. 28–39). Cambridge, MA: The MIT Press.

Srinivas, N., & Deb, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, *2*(3), 221–248.

Stanley, K. O., & Miikkulainen, R. (2002). Continual coevolution through complexification. In W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, & N. Jonoska (Eds.), *Proceedings of the genetic and evolutionary computation conference, GECCO-02:* (pp. 113–120). San Francisco, CA: Morgan Kaufmann.

Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations.* Unpublished doctoral dissertation, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Von Neumann, J. (1928). Zur Theorie der Gesellshaftsspiele. *Mathematical Annals*, *100*, 295–320.

Watson, R. A., & Pollack, J. B. (2000). Symbiotic combination as an alternative to sexual recombination in genetic algorithms. In M. Schoenauer et al. (Ed.), *Parallel Problem Solving from Nature, PPSN-VI* (Vol. 1917). Berlin: Springer.

Watson, R. A., & Pollack, J. B. (2001). Coevolutionary dynamics in a minimal substrate. In L. Spector, E. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, & E. Burke (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-01* (p. 702-709). San Francisco, CA: Morgan Kaufmann.

Watson, R. A., & Pollack, J. B. (2003). A computational model of symbiotic composition in evolutionary transitions. *Biosystems*, *69*(2-3), 187–209. (Special Issue on Evolvability, ed. Nehaniv)

Werfel, J., Mitchell, M., & Crutchfield, J. P. (2000). Resource sharing and coevolution in evolving cellular automata [Letter]. *IEEE Transactions on Evolutionary Computation*, *4*(4), 388.

Werner, G. M., & Dyer, M. G. (1991). Evolution of communication in artificial organisms. In C. Langton, C. Taylor, J. Farmer, & S. Rasmussen (Eds.), *Artificial Life II* (Vol. X). Redwood City, CA: Addison-Wesley.

Zitzler, E., & Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, *3*(4), 257–271.