A Mathematical Framework for the Study of Coevolution

Anthony Bucci
DEMO Lab
Computer Science Department, MS018
Brandeis University
Waltham, MA 02454
abucci@cs.brandeis.edu

Jordan B. Pollack
DEMO Lab
Computer Science Department, MS018
Brandeis University
Waltham, MA 02454
pollack@cs.brandeis.edu

Abstract

Despite achieving compelling results in engineering and optimization problems, coevolutionary algorithms remain difficult to understand, with most knowledge to date coming from practical successes and failures, not from theoretical understanding. Thus, explaining why coevolution succeeds is still more art than science. In this paper, we present a theoretical framework for studying coevolution based on the mathematics of ordered sets. We use this framework to describe solutions for coevolutionary optimization problems, generalizing the notion of Pareto non-dominated front from the field of multi-objective optimization. Our framework focuses attention on the order structure of solution and test sets, which we argue is a key source of difficulty in coevolutionary optimization problems. As an application of the framework we show, in the special case of two-player games, that Pareto dominance is closely related to intransitivities in the game.

1 Introduction

Coevolutionary algorithms progress from a simple intuition: evolve the fitness function together with the evolving individuals. By adjusting the challenge put to evolving individuals, we hope algorithms might tune the fitness function to push individuals into continually increasing their capabilities. There have been a number of compelling successes in the field over the past decade, in domains such as cellular automata [6, 9], game playing [10], sorting networks [8] and robotics [12], suggesting coevolution holds great potential to produce further useful and interesting results. Karl Sims' work in particular offers the hope that, besides being successful at engineering problems, coevolution might also support open-ended dynamics in which evolving entities increase in capability and complexity indefinitely, as species in nature evidently do.

Unfortunately, the price paid for varying the fitness function during search is that coevolutionary dynamics can be complex and difficult to understand. Several well-known, though imperfectly-understood, issues threaten any application of a coevolutionary algorithm. The Red Queen Effect [3] can obscure progress in coevolving systems, rendering it difficult or impossible to tell if the algorithm has accomplished something useful. Mediocre stable states [5] arise when collusion permits suboptimal individuals to appear better than they are. Intransitive superiority cycles, and the related problem of overspecialization [14], can cause coevolutionary dynamics to cycle through a set of suboptimal individuals without making progress. These issues are apparently related. Indeed, it is unclear how one can even reasonably discuss notions like "goodness" or "progress" in a coevolutionary setting. Understanding these issues in a common framework is the motivation behind our work. We will see, particularly in section 3, that intransitivities are closely related to the underlying payoff structure of a problem. While this statement is intuitively clear, our framework makes this intuition precise, exposing methods for approaching the issue of intransitivity systematically.

One noteworthy theoretical approach which has already made strides in this direction is Ficici et al.'s Simple Coevolutionary Algorithm [4]. The Simple Coevolutionary Algorithm is an extension of Vose's Simple Genetic Algorithm [13] to model the dynamics of coevolution . An advance latent in this work and later articulated in [6] is *Pareto coevolution*. Borrowing ideas from multi-objective optimization [7], Pareto coevolution treats each possible individual as an objective against which other evolving individuals are optimizing themselves. The relevant measure of individuals in this case is then *Pareto dominance*: an individual is dominated if there is some other individual which does at least as well as it does against all others, and better against at least one. Pareto dominance offers us a notion of "goodness" and "progress" in a coevolutionary domain.

In this paper, we formalize the ideas behind Pareto coevolution, building a mathematical framework in which to study coevolution as an optimization procedure. We consider a class of coevolutionary optimization problems represented with a function of the form $p : S \times T \to R$, where R is a preordered set [11]. As a mnemonic device, we think of the function p as payoff, S as the set of candidate solutions, T as tests, and R as results (outcomes). p is encoding all possible interactions between candidate solutions and tests. The results preorder R is encoding which outcomes are better than others.

Such a function resembles the payoff matrices found in game theory.[1] However, in contrast to payoff matrices, the function p indexes values by what might be distinct and infinite sets; furthermore, the entries of the matrix come from an arbitrary ordered set R instead of a particular one like $\mathbb{R}$. We will call these problems *coevolutionary optimization problems*. Part of our aim is to explore what might be optimized in such problems.

To this end, in section 2 we will define a notion of solution for coevolutionary optimization problems of this form, generalizing common solution concepts used in genetic algorithm function optimization (GA) and multi-objective optimization (MOO). We will also introduce a dual notion, the set of maximally-informative tests, which tells us something about the structure and difficulty of the problem. As an application of the framework, in section 3 we consider the special case of two-player, two-outcome games, showing the Pareto dominance relation offers new information exactly

[1] Indeed, game-theoretic payoff matrices provide ready examples of such functions.

when the game is intransitive. In section 4 we return to issues of coevolutionary dynamics, discussing how they might be understood in terms of our framework.

We assume the reader is familiar with discrete mathematics. We give relevant background material and establish notational conventions in appendix A. We will freely use the notions of pullback order (definition A.9) and currying a function (definition A.12).

2 The Framework

In this section we develop a theoretical ordering of individuals in a coevolutionary optimization problem. The order allows us to rank individuals in such a way that we can express a solution as the set of *maximal candidates*. When applied to the special case of function optimization, the set of maximal candidates is the set of maxima of the objective function. When applied to multi-objective optimization, the set of maximal candidates is exactly the Pareto front.

