Learning the Ideal Evaluation Function

Edwin D. de Jong* and Jordan B. Pollack

DEMO Lab, Volen National Center for Complex Systems, Brandeis University
MS018, 415 South street, Waltham MA 02454-9110, USA
{edwin|pollack}@cs.brandeis.edu, http://demo.cs.brandeis.edu

Abstract. Designing an adequate fitness function requires substantial knowledge of a problem and of features that indicate progress towards a solution. Coevolution takes the human out of the loop by dynamically constructing the evaluation function based on interactions between evolving individuals. A question is to what extent such automatic evaluation can be adequate. We define the notion of an ideal evaluation function. It is shown that coevolution can in principle achieve ideal evaluation. Moreover, progress towards ideal evaluation can be measured. This observation leads to an algorithm for coevolution. The algorithm makes stable progress on several challenging abstract test problems.

Keywords: Coevolution, Pareto-Coevolution, Complete Evaluation Set, ideal evaluation, underlying objectives, Pareto-hillclimber, over-specialization

Designing an adequate fitness function requires substantial domain knowledge and can be a critical factor in evolution, see e.g. [9]. Often though, tests revealing information about the qualities of individuals can readily be performed. In chess for example, absolute evaluation of strategies is extremely difficult, while comparing individuals only requires knowledge of the rules of the game. If individuals can be evaluated based on tests, coevolution can be used to circumvent the problem of defining a fitness function.

Coevolution has already produced a number of promising results [10,19,12, 17]. However, there are various ways in which evaluation in coevolution can become inaccurate [21,2,16]. As a step towards accurate evaluation, Juillé defines a domain-specific *ideal trainer* [11]. Rosin provides an *automatic* mechanism for accurate evaluation, but the approach is based on a single-objective perspective, and likely to stall for problems with multiple underlying objectives. Pareto-coevolution [6,20] uses the outcomes of a learner against coevolving evaluators (tests) as objectives in the sense of Evolutionary Multi-Objective Optimization.

By combining Rosin's complete set of tests with Ficici's important notion of *distinctions* [7], we arrive at the concept of a *Complete Evaluation Set*. The complete evaluation set was first described in [3], and detects all differences between learners relevant to selection.

* Current address: DSS Group, Utrecht University. dejong@cs.uu.nl

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2723, pp. 274–285, 2003.
© Springer-Verlag Berlin Heidelberg 2003

We prove that given a complete evaluation set as evaluators, Pareto-coevolution leads to *ideal evaluation*, i.e. evaluation according to all underlying objectives of a problem. Using order theory, Bucci has defined a set of *maximally informative evaluators* [1]. While this set also makes all distinctions necessary for learner selection, it is different, as the complete evaluation set is a *maximally informative set* of evaluators. By virtue of this, the complete evaluation set has the property that its required size is bounded and small.

The complete evaluation set provides a practical way for coevolution methods to approximate ideal evaluation. An algorithm based on this principle is described, and found to achieve stable progress on a number of test problems that could not be addressed by standard coevolution methods used for comparison. This paper summarizes the results described in our technical report [3]. A more extensive account of this work is to appear in [4].

1 Evaluation in Coevolution

We consider problems where multiple objectives may underly performance. This includes as a special case single fitness value problems. The theoretical *ideal evaluation function* specifies which individuals would be preferred over which other individuals if the underlying objectives would be available. We demonstrate that using the outcomes of interactions between coevolving individuals as objectives, it is possible to construct an evaluation function that is precisely equivalent to the ideal evaluation function.

1.1 An Ideal Evaluation Function

The problem of evaluating individuals according to multiple objectives is studied in Evolutionary Multi-Objective Optimization (EMOO), see e.g. [8,5]. We follow EMOO in using the Pareto-dominance relation to compare individuals:

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.

To obtain an evaluation function F_{ideal} that determines for any pair of individuals a and b whether a is to be preferred over b, we can directly employ the Pareto-dominance relation based on the (unknown) underlying objectives U:

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

$$U(x,i) = x_i \tag{3}$$

In general, the solution to a multi-objective problem is a tradeoff front of individuals that achieve the different objectives to different degrees. If a single optimum exists, as in problems with scalar fitness functions, this individual is also the solution of the corresponding EMOO problem.

1.2 Coevolution: Interactions as a Basis for Evaluation

The difficulty of evaluation in coevolution is that selection does not have access to the ideal evaluation function. Instead, selection decisions must be based on the outcomes of interactions between individuals. We will demonstrate that these interactions can provide sufficient information for ideal evaluation.

