Intransitivity in coevolution

Edwin D. de Jong

Universiteit Utrecht
Decision Support Systems Group
The Netherlands
http://www.cs.uu.nl/∼dejong
dejong@cs.uu.nl

Abstract. We review and investigate the current status of intransitivity as a potential obstacle in coevolution. Pareto-Coevolution avoids intransitivity by translating any standard superiority relation into a transitive Pareto-dominance relation. Even for transitive problems though, cycling is possible. Recently however, algorithms that provide monotonic progress for Pareto-Coevolution have become available. The use of such algorithms avoids cycling, whether caused by intransitivity or not. We investigate this in experiments with two intransitive test problems, and find that the IPCA and LAPCA archive methods establish monotonic progress on both test problems, thereby substantially outperforming the same method without an archive.

Coevolution offers algorithms for problems where the performance of individuals can be evaluated using tests [1–7]. Since evaluation in coevolution is based on evolving individuals, coevolution setups can suffer from inaccurate evaluation, leading to problems such as over-specialization, Red Queen dynamics, and disengagement [8, 9].

A problem feature that has received particular interest in the past is that of *intransitivity* [9]. A relation R is *transitive* if $aRb \wedge bRc$ implies aRc; if this cannot be guaranteed, the relation is intransitive. An example of a problem where the relation used to compare individuals is intransitive, is *Rock, Paper, Scissors*; while *scissors* beats *paper* and *paper* beats *rock*, *scissors* is beaten by *rock*. The existence of such intransitive relations in a coevolution problem can lead to *cycling*, i.e. the recurrence of previously visited states of the population.

Intransitivity has been viewed as an inherent feature of coevolution that can render algorithms unreliable. Indeed, the resulting problem of *cycling* has been thought of as an obstacle that could prevent coevolution from becoming a reliable problem solving technique, as attested to by the following quote: "We believe that the cycling problem, like the local minima problem in gradient-descent methods..., is an intrinsic problem of coevolution that cannot be eliminated completely" [10].

Recently, it has been shown that coevolution can in principle approximate *ideal evaluation* [11], i.e. equivalence to evaluation on the space of all tests. This result is based on the solution concept offered by Pareto-Coevolution [12, 13],

consisting of all candidate solutions whose performance cannot be improved on any test without decreasing the individual's outcome on some other test.

Another approach to achieve reliability in coevolution is to use an *archive* to maintain promising candidate solutions and informative tests. If an archive can avoid regress, then generating all possible individuals with non-zero probability guarantees that the algorithm can only make progress and will occasionally do so, thus enabling the coevolutionary goal of open-ended, sustained progress.

Rosin's *covering competitive algorithm* [14], alternates between finding a first-player strategy that beats all second-player strategies in the archive and *vice versa*. This guarantees that regress can be avoided, but the strict criterion of defeating all previous opposition is likely to result in stalling as soon as mutually exclusive tests exist, i.e. tests that cannot all be solved by a single learner but can be solved individually by different learners. Ficici and Pollack's Nash Memory [15] guarantees progress for the solution concept of the Nash Equilibrium. It is limited to symmetric games, but extension to the case of asymmetric games is noted to be straightforward. The Incremental Pareto-Coevolution Archive (IPCA) [16] guarantees progress for Pareto-Coevolution, and is applicable to both symmetric and asymmetric problems. All of the above archive methods can only guarantee progress however if the archive size is unlimited. A layered variant of IPCA was found empirically to produce reliable progress on test problems using a bounded archive [17].

Our aim here is to investigate the role of intransitivity in coevolution in the light of the above developments. It is known that the use of Pareto-coevolution transforms intransitive superiority relations into transitive relations; we provide a concise proof demonstrating this. While this transitive relation provides an appropriate basis for reliable coevolution, cycling is also possible for transitive relations, as shown in a simple example. We discuss the potential of coevolution methods to avoid cycling, and investigate the performance of two recent algorithms on test problems featuring intransitivity.

The structure of this paper is as follows. Section 1 discusses intransitivity and recent results in achieving reliable progress in coevolution. Section 2 presents experimental results, and Section 3 concludes.

1 Intransitivity

1.1 Intransitivity can lead to cycling

A familiar example of intransitivity is provided by the game of *Rock, Paper, Scissors*, where *scissors* beats *paper*, *paper* beats *rock*, and *rock* beats *scissors*. Suppose we administer this problem to the minimal setup that may be called coevolution, namely a hillclimbing individual that learns by self-play. The expected transcript of this experiment if the population is initialized with *Rock* (R), and a newly generated individual replaces the current individual if and only if the former beats the latter is R, P, S, R. As this example demonstrates, the application of a hillclimber to an intransitive problem can lead to cycling.

1.2 Pareto-Coevolution

In Pareto-Coevolution [12, 13], the individuals that are used as tests in evaluating other individuals are used as *objectives*, in the sense of Evolutionary Multi-Objective Optimization. Table 1 shows an example of interactions between three different individuals, A, B, and C. The outcomes in the matrix may be viewed for example as the score of the row player against the column player in a two-player game G with the row player as first player, so that for example $G(a, a) = 0, G(a, b = 1)$. The outcomes can be used to define a relation R that specifies which games result in a win for the first player: aRb, bRa, cRa, cRb. In standard coevolution, this relation is used directly to assess the quality of the different individuals by interpreting it as a 'greater than' relation. Fitness can then for instance be based on the number of wins an individual achieves. However, as the example shows, this direct use of the relation can yield conflicting information (e.g. aRb and bRa), resulting from the fact that first player strategies are compared against second player strategies.