Dually, we can order tests by informativeness, and define the set of *maximally-informative tests*. A key result, expressed in theorem 2.7, is that the maximally-informative tests induce the same set of maximal candidates as the full set of tests. Thus, we see the same information about ranking candidate solutions using just the maximally-informative tests, a set which might be much smaller than the full set T. Reducing the number of tests required to solve a problem will have an impact on the efficiency of practical algorithms.

We conclude the section by arguing the difficulty of a coevolutionary problem relates directly to the order structure of the set of maximal candidates and the set of maximally-informative tests.

Throughout this section we will consider coevolutionary optimization problems which are expressed with a function $p : S \times T \to R$. The only constraints we place on p is that R be a preordered set. We will write the order on R as $\leq_R$.

2.1 Solution As Set of Maximal Candidates

A common class of problems attacked with GAs start with a function $f : S \to \mathbb{R}$, with the task of finding elements of S which maximize f. Similarly, in a common class of MOO problems, one starts with a set of functions $f_i : S \to \mathbb{R}$, and the task is to find the Pareto front of the f_i. In fact, the Pareto front is a type of maximum too:

Proposition 2.1 (MOO as Maximization) *The Pareto front of a set of objectives $f_i : S \to \mathbb{R}$ $(1 \leq i \leq n)$ is $\widehat{S}_{\langle f_1,\ldots,f_n\rangle}$, the set of maximal elements of the preorder induced on S by the function $\langle f_1, \ldots, f_n\rangle$ into the partial order $\mathbb{R}^n$.*

Proof The Pareto front consists of those $\widehat{s} \in S$ which are not dominated by any other $s \in S$. Define a preorder $\preceq$ on S as follows: $s \preceq s' \Leftrightarrow \forall i,\ f_i(s) \leq f_i(s')$ for all $s, s' \in S$. $s \preceq s'$ expresses that s' dominates or is equal to s. Observe that $s \preceq s' \Leftrightarrow s \leq_{\langle f_1,\ldots,f_n\rangle} s'$ (see definition A.9). The non-dominated front is then $F = \{\widehat{s} \in S \mid \forall s \in S, s \preceq \widehat{s}\}$. The condition $\forall s \in S, s \preceq \widehat{s}$ is logically equivalent to the condition $\forall s \in S, \widehat{s} \preceq s \Rightarrow s \preceq \widehat{s}$. Consequently, we have that $F = \{\widehat{s} \in S \mid \forall s \in S, \widehat{s} \preceq s \Rightarrow s \preceq \widehat{s}\} = \{\widehat{s} \in S \mid \forall s \in S, \widehat{s} \leq_f s \Rightarrow s \leq_f \widehat{s}\}$, where $f = \langle f_1, \ldots, f_n\rangle$. However, the latter set is $\widehat{S}_f$. Thus, we have shown $F = \widehat{S}_f$. □

As a result of this proposition, we can view MOO as a form of maximization. Likewise, we can also view coevolutionary optimization problems as maximization problems. The critical step is to curry the function $p : S \times T \to R$ on T to produce a function $\lambda t.p : S \to [T \to R]$. This function precisely expresses the association between a candidate, which is an element $s \in S$, and its "vector" of objective values $\lambda t.p(s)$, which is an element of $[T \to R]$.

We can order the range $[T \to R]$ using the pointwise order $\leq_{pw}$.[2] Furthermore, we can pull the order on $[T \to R]$ back through the curried function $\lambda t.p$ to produce an order on S. This pulled back order expresses the practice in multi-objective optimization of ordering two individuals $s_1, s_2 \in S$ by comparing their vectors of objective valued. Namely, $s_1 \preceq s_2$ exactly when $\lambda t.p(s_1) \leq_{pw} \lambda t.p(s_2)$.

Now that we have a preorder on S, we propose the set of maximal elements as a solution to the problem $p : S \times T \to R$. Formally,

Definition 2.2 (Maximal Candidates) The set of maximal candidates of the coevolutionary optimization problem $p : S \times T \to R$ is $\mathsf{S}_p = \widehat{S}_{\lambda t.p}$. Explicitly, $\mathsf{S}_p = \{\hat{s} \in S \mid \forall s \in S, [\forall t \in T, p(\hat{s},t) \not\geq_R p(s,t)] \Rightarrow [\forall t \in T, p(s,t) \not\geq_R p(\hat{s},t)]\}$. We will call $\mathsf{S}_p \subseteq S$ the solution set of the problem p.

Here are two examples illustrating the definition:

Example 2.3 *Rock-paper-scissors*

In this simple game, $S = \{rock, paper, scissors\}$, $T = S$ and $R = \{0 < 1\}$. According to the rules of the game, the result of comparing *rock* with *scissors*, for example, is that *rock* wins. We therefore give $p : S \times S \to R$ as the matrix:

	rock	*paper*	*scissors*
rock	1	0	1
paper	1	1	0
scissors	0	1	1

Then the functions $\lambda t.p(s)$ are the rows of the matrix. Comparing these rows pointwise, we see they are all incomparable. Consequently, in rock-paper-scissors, $\mathsf{S}_p = \{rock, paper, scissors\} = S$. $\square$

Example 2.4 *GA optimization and MOO*

Consider optimization problems in which we have objectives $f_i : S \to \mathbb{R}$ for $1 \leq i \leq n$. We can use these objectives to define a payoff function $p(s,s') = f(s) - f(s')$, where $f = \langle f_1, \ldots, f_n \rangle$. For all $s_1, s_2, s' \in S$, $f(s_1) - f(s') \leq^n f(s_2) - f(s') \Leftrightarrow f(s_1) \leq^n f(s_2)$ by adding $f(s')$ to both sides of the inequality. It follows, therefore, that $\widehat{s}$ is a maximal element with respect to $S_{\lambda t.p}$ if and only if $\widehat{s}$ is a maximal element of the function f. As a result, the solution set S_p is exactly the Pareto front of f (see proposition 2.1). $\square$

In light of the observation that definition 2.2 generalizes common solution concepts used in MOO and GA optimization, it is a natural notion of solution for coevolution as well. This particular solution concept is independent of algorithm choices, in the same way that the problem statement "find the maxima of the function $f : S \to \mathbb{R}$" is independent of which flavor of genetic algorithm one uses to solve it. In that respect, this notion of solution lies at a more abstract level than other solution concepts in common use such as "maximize average fitness over the population."