We distinguish between *learners*, and *evaluators*. Learners are to address the problem at hand. The aim of the evaluators is to distinguish between learners. The set of all possible learners is denoted as $\mathbb{L}$, and the set of all possible evaluators as $\mathbb{E}$. Particular sets of learners and evaluators are denoted as L and E.

All interactions are assumed to be pairwise. An interaction is a function $G : \mathbb{L}\text{x}\mathbb{E} \to \mathbb{O}$ that accepts a learner and an evaluator. It returns an outcome for the learner from some ordered set of values $\mathbb{O}$, e.g. real numbers or game outcomes. 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. 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 that determine the quality of individuals. Specifically, we require that any increase in an underlying objective of an individual a must be reflected in an increased outcome of its interaction with some player b. Conversely, the information contained in G should not provide misleading information by indicating an improvement when there is none.

Formally, the *interaction requirement* specifies that for any pair of learners $a, b \in \mathbb{L}$:

$$\exists i : a_i > b_i \quad \iff \quad \exists e \in \mathbb{E} : G(a, e) > G(b, e) \tag{4}$$

Each learner is evaluated based on its outcomes against the current set of evaluators. Following Pareto-coevolution [6,20], these outcomes are treated as objectives. This results in the following evaluation function F_{coev} for learners:

$$F_{\text{coev}} = \underset{O_G^E}{dom}(a, b) \tag{5}$$

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 G with the k^{th} evaluator $E^k \in E$:

$$O_G^E(L^i, k) = G(L^i, E^k) \tag{6}$$

2 Principled Evaluation in Coevolution

An evaluator $e \in \mathbb{E}$ *distinguishes* between two learners $a, b \in \mathbb{L}$ if a's outcome against e is higher than b's outcome:

$$dist(e, a, b) \iff G(a, e) > G(b, e) \tag{7}$$

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 L:

Definition 2 (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:*

$$\forall a, b \in L : \; [\exists e \in \mathbb{E} : G(a,e) > G(b,e) \implies \exists e' \in E : G(a,e') > G(b,e')] \quad (8)$$

We will write E_L^* to denote an evaluation set that satisfies this property for a set of learners L. The theoretical result of this paper 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 $a, b \in L : F_{\text{coev}}(a,b) = F_{\text{ideal}}(a,b)$.*

A proof is given in appendix A. The finding implies that by treating the outcomes of learners against evaluators as objectives, ideal evaluation can in principle be achieved. Thus, it may be seen as a motivation for Pareto-Coevolution.

2.1 Approximating the Complete Evaluation Set

We now consider how algorithms may approximate the complete evaluation set. This is surprisingly tractable, since the number of potential distinctions is 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:

$$O(E^k, n_l \cdot i + j) = \begin{cases} 1 & \text{if} \quad G(L^i, E^k) > G(L^j, E^k) \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

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 and $G(l,e)$ is the interaction function accepting a learner and an evaluator. 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 entry represents a distinction between two learners, see figure 1 and eq. 7.

3 An Algorithm for Pareto-Coevolution

The above idea can be translated into an outline for algorithms by combining a current population of learners and a set of offspring into a single set of learners. To obtain an evaluation set for this set of learners, we invoke a secondary evolutionary process. This leads to an outline for algorithms, see figure 2.

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)

Fig. 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 (1) if an evaluator E^k exists such that the outcome of L_i against E^k exceeds that of L_j.

Convergence to ideal evaluation can be guaranteed in the limit by generating every possible evaluator with non-zero probability, and collecting any evaluator making a new distinction; for n learners, this leads to a set of at most n^2 evaluators. In practice, we iterate the inner loop for a single step only, so as to balance the computational effort spent on evolving learners and evaluators.

Concerning learner selection, preliminary experiments led to the finding that non-dominance is not strict enough as a selection criterion for learners and can result in regress. Therefore, a learner may replace an existing individual only if it dominates that individual. This simple technique is sufficient when a global optimum exists. For an algorithm also striving towards a balanced distribution of individuals over the tradeoff front, see [14].

```
L_pop:=random_population()
E_pop:=random_population()
while ¬ performance-criterion
    L_tot := L_pop ∪ generate(L_pop)
    while ¬ distinctions-criterion
        E_tot := E_pop ∪ generate(E_pop)
        ∀i,k : G[i,k] := G(L^i, E^k)
        ∀k,i,j : d[k,i,j] := (G[i,k] >G[j,k])
        evaluate(E_tot,d)
        E_pop := select (E_tot)
    end
    evaluate(L_tot,G)
    L_pop := select (L_tot)
end
```

Fig. 2. Outline for coevolution algorithms that approximate the ideal evaluation function.