[2] Proposition 2.1 suggests $\leq_{pw}$ really is the appropriate order to use.

2.2 Maximally-Informative Test Set

Now we define the informativeness order among tests. The impact of the definition is expressed in theorem 2.7, where we show the maxima of this informativeness relation are sufficient for inducing the solution set. Finally, we discuss the structure of the maximally-informative test set as a measure for categorizing coevolutionary optimization problems.

Let $\leq$ be an order on a set S. Recall the similarity relation $\sim$, defined $a \sim b \Leftrightarrow a \leq b \wedge b \leq a$, tells us which elements in S look "equal" according to $\leq$. We can now define a relation among the possible orders on S.

Definition 2.5 (Informativeness) Let $\leq_1$ and $\leq_2$ be two orders on S, and let $\sim_1$ and $\sim_2$ be the corresponding similarity relations. Say $\leq_2$ is *more informative than* $\leq_1$, written $\leq_1 \preceq \leq_2$, if $\leq_1 \subseteq \leq_2$ and $\sim_2 \subseteq \sim_1$. If we write $\leq_1 \subseteq^{=} \leq_2$ for the latter condition, then $\preceq = \subseteq \cap \subseteq^{=}$.

Roughly speaking, to be informative, an order should have neither incomparable elements nor equal elements. The idea behind the definition is that a test which shows two candidates to be incomparable, or shows them to be equal, does not give us any information about how they relate to one another. The relation $\subseteq$ tells us when one order has fewer incomparable elements; the relation $\subseteq^{=}$ tells us when an order has fewer equal elements. Therefore, the intersection $\subseteq \cap \subseteq^{=}$ tells us about both incomparability and equality.

We can use $\preceq$ to order $[S \to R]$. Given $f, g \in [S \to R]$, write $f \preceq g$ when $S_f \preceq S_g$. Now we are in a position to describe the maximally-informative test set. In words, it is the set of maximal elements in T with respect to the pullback of the informativeness order on $[S \to R]$. Formally,

Definition 2.6 (Maximally-Informative Test Set) Let $p : S \times T \to R$ represent a coevolutionary optimization problem, and let $\lambda s.p : T \to [S \to R]$ be the curried form of p. Let $[S \to R]$ have the informativeness order $\preceq$. Pull this order back through $\lambda s.p$ into T, and write the resulting ordered set as $T_{\preceq}$. Then the maximally-informative test set for this problem is $\mathsf{T}_p = \widehat{T_{\preceq}} \subseteq T$.

That definition 2.6 is useful is borne out by the following theorem:

Theorem 2.7 *Let $p : S \times T \to R$ be a coevolutionary optimization problem, and consider $p\mid_{\mathsf{T}_p}: S \times \mathsf{T}_p \to R$, the restriction of p to the maximally-informative tests. For brevity, write q for $p\mid_{\mathsf{T}_p}$. Then $S_{\lambda t.p} \cong S_{\lambda t.q}$; in other words, the maximally-informative set of tests induces the same order on S as the full set of tests T. Consequently, it also induces the same set of maximal candidates.*

Proof We must show $S_{\lambda t.p} \cong S_{\lambda t.q}$. Explicitly, this isomorphism is equivalent to the following: for all $s_1, s_2 \in S$, $\lambda t.p(s_1) \leq_{pw} \lambda t.p(s_2) \Leftrightarrow \lambda t.p\mid_{\mathsf{T}_p}(s_1) \leq_{pw} \lambda t.p\mid_{\mathsf{T}_p}(s_2)$. Unrolling still further, this equivalence translates into: for all $s_1, s_2 \in S, [\forall t \in T, p(s_1,t) \not\geq_R p(s_2,t) \Leftrightarrow \forall t \in \mathsf{T}_p, p(s_1,t) \not\geq_R p(s_2,t)]$.

The forward implication holds trivially, because $\mathsf{T}_p \subseteq T$. Consequently, we focus our attention on showing $\forall t \in \mathsf{T}_p, p(s_1,t) \not\geq_R p(s_2,t) \Rightarrow \forall t \in T, p(s_1,t) \not\geq_R p(s_2,t)$, for all $s_1, s_2 \in S$. If we can show this implication, we have the result. Let $t \in T$. By definition of T_p, $\exists \widehat{t} \in \mathsf{T}_p$ such that $t \preceq \widehat{t}$. Assume $p(s_1,t) \geq_R p(s_2,t)$; then it follows $p(s_1,\widehat{t}) \geq_R p(s_2,\widehat{t})$, because $\widehat{t}$ is more informative than t. Consequently, we have that $t \preceq \hat{t}$ and $p(s_1,\hat{t}) \not\geq_R p(s_2,\hat{t})$ imply $p(s_1,t) \not\geq_R p(s_2,t)$. The latter holds for any $t \in T$ and $s_1, s_2 \in S$; therefore, we have our result. $\square$

Theorem 2.7 shows that we do not need to use the full set of tests T in order to distinguish individuals in S. In fact, the maximally-informative test set T_p will induce the same order on S and so the same maximal candidates. If T_p is a strict subset of T, then we can in theory solve the same problem p using fewer tests.

Here are some illustrative examples:

Example 2.8 *Rock-paper-scissors, revisited*

In the rock-paper-scissors incidence matrix (see example 2.3), the columns are the $\lambda s.p(t)$. Reading left to right, the induced orders are $\{scissors < rock \sim paper\}$, $\{rock < paper \sim scissors\}$, and $\{paper < rock \sim scissors\}$. None of these orders is a suborder of another. It follows that $\mathsf{T}_p = \{rock, paper, scissors\} = T$. □

Example 2.9 Consider the formal game where $S = T = \{a,b,c\}$ and $R = \{0 < 1 < 2, x\}$, where the outcome x is incomparable to 0, 1 and 2. p is given by the matrix

	a	b	c
a	0	1	0
b	1	0	1
c	2	1	x

The orders induced on S are, left to right, $\{a < b < c\}$, $\{b < a \sim c\}$, and $\{a < b, c\}$. Observe that $c \preceq a$, so $\mathsf{T}_p = \{a,b\} \neq T$. Notice also that $\mathsf{S}_p = \{c\}$, so this example shows S_p and T_p can be distinct. That is, solutions need not make good tests. □

2.3 Categorizing Problems Using Tests

T_p is too big. It could be there are tests $t_1, t_2 \in \mathsf{T}_p$ such that $t_1 \sim t_2$, where we are taking $\sim$ with respect to the informativeness order. In that case, we really only need one representative from each equivalence class of $\sim$. This observation leads us to:

Definition 2.10 *The value $|\mathsf{T}_p/\sim|$ is the test-dimension of the problem p.*

Then we have the following:

Theorem 2.11 *Let $n \geq 1$, and let the function $f : S \to \mathbb{R}^n$ be an optimization problem. Define $p : S \times S \to \mathbb{R}^n$ by $p(s,s') = f(s) - f(s')$ for all $s, s' \in S$. Then p has test-dimension 1.*

Proof The observation in example 2.4 that $f_i(s_1) - f(s') \leq f(s_2) - f(s') \Leftrightarrow f(s_1) \leq f(s_2)$ leads to the result, because all tests s' are equivalent. □

Remark A MOO problem with n objectives looks like it should have test dimension n. However, in theorem 2.11 we are using the objectives to compare pairs of individuals. Then the *individuals* are the tests, not the objectives themselves. In that case, any single individual will do as a test. If we were to treat the objectives themselves as tests, then a MOO problem with n objectives would indeed have test dimension n.

We interpret theorem 2.11 as saying that “difficult” coevolutionary optimization problems have test-dimension > 1. Rock-paper-scissors has test-dimension 3, for example. Therefore, we can categorize problems on the basis of the test set structure, independently of any search algorithms we might employ. Furthermore, problems with simple test set structure are likely to be simpler to solve in practice. For instance, rock-paper-scissors has an intransitive cycle which is reflected in its test set structure. As we saw, multi-objective optimization problems, by contrast, have a particularly simple test set structure. Indeed, this structure is spelled out explicitly in the definition of the problem, whereas in the typical coevolution application, the test set structure is not known in advance but is implicit in the function $p : S \times T \to R$. See Juillé's discussion of the cellular automaton majority function problem in [9]. The test-dimension is a simple numerical measure of the test set structure which gives us some information about the difficulty of a problem.

3 Pareto Dominance and Intransitivity

As an application of the concepts in section 2, we consider the special class of problems represented by functions $p : S \times S \to \{0 < 1\}$. Functions of this form arise in the problem of learning deterministic game-playing strategies through self-play, or in perfect-information, zero-sum games encountered in game theory.

In this setting, we can view p as representing a relation on S. To be specific, the relation is the subset $p^{-1}(1) \subset S \times S$. In other words, a relates to b exactly when $p(a,b) = 1$. We will write this relation aR_pb. Intuitively, we interpret aR_pb as "a loses or draws to b." If p represents a game, the usual notion of transitivity for a game is equivalent to the transitivity of the relation R_p. A critical problem which arises in the dynamics of coevolutionary algorithms stems from intransitive superiority cycles. Such a cycle always occurs when p is not transitive because, in that case, there is a finite set of players $a_i \in S$ such that $p(a_{i-1}, a_i) = 1$ and $p(a_i, a_{i-1}) = 0$ for $1 \leq i \leq n-1$, but $p(a_n, a_0) = 1$. Transitivity would dictate $p(a_0, a_n) = 1$. Coevolutionary dynamics operating on such a game can become stuck cycling amongst the a_i without ever making "real" progress. See [14] for a discussion of this issue.

One of the promises of Pareto coevolution is that it can help with intransitive cycles by revealing the true relationship among the individuals in a cycle.[3] We will show there is a close relationship between the transitivity of p and the Pareto dominance relation. In particular, we will show in theorem 3.3 that R_p is a preorder if and only if Pareto dominance over p is equal to R_p itself. In other words, Pareto dominance gives us different information than the payoff function p exactly when the latter is intransitive.[4] One conclusion we can draw from this fact is that Pareto coevolution can detect intransitive cycles. Another conclusion, corollary 3.6, is a potentially useful negative result. At first glance, one might think multiple iterations of the Pareto relation construction (definition 3.1) would provide ever more information about a problem. Corollary 3.6 shows this is not the case. Applying Pareto dominance to a relation once sometimes produces a new relation, but applying the construction twice is always equivalent to applying it once.