The strict selection consideration also applies to evaluator selection. In addition, diverse evaluators must be maintained, representing all underlying objectives. Therefore, an evaluator will be replaced by its offspring only, and only if this offspring dominates it. This is similar to the *deterministic crowding* method for diversity maintenance, see [15]. We call such individuals *Pareto-hillclimbers*; the PAES algorithm [13] is another example of a Pareto-hillclimber.

We have arrived at a setup where, 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.

4 Test Problems and Experimental Setup

We will now investigate the algorithm derived from the ideal evaluation principle in experiments. The test problems employed are variants of the Numbers Game [21]. Individuals are vectors of real valued variables. The underlying objectives for the problems correspond precisely to these variables. Hence, the aim should be to maximize each of the individual's variables. However, as we aim to study coevolution, the selection mechanism may not use knowledge of the underlying objectives, but is based on the outcomes of interactions between individuals. The difficulty of the task is determined among other factors by the information the interaction function G provides about the underlying objectives of an individual.

The purpose of the test problems is to test to what extent coevolution algorithms are able to provide accurate evaluation, i.e. evaluation according to all underlying objectives. To this end, the problems should make accurate evaluation difficult. This is achieved by making it likely for evaluators to represent only a subset of the dimensions or objectives in the problem. When this occurs, learners can only progress on a subset of the underlying objectives, a phenomenon called *over-specialization* or *focusing* [21]. In this case the minimum value of learners will not increase further. By using the minimum value of individuals as a performance measure, we can detect whether progress is being made on all underlying objectives.

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:

$$G_{\text{all}}(a, e) = \begin{cases} 1 & \text{if} \quad \forall i : a_i \geq e_i \\ -1 & \text{otherwise} \end{cases} \tag{10}$$

where a is a learner, e is an evaluator, and x_i denotes the value of individual x in dimension i. In the COMPARE-ON-ONE problem, the learner and the evaluator are compared based on only *one* of the evaluator's dimensions, namely the

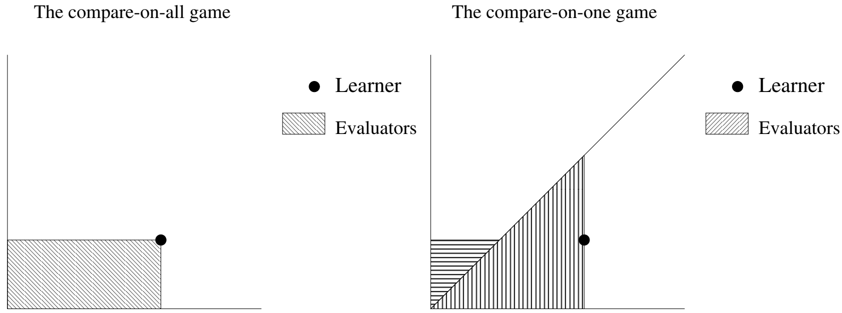

Fig. 3. The grey areas show all evaluators that are solved by the learner in the figure. Left: the COMPARE-ON-ALL game. A learner receives a positive outcome if it is equal or greater than the evaluator in every dimension. Right: the COMPARE-ON-ONE game. A learner receives a positive outcome if it is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

evaluator's dimension with the highest value. The games are illustrated in figure 3.

$$m = \arg\max_i e_i \tag{11}$$

$$G_{one}(a, e) = \begin{cases} 1 & \text{if} \quad a_m \geq e_m \\ -1 & \text{otherwise} \end{cases} \tag{12}$$

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 of the space must be maintained. This results in a strong risk of maintaining evaluators for only some of the underlying objectives, as desired.

5 Experimental Results

The setup is as follows. Initial values in each dimension are chosen uniformly from $[0, 0.05]$. A new generation of individuals is created using mutation. Mutation adds a value chosen uniformly from $[-d-b, d-b]$ to a dimension i, where $d = 0.1$ is the *mutation distance* and $b = 0.05$ (where used) is the *mutation bias*. Mutation is applied to two randomly chosen dimensions. Thus, an increase in one dimension will often be accompanied by a decrease in another, and an improved interaction outcome does not imply improvement on all objectives. The size of learner and evaluator populations and of new generations is 50, resulting in learner and evaluator sets of size 100. All experiments (except the trajectory graph) are averaged over 100 runs.

We performed experiments with the COMPARE-ON-ALL and COMPARE-ON-ONE game in 2-dimensional and 5-dimensional form, with and without mutation bias. Due to space limits, we present results for the easiest and most difficult variants in the problem set: 2-dimensional COMPARE-ON-ALL without mutation

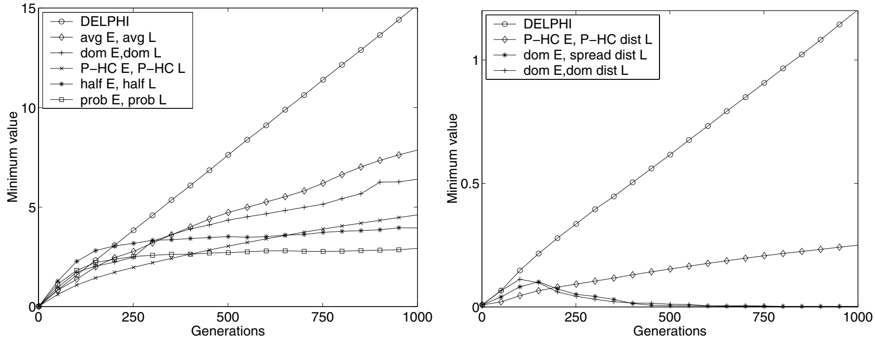

Fig. 4. Left: Performance of DELPHI and a number of competitive methods on the 2-dimensional COMPARE-ON-ALL problem. All methods achieve some progress on this problem. **Right:** DELPHI and comparison methods on 5-dimensional COMPARE-ON-ONE with mutation bias. Only the methods employing Pareto-Hillclimbing still achieve sustained progress; the other methods overspecialize, and neglect one or more objectives.

bias and 5-dimensional COMPARE-ON-ONE with mutation bias. For the latter problem, 86% of the mutations that produce an increase in some dimension cause a (typically larger) decrease in some other dimension.

We first compare DELPHI to several competitive coevolution methods. In AVG E, AVG L, the fitness of learners is the average score against evaluators, vice versa. Individuals are selected into the next population with a fitness-proportional probability. PROB E, PROB L views the outcomes as objectives, and employs a standard EMOO method [8] sorting individuals based on the number of individuals they are dominated by and using the normalized rank as the probability of selection. A stricter variant HALF E, HALF L selects the best half of the population. Still more strict is a method replacing an existing individual by any new individual that dominates it (DOM E, DOM L). Finally, we require that the replacer must be the offspring of the replacee(P-HC E, P-HC L), so that both learners and evaluators are Pareto-hillclimbers.

Figure 4 shows the average minimum value for the two-dimensional COMPARE-ON-ALL problem. All competitive methods are able to achieve some progress. DELPHI outperforms all of these, and makes remarkably constant progress.

To test whether choices made in developing DELPHI are necessary, we perform several control experiments. This time, the much more difficult COMPARE-ON-ONE problem is used with five dimensions and with mutation bias. All methods use the outcomes of interactions with evaluators as the objectives for learners, and use the distinctions between learners as objectives for evaluators. DOM E, SPREAD DIST L attempts to make evaluators spread over the possible distinctions. The fitness contribution for making a distinction is shared with other evaluators making the distinction. This *competitive fitness sharing* [18] method was the most successful of several methods used in [7] when applied to distinc-

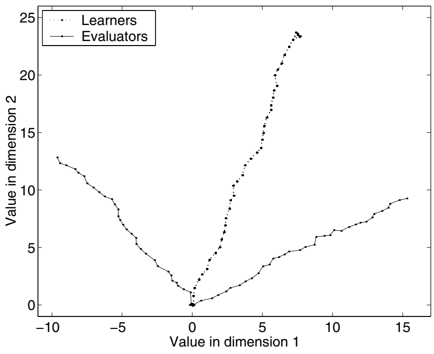

Fig. 5. Trajectories in version of the COMPARE-ON-ONE problem where the underlying objectives have been rotated 30 degrees anti-clockwise. The evaluators still identify the underlying dimensions when these do not correspond to the variables of the problem.

tions, as it is here. P-HC E, P-HC L tests whether learners may also benefit from the parent criterion; both learners and evaluators are Pareto-hillclimbers. To test if the parent criterion is necessary in evaluator selection, DOM E, DOM DIST L uses dominance for both learner and evaluator selection.