We will prove all results with respect to any relation R on a set S, R_p then being a special case. We begin by constructing a new relation over S from R which captures the Pareto dominance relation:

Definition 3.1 (Pareto Relation) The *Pareto relation over R*, written $\preceq_R$, is defined as follows. For all $a, b \in S$, $a \preceq_R b$ if and only if $xRa \Rightarrow xRb$ for all $x \in S$.

[3] We would like Pareto dominance to reveal they are incomparable.

[4] It is worth emphasizing that these results apply only in this specific context, not in general. We make critical use of the symmetry in roles between candidates and tests, and the fact that R contains only two comparable outcomes.

Remark The Pareto relation over R_p corresponds to our usual notion of Pareto dominance. Intuitively, the definition says "b dominates, or is equal to, a if, whenever x draws or loses to a, then x draws or loses to b as well."

The following proposition will be useful:

Proposition 3.2 *$\preceq_R$ is a preorder.*

Proof Clearly $xRa \Rightarrow xRa$ for all $x \in S$; thus, $a \preceq_R a$, making $\preceq_R$ reflexive. If $a \preceq_R b$ and $b \preceq_R c$ for some $a,b,c \in S$, we want to show $a \preceq_R c$. But this follows from the transitivity of $\Rightarrow$: $a \preceq_R b$ means $xRa \Rightarrow xRb$, while $b \preceq_R c$ means $xRb \Rightarrow xRc$, for all $x \in S$. So if we have $xRa \Rightarrow xRb$ and $xRb \Rightarrow xRc$, it follows $xRa \Rightarrow xRc$, in other words, $a \preceq_R c$. $\preceq_R$ is thus transitive, so is a preorder. □

Now we have the theorem:

Theorem 3.3 *$R = \preceq_R$ if and only if R is a preorder*

Proof The theorem is a consequence of the following two lemmas:

Lemma 3.4 *R is reflexive if and only if $\preceq_R \subseteq R$.*

Proof To prove the forward implication, assume R is reflexive, and imagine $a \preceq_R b$. We must show aRb. By definition, $a \preceq_R b$ means $xRa \Rightarrow xRb$ for all $x \in S$. In particular, since aRa, it follows aRb. Thus, $\preceq_R \subseteq R$.

For the reverse implication, assume $\preceq_R \subseteq R$. Since $\preceq_R$ is a preorder (proposition 3.2), it is reflexive; it follows at once R must be also. □

Lemma 3.5 *R is transitive if and only if $R \subseteq \preceq_R$.*

Proof First consider the forward implication. Assume R is transitive; we must show $R \subseteq \preceq_R$. Let aRb. By transitivity of R, we know that for any $s \in S$, xRa and aRb imply xRb. It follows $a \preceq_R b$. Thus we have shown $R \subseteq \preceq_R$.

Now to the reverse implication. Assume $R \subseteq \preceq_R$. We want to show R is transitive. So, let aRb and bRc. Because $R \subseteq \preceq_R$, the latter implies $b \preceq_R c$, or equivalently, $xRb \Rightarrow xRc$. Coupled with the assumption that aRb, we have immediately that aRc. In other words, we have shown R is transitive, as needed. □

An important consequence of theorem 3.3 is that iterating the Pareto relation construction "tops out" after two applications. Formally,

Corollary 3.6 *The mapping $R \mapsto \preceq_R$ is idempotent; i.e., for any relation R on a set S, $\preceq_{\preceq_R} = \preceq_R$.*

Proof Follows directly from proposition 3.2 and theorem 3.3. □

Corollary 3.6 tells us that repeating the Pareto dominance construction does not reveal any new information. For instance, if R represents the rock-paper-scissors game, then $\preceq_R$ is the identity relation I since each strategy is incomparable to each other. $\preceq_I = I$, as well, so $\preceq_{\preceq_R} = \preceq_R$.

4 Discussion and Future Work

Our framework was aimed at understanding the static features of a class of coevolutionary optimization problems, prior to any algorithm choices. Nonetheless, the insights gained from our static analysis informs our understanding of coevolutionary dynamics. By separating the definition of a coevolutionary optimization problem from the dynamics of search algorithms, we have gained insight into issues which plague coevolutionary dynamics. In particular, we can state in a precise way what it means for a coevolutionary algorithm to make progress: if, at time t_2, the algorithm has found individuals which Pareto dominate the individuals at time t_1, then the algorithm has made progress. We stated, in definition 2.2, what it means for an algorithm to solve a coevolutionary optimization problem: it has solved the problem if it has found the set of maximal candidates S_p. In general, we cannot know for certain if we have made progress or found the solution set if we are observing only subsets of S and T, as a real algorithm would . However, these ideas do offer approximations which may work in many practical situations. It would be worthwhile to examine in more depth when such approximations are meaningful.

We introduced and motivated our work with the observation that intransitivities produce difficult issues for coevolutionary algorithms. We subsequently observed, in section 3, that the Pareto dominance relation is closely tied to the transitivity of the underlying payoff structure. It is possible to extend this result to a wider class of payoff structures. More importantly, we feel there is important information to be found in considering local views of a payoff function $p : S \times T \to R$. For instance, imagine $S_i \subset S$ and $T_i \subset T$ are the populations at time i in the run of a coevolutionary algorithm. The act of testing all candidates in S_i against all tests in T_i exposes a restricted view of p; mathematically, this information can be written $p|_{S_i \times T_i}$. A critical question arising in any coevolutionary optimization algorithm is: can we use the information in $p|_{S_i \times T_i}$ to deduce properties of p itself in a way which leads us to the solution set S_p? A preliminary result in this direction would be to show that any two incomparable (non-dominated) individuals in the Pareto dominance relation must lie in a cycle in the underlying game. Theorem 3.3 goes part way towards this result. This result would be useful to algorithms because it is possible to deduce Pareto non-dominance with respect to p using only the local information in $p|_{S_i \times T_i}$; in other words, once we had found that two individuals were non-dominated, we would know they lie in a cycle and we should be cautious until we have found a candidate dominating them.

The results in section 3 are specific to the outcome order $R = \{0 < 1\}$. If we include more outcomes in R, we lose theorem 3.3 and in particular corollary 3.6. Apparently, when the outcome order has more than two elements the situation becomes more complicated. Puzzling out what that might be is a topic of current research.

Finally, our formulation of coevolutionary optimization problems using a function of form $p : S \times T \to R$ excludes multi-player games and cooperative coevolution. We are also currently investigating how to extend the framework to include these more complicated approaches.

Acknowledgments

We wish to thank the members of the DEMO lab for their unrelenting help and support. In particular, frequent conversations with Sevan Ficici, Edwin de Jong, and Richard Watson were vital to bringing these ideas to fruition.

A Notation and Mathematical Background

Here we briefly recall some definitions from discrete mathematics, particularly from the theory of ordered sets, also establishing notational conventions we will use. We first define ordered sets as mathematical objects; we then examine some ways these objects combine and relate. See [11] for an elementary introduction to ordered sets, and [2] for information on more advanced concepts such as pullbacks.

A.1 Ordered Sets

Recall the Cartesian product of two sets S and T is the set of ordered pairs $S \times T = \{(s,t) \mid s \in S, t \in T\}$. A *binary relation* on a set S is a subset $Q \subset S \times S$. Given $s_1, s_2 \in S$ and a binary relation Q on S, we say s_1 and s_2 *relate under* Q, written $s_1 Q s_2$, when $(s_1, s_2) \in Q$. We also say that two elements s_1 and s_2 which relate under Q are *comparable* according to Q; otherwise, they are *incomparable.*

A binary relation Q on a set S is *reflexive* when, for all $s \in S$, sQs. Q is *transitive* if, for all $s_1, s_2, s_3 \in S$, $s_1 Q s_2$ and $s_2 Q s_3$ imply $s_1 Q s_3$. Q is *anti-symmetric* if for all $s_1, s_2 \in S$, $s_1 Q s_2$ and $s_2 Q s_1$ imply $s_1 = s_2$.

A binary relation Q is a *preorder* if it is both reflexive and transitive. If a preorder is also anti-symmetric, it is a *partial order*. If, finally, all pairs of individuals from S are comparable according to the partial order Q, then Q is a *total order* or *linear order*. Note that, in analogy with partial functions, a partial (or pre-) order need not define relations between all pairs of members of S, whereas a total order must. We will call S a *preordered set* or simply an *ordered set* when it is equipped with a relation which is a pre-, partial, or total order. We will write the order $\leq_S$ when we need to refer to it directly.

A binary relation Q on a set S expresses the same information as a directed graph with vertices S; the elements of Q correspond to the edges of the graph. Moreover, we can think about graphs in terms of their incidence matrices. Consequently, one can think of these concepts in any of these ways, as convenient.

Two ordered sets can be combined in a number of ways. For our purposes, the two most useful are Cartesian product and intersection.

Definition A.1 (Cartesian Product of Preordered Sets) Let S and T be preordered sets. As sets, $\leq_S \times \leq_T \subset S \times S \times T \times T$. Hence, we can interpret $\leq_S \times \leq_T$ as a relation between $S \times S$ and $T \times T$, relating ordered pairs on S to ordered pairs on T. $\leq_S \times \leq_T$ will be an order of some kind, but as shown in the example below, the type of order may change. Thus, to be precise we define the Cartesian product of two preordered sets $(S, \leq_S)$ and $(T, \leq_T)$ by $(S, \leq_S) \times (T, \leq_T) = (S \times T, \leq_S \times \leq_T)$. We will write this product simply as $S \times T$. If we take the Cartesian product of a preordered set S with itself, we write it as S^2 and the relation in particular as $\leq_S^2$. We define S^n and $\leq_S^n$ similarly.

Example A.2 The set of real numbers $\mathbb{R}$ is totally ordered by the usual order $\leq$. $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$ is the familiar Cartesian plane. The order on $\mathbb{R}^2$ is $\leq \times \leq = \leq^2$. Unrolling the definition: $(x_1, y_1) \leq^2 (x_2, y_2) \Leftrightarrow x_1 \leq y_1 \wedge x_2 \leq y_2$. It is straightforward to verify $\leq^2$ is a partial order. It is not a total order because, for example, $(0,1)$ and $(1,0)$ are incomparable with respect to $\leq^2$. $\square$

Definition A.3 (Intersection of Preordered Sets) Let S and T be preordered sets. As a set, $\leq_S \cap \leq_T \subset (S \cap T) \times (S \cap T)$. Thus $\leq_S \cap \leq_T$ is a relation on $S \cap T$ which can be verified to be an order. Consequently, we define the intersection of two orders S and T to be $(S \cap T, \leq_S \cap \leq_T)$. As with the Cartesian product, we will write this intersection as $S \cap T$ or $\leq_S \cap \leq_T$.