For this difficult test problem, only methods employing Pareto-Hillclimbing for the evaluation of evaluators achieve sustained progress on all objectives, see fig. 4. The comparison methods are unable to do so, and even deteriorate due to overspecialization, i.e. values are not maintained or improved for all objectives simultaneously. In summary, only DELPHI displays consistent and considerable progress across all test problems.

Finally, we investigate whether evaluators identify the underlying objectives when these have no direct correspondence to the variables of the problem. To test this, individuals in COMPARE-ON-ONE are projected onto a rotated coordinate system. The variables and operators of variation remain unchanged. As the trajectories in figure 5 show, the evaluators approximately identify the new underlying objectives of the problem, while learners progress evenly in both of the extracted underlying dimensions. Thus, the identification of the underlying objectives was not merely due to a correspondence between the variables and objectives of the problem.

6 Conclusions

Coevolution in principle offers a potential for learning in problems where no adequate evaluation function is known. We began by considering what the ideal evaluation function would be if one would have access to the underlying objectives of a problem. Since these underlying objectives are not available, actual evaluation in coevolution must be based on interactions between individuals. The theoretical result of the article is that in the limit of finding all possible distinctions, this evaluation becomes equal to the ideal evaluation function.

The result immediately suggests a practical operational criterion for *approximating* the ideal evaluation function in the form of Ficici's distinctions [7]. We have developed an algorithm based on this principle called DELPHI. The algorithm evaluates learners by using coevolving *evaluators* as objectives, while these evaluators are evaluated by using their ability to make distinctions between learners as objectives. Strict criteria for learner and evaluator selection are found to be instrumental in DELPHI's ability to achieve sustained progress.

DELPHI was found to substantially outperform comparison methods on several abstract test problems of varying difficulty. Experimental evidence was presented indicating that the evaluators identify the underlying objectives of the problem. While the current article has explored one particular algorithm, the idea of approximating the ideal evaluation function can be taken up in many different ways, and provides a principled approach to evaluation in coevolution. We therefore hope that this work may stimulate the development of new, reliable algorithms for coevolution.

Acknowledgements. The authors would like to thank Kenneth Stanley 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).

References

1. Anthony Bucci and Jordan B. Pollack. Order-theoretic analysis of coevolution problems: Coevolutionary statics. In *Proceedings of the GECCO-2002 Workshop on Coevolution: Understanding Coevolution*, 2002.
2. D. Cliff and G. F. Miller. Tracking the Red Queen: Measurements of adaptive progress in co-evolutionary simulations. In F. Morán, A. Moreno, J. J. Merelo, and P. Chacón, editors, *Proceedings of the Third European Conference on Artificial Life: Advances in Artificial Life*, volume 929 of *LNAI*, pages 200–218, Berlin, 1995. Springer.
3. Edwin D. De Jong and Jordan B. Pollack. Principled Evaluation in Coevolution. Technical Report CS-02-225, Brandeis University, May 31, 2002.
4. Edwin D. De Jong and Jordan B. Pollack. Ideal evaluation from coevolution. *Evolutionary Computation*, accepted for publication.
5. Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* Wiley & Sons, New York, NY, 2001.
6. Sevan G. Ficici and Jordan B. Pollack. A game-theoretic approach to the simple coevolutionary algorithm. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Julian Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature, PPSN-VI*, volume 1917 of *LNCS*, Berlin, 2000. Springer.
7. Sevan G. Ficici and Jordan B. Pollack. Pareto optimality in coevolutionary learning. In Jozef Kelemen, editor, *Sixth European Conference on Artificial Life*, Berlin, 2001. Springer.

8. Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms, ICGA-93*, pages 416–423, San Francisco, CA, 1993. Morgan Kaufmann.
9. Pablo Funes. *Evolution of Complexity in Real-World Domains.* PhD thesis, Brandeis University, Waltham, MA, 2001.
10. D. W. Hillis. Co-evolving parasites improve simulated evolution in an optimization procedure. *Physica D*, 42:228–234, 1990.
11. Hugues Juille. *Methods for Statistical Inference: Extending the Evolutionary Computation Paradigm.* PhD thesis, Brandeis University, 1999.
12. Hugues Juille and Jordan B. Pollack. Co-evolving intertwined spirals. In L.J. Fogel, P.J. Angeline, and T. Baeck, editors, *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 461–467, Cambridge, MA, 1996. The MIT Press.
13. Joshua D. Knowles, Richard A. Watson, and David W. Corne. Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In E. Zitzler, K. Deb, L. Thiele, C.A. Coello Coello, and D. Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, volume 1993 of *LNCS*, pages 268–282. Springer-Verlag, 2001.
14. Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, 10(3):263–282, 2002.
15. Samir W. Mahfoud. *Niching Methods for Genetic Algorithms.* PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, May 1995. IlliGAL Report 95001.
16. Ludo Pagie and Paulien Hogeweg. Information integration and Red Queen dynamics in coevolutionary optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation, CEC-00*, pages 1260–1267, Piscataway, NJ, 2000. IEEE Press.
17. Jordan B. Pollack and Alan D. Blair. Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, 32(1):225–240, 1998.
18. Christopher D. Rosin. *Coevolutionary Search among Adversaries.* PhD thesis, University of California, San Diego, CA, 1997.
19. Karl Sims. Evolving 3D morphology and behavior by competition. In R. Brooks and P. Maes, editors, *Artificial Life IV*, pages 28–39, Cambridge, MA, 1994. The MIT Press.
20. Richard A. Watson and Jordan B. Pollack. Symbiotic combination as an alternative to sexual recombination in genetic algorithms. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Julian Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature, PPSN-VI*, volume 1917 of *LNCS*, Berlin, 2000. Springer.
21. Richard A. Watson and Jordan B. Pollack. Coevolutionary dynamics in a minimal substrate. In L. Spector, E. Goodman, A. Wu, W.B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2001*, pages 702–709, San Francisco, CA, 2001. Morgan Kaufmann.

Appendix A: Proof of the Equivalence

Proof (Equivalence with the ideal evaluation function). To prove the equivalence theorem, we show that given the interaction requirement for G, the coevolutionary evaluation function F_{coev} equals the ideal evaluation function F_{ideal}:

$$
\begin{aligned}
&F_{\text{coev}}(a,b) \iff F_{\text{ideal}}(a,b) && (13)\\
&\underset{O_G^{E_L^*}}{dom}(a,b) \iff \underset{O}{dom}(a,b) \text{ (by (5) and (2))} && (14)\\
&\quad [\forall e \in E_L^* : G(a,e) \geq G(b,e) \wedge \exists e \in E_L^* : G(a,e) > G(b,e)] && (15)\\
&\quad \iff [\forall i : a_i \geq b_i \wedge \exists i : a_i > b_i] \text{ (by (6) and (3))} && (16)\\
&\text{Assume: } \forall e \in E_L^* : G(a,e) \geq G(b,e) \wedge \exists e \in E_L^* : G(a,e) > G(b,e) && (17)\\
&\quad \text{Assume: } \exists i : b_i > a_i && (18)\\
&\Rightarrow \exists e \in \mathbb{E} : G(b,e) > G(a,e) \text{ (by (4))} && (19)\\
&\Rightarrow \exists e \in E_L^* : G(b,e) > G(a,e) \text{ (by (8))} && (20)\\
&\quad \text{This contradicts (17). Therefore (18) cannot hold, so:} && (21)\\
&\quad \nexists i : b_i > a_i && (22)\\
&\Rightarrow \forall i : a_i \geq b_i && (23)\\
&\quad \text{Furthermore: } \exists i : a_i > b_i \text{ (by (17, right) and (4))} && (24)
\end{aligned}
$$

Combining (23) and (24) proves the implication. To show the reverse implication:

$$
\begin{aligned}
&\text{Assume: } \forall i : a_i \geq b_i \wedge \exists i : a_i > b_i && (25)\\
&\quad \text{Assume: } \exists e \in \mathbb{E} : G(b,e) > G(a,e) && (26)\\
&\quad \exists i : b_i > a_i \text{ (by (4))} && (27)\\
&\quad \text{This contradicts (25). Therefore (26) cannot hold, so:} && (28)\\
&\quad \nexists e \in \mathbb{E} : G(b,e) > G(a,e) && (29)\\
&\Rightarrow \forall e \in \mathbb{E} : G(a,e) \geq G(b,e) && (30)\\
&\quad \text{And since } E_L^* \text{ is a subset of } \mathbb{E}\text{:} && (31)\\
&\Rightarrow \forall e \in E_L^* : G(a,e) \geq G(b,e) && (32)\\
&\quad \exists e \in \mathbb{E} : G(a,e) > G(b,e) \text{ (by (25, right) and (4))} && (33)\\
&\quad \exists e \in E_L^* : G(a,e) > G(b,e) \text{ (by (33) and (8))} && (34)
\end{aligned}
$$

Combining (32) and (34) proves the reverse implication, and completes the proof. ■