Whereas a total order can have at most one maximum, a partial or preorder can have many maxima, referred to as *maximal elements.*[5]

Definition A.4 (Maximal Elements in Preordered Sets) A *maximal element* of the preordered set S is any element $\widehat{s} \in S$ with the property that, for all other $s \in S$, $\widehat{s} \leq_S s \Rightarrow s \leq_S \widehat{s}$. We will write $\widehat{S}$ for the set of all maximal elements of the preordered set S. It is possible $\widehat{S} = \emptyset$ or $\widehat{S} = S$.

The following two relations can be derived from any order.

Definition A.5 (Strict Relation) Let S be an ordered set. Define $<_S$ as follows: for all $a, b \in S$, $a <_S b$ if and only if $a \leq_S b$ and $b \not\leq_S a$.

Definition A.6 (Similarity Relation) Let S be an ordered set. Define the equivalence relation $\sim_S$ on S as follows: for all $a, b \in S$, $a \sim_S b$ if and only if $a \leq_S b$ and $b \leq_S a$.

In a partial order, the similarity relation is the same as equality because of anti-symmetry. However, the same is not the case in a preordered set, and therein lies the difference between the two concepts. We can rephrase definitions x and y in more familiar terms using $\sim_R$. For instance, a maximal element is a $\hat{s} \in S$ such that for all $s \in S$, $\hat{s} \leq_S s \Rightarrow \hat{s} \sim_S s$. Also, $a <_S b$ if and only if $a \leq_S b$ and $a \not\sim_S b$.

We can make any preorder into a partial order by taking the quotient with respect to $\sim_S$:

Definition A.7 (Quotient Order) Let S be an ordered set. Let $S/{\sim_S}$, read "S modulo (the equivalence) $\sim_S$" be the set of equivalence classes of S under $\sim_S$. Given an $a \in S$, write $[a]$ for the equivalence class of a under $\sim_S$; in particular, $[a] = \{a' \in S \mid a' \sim_S a\}$. We define an order on $S/{\sim_S}$, which we will also write $\leq_S$, as follows. If $[a]$ and $[b]$ are two equivalence classes in $S/{\sim_S}$, then $[a] \leq_S [b]$ in $S/{\sim_S}$ if and only if $a \leq_S b$ in S.

The following proposition shows definition A.7 is reasonable:

Proposition A.8 *The order $\leq_S$ on $S/{\sim_S}$ is well-defined; furthermore, it is a partial order.*

Proof Let $a_1, a_2, b_1, b_2 \in S$ be such that $a_1 \sim_S a_2$, $b_1 \sim_S b_2$ and $a_1 \leq_S b_1$. To show well-definedness of $\leq_S$ on $S/{\sim_S}$, it suffices to show $a_2 \leq_S b_2$. By similarity, we know $a_2 \leq_S a_1$ and $b_1 \leq_S b_2$. We therefore have the following chain: $a_2 \leq_S a_1 \leq_S b_1 \leq_S b_2$. Then $a_2 \leq_S b_2$ by transitivity, and $\leq_S$ is a well-defined relation on $S/{\sim_S}$. The reflexivity and transitivity of $\leq_S$ on $S/{\sim_S}$ are clear from the definition. What remains is to show this relation is antisymmetric. However, if $a \leq_S b$ and $b \leq_S a$ in S, then $a \sim_S b$, and it follows immediately that $[a] = [b]$. Consequently, $[a] \leq_S [b]$ and $[b] \leq_S [a]$ imply $[a] = [b]$. Thus $\leq_S$ is a partial order on $S/{\sim_S}$. $\square$

A.2 Functions into Ordered Sets

Let S and T be preordered sets, and let $f : S \to T$ be a function. f is *monotone*, or *monotonic*, if $s_1 \leq_S s_2 \Rightarrow f(s_1) \leq_T f(s_2)$ for all $s_1, s_2 \in S$. The intuition behind this definition is that f preserves

[5] There is also a definition of maximum in partial and preorders which must be larger than all other elements in the set. However, for our purposes maximal elements are much more useful.

	f_1	f_2	f_3
a	0	2	1
b	1	1	0

Figure 1 $\leq_{pw}$ and $\subseteq$ can be distinct orders on $[S \to R]$. Let $S = \{a,b\}$, $R = \{0 < 1 < 2\}$. Observe that $f_1 \leq_{pw} f_2$, but $f_1 \not\subseteq f_2$; and, $f_2 \subseteq f_3$, but $f_2 \not\leq_{pw} f_3$.

order; put differently, passage through the function f does not destroy any pairwise relations. If we regard S and T as graphs, then a monotone f is exactly a graph homomorphism. f is an *isomorphism* of preordered sets if f is a monotone bijection and, additionally, f^{-1} is monotone. Two isomorphic preordered sets are "the same;" that is, they typify the same order structure, possibly differing in how their elements are labeled. We will write $S \cong T$ to indicate S and T are isomorphic preordered sets. Isomorphic preordered sets have "the same" maximal elements; i.e., if $f : S \to T$ is an isomorphism of preordered sets, then $f(\widehat{S}) = \widehat{T}$.

Let S be a set, R a preordered set, and let $f : S \to R$ be a function; we will call f a *function into the preordered set R*. Given such a function f, we can *pullback* the order of R into S [2]. To be more precise,

Definition A.9 (Pullback Orders) Let $f : S \to R$ be a function into the preordered set R. Define the preorder $\leq_f$ on S as follows: $s_1 \leq_f s_2 \Leftrightarrow f(s_1) \leq_R f(s_2)$ for all $s_1, s_2 \in S$. We will write the resulting preordered set $(S, \leq_f)$ as S_f; we will refer to it as the *preorder induced on S by* f[6]. As defined, $\leq_f$ is the largest preorder on S making the function f monotone.

Following the convention in domain theory [1], write $[S \to R]$ for the set of all functions from S to R. If R is a preordered set, we can order $[S \to R]$ in several ways. First, we consider the *pointwise order:*

Definition A.10 (Pointwise Order) Two functions $f, g \in [S \to R]$ lie in order pointwise, which we write $f \leq_{pw} g$ or just $f \leq g$, if for all $s \in S$, whenever $f(s)$ and $g(s)$ related, it must be that $f(s) \leq_R g(s)$. Note that another way to state this condition is that for all $s \in S$, $f(s) \not\geq_R g(s)$. The pointwise order is the default order on $[S \to R]$. When we speak of $[S \to R]$ as if it were ordered, we assume it has the pointwise order.

The second order we consider on $[S \to R]$ is via suborder.:

Definition A.11 (Suborder Order) Recall that an element $f \in [S \to R]$ corresponds to a preorder on S, namely the pullback order S_f defined above. Given two functions f and g, we can ask whether $S_f \subseteq S_g$. Therefore, we write $f \subseteq g$ when $S_f \subseteq S_g$. Explicitly, $f \subseteq g$ holds when, for all $s_1, s_2 \in S$, $f(s_1) \leq_R f(s_2) \Rightarrow g(s_1) \leq_R g(s_2)$.

Figure 1 shows $\leq_{pw}$ and $\subseteq$ are distinct orders in general.

The action of *currying* a function, borrowed from the lambda calculus.[7] will be useful:

[6] Occasionally we will use the same symbol for the induced preorder on S. In other words, if $\leq_R$ is the order on R, we will sometimes write $s \leq_R s'$ instead of $s \leq_f s'$. The context will make clear what we mean by this abuse of notation.

[7] Take note that lambda application comes from the adjunction between the Cartesian product functor and the exponential functor in any Cartesian closed category; see [2] for details.

Definition A.12 (Currying) Given any function $f : A \times B \to C$, there is an associated *curried* function $A \to [B \to C]$. The lambda calculus makes heavy use of this association; thus, we will suggestively write this function $\lambda b.\, f : A \to [B \to C]$ and call it f *curried on B*. It is defined as follows. For any $a \in A$, the function $\lambda b.\, f(a)$ maps $b \in B$ to $f(a,b) \in C$. $\lambda a.\, f$ is defined similarly.

References

[1] Samson Abramsky and Achim Jung. Domain theory. In S. Abramsky, D. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science Volume 3*, pages 1–168. Oxford University Press, 1994.

[2] Michael Barr and Charles Wells. *Category Theory for Computing Science*. Prentice Hall International Series in Computer Science. Prentice Hall, New York, 1st edition, 1990.

[3] Dave Cliff and Geoffrey F. Miller. Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations. In *European Conference on Artificial Life*, pages 200–218, 1995.

[4] S. G. Ficici and J. B. Pollack. A game-theoretic approach to the simple coevolutionary algorithm. In Hans-Paul Schwefel Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, editor, *Parallel Problem Solving from Nature - PPSN VI 6th International Conference*, Paris, France, 16-20 2000. Springer Verlag.

[5] Sevan G. Ficici and Jordan B. Pollack. Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states. In Christoph Adami, Richard K. Belew, Hiroaki Kitano, and Charles Taylor, editors, *Artificial Life VI: Proc. of the Sixth Int. Conf. on Artificial Life*, pages 238–247, Cambridge, MA, 1998. The MIT Press.

[6] Sevan G. Ficici and Jordan B. Pollack. Pareto optimality in coevolutionary learning. In *European Conference on Artificial Life*, pages 316–325, 2001.

[7] Carlos M. Fonseca and Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[8] W. Daniel Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. In Christopher G. Langton, Charles Taylor, J. Doyne Farmer, and Steen Rasmussen, editors, *Artificial Life II*, volume X, pages 313–324. Addison-Wesley, Santa Fe Institute, New Mexico, USA, 1990 1992.

[9] Hugues Juille and Jordan B. Pollack. Coevolving the ideal trainer: Application to the discovery of cellular automata rules. In John R. Koza, Wolfgang Banzhaf, Kumar Chellapilla, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max H. Garzon, David E. Goldberg, Hitoshi Iba, and Rick Riolo, editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 519–527, University of Wisconsin, Madison, Wisconsin, USA, 22-25 1998. Morgan Kaufmann.

[10] Christopher D. Rosin. *Coevolutionary search among adversaries*. PhD thesis, University of California, San Diego, San Diego, CA, 1997.

[11] Edward R. Scheinerman. *Mathematics: A Discrete Introduction*. Brooks/Cole, Pacific Grove, CA, 2000.

[12] Karl Sims. Evolving virtual creatures. *Computer Graphics*, 28(Annual Conference Series):15–22, 1994.

[13] Michael D. Vose. *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, Cambridge, MA, 1999.

[14] Richard Watson and Jordan 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*, San Francisco, CA, 2001. Morgan Kaufmann Publishers.