Table 1. Example outcomes for three individuals, a, b, and c. Each table entry lists the outcome of a row player against a column player.

	a	b	c
a	0	1	0
b	1	0	0
c	1	1	0

When Pareto-Coevolution is used, a distinction is made between *learners* (candidate solutions) and *tests* (individuals used to evaluate the quality of learners). If our aim is to find an optimal first player strategy, then individuals functioning as first (row) players are learners, and the second (column) players are tests. This shift in perspective is illustrated by renaming the column players as a', b', and c' respectively to signal their different role when used as a test.

Pareto-Coevolution now evaluates a learner by viewing its outcomes against tests as *objectives*. Two learners are compared using the Pareto dominance relation, defined as follows: an individuals x with objective values x_i dominates another individual y with objective values y_i if and only if:

$$\forall i : x_i \geq y_i \quad \wedge \quad \exists i : x_i > y_i$$

Thus, in the example, c dominates a, as c's score is at least as high as a's score for all tests, and higher for at least one test (a'). Likewise, c also dominates b, while a and b do not dominate any other learners. The dominance relation is a new relation that can be used to evaluate and compare learners. By viewing a learner's outcomes against tests as objectives and comparing learners based on Pareto-dominance, we have obtained a second relation R' that can be derived from R. In this new relation conflicting information, such as $aR'b \wedge bR'a$, cannot occur.

1.3 Pareto-Coevolution transforms intransitive relations into transitive ones

The previous section illustrated how Pareto-Coevolution takes a given relation R between individuals and uses it to define a new relation R' specifying dominance relations between learners, and using test outcomes as objectives. It has been observed that using test outcomes as objectives results in an evaluation function that is transitive in each objective, and that the relation induced on learners by Pareto-Coevolution cannot be intransitive for any fixed set of tests [18, 11]. Bucci and Pollack [19] observe that Pareto-Coevolution induces a preorder relation on learners which can be embedded into $\mathbb{R}^n$, and is therefore transitive.

In summary, Pareto-Coevolution transforms a given, possibly intransitive superiority relation between individuals into a transitive dominance relation. This can be seen with the following brief proof:

Let R be a relation over a set of individuals I. For any $a, b \in I$, aRb may be interpreted as stating that a obtains a positive outcome against b. Without loss of generality, we assume that the first players x in interactions xRy are the learners whose outcome are to be maximized, and the second players y are tests. Then Pareto-Coevolution transforms the initial relation R into a new relation R', defined as the Pareto-dominance over learners that uses the tests as objectives:

$$a_1 R' a_2 \quad \Longleftrightarrow \quad \forall x \in I \quad a_1 R x \geq a_2 R x \quad \wedge \quad \exists x \in I \quad a_1 R x > a_2 R x \tag{1}$$

$$R' \text{ is transitive if } \forall a, b, c \in I \quad aR'b \wedge bR'c \implies aR'c \tag{2}$$

Assume $aR'b \wedge bR'c$. Then

$$\forall x \in I \quad aRx \geq bRx \quad \wedge \tag{3}$$

$$\exists x \in I \quad aRx > bRx \tag{4}$$

$$\forall x \in I \quad bRx \geq cRx \quad \wedge \tag{5}$$

$$\exists x \in I \quad bRx > cRx \tag{6}$$

$$\text{From (3) and (5): } \forall x \in I \quad aRx \geq cRx \tag{7}$$

$$\text{From (4) and (5): } \exists x \in I \quad aRx > cRx \tag{8}$$

$$\text{By combining (7) and (8): } \forall x \in I \quad aRx \geq cRx \quad \wedge \quad \exists x \in I \quad aRx > cRx$$

Therefore $aR'c$. ■

1.4 Transitive games can lead to cycling

Even in a transitive game, cycling is possible. This can be shown by a small example, see the payoff matrix for learners (a1, a2) in Table 2; the payoff for tests (b1, b2) is the inverse. Table 2 shows the sequence of learner and test populations when two size one populations are initialized with learner $a1$ as the learner population and test $b1$ as the test population. Both populations are hillclimbers that maximize their score against the other population. Given $a1$ as a learner, the best strategy for the test population is $b2$. Given $b2$, the best strategy for the learner population is $a2$, and so on. After four replacement steps, the populations are back to their initial state, and cycling has thus occurred.

Table 2. Example demonstrating cycling in a transitive problem. Left: payoff matrix. Right: After four transitions, both population are in their initial state again.

	b1	b2
a1	1	0
a2	0	1

L	a1	a1	a2	a2	a1
T	b1	b2	b2	b1	b1

1.5 Monotonic progress avoids cycling

Recently, an archive method for coevolution has been presented that guarantees monotonic progress for Pareto-Coevolution. The archive, called the Incremental Pareto-Coevolution Archive (IPCA) [16] maintains a learner archive and a test archive. The archive accepts learners and tests from a *generator*, which can be any coevolutionary algorithm. The generator can use the reliability of the archive by using individuals from the archive as a basis for generating new learners and test, and does not need to be reliable itself.

IPCA provides conditions determining which learners and tests produced by the generator are accepted into the archive. The learner archive accepts individuals that are non-dominated and different from existing individuals in the learner archive, and discards any individuals dominated by newly introduced individuals. The test archive accepts tests required to detect the uniqueness or value of learners, and grows incrementally. For a detailed description, please consult [16].

IPCA guarantees monotonic progress, i.e. the archive avoids regress. This criterion is defined as follows. The learner archives obtained over time form a series of approximations of the solution concept of the non-dominated front of learners relative to the set of all possible tests. Let these approximations be denoted as $L^1, L^2, \ldots L^t$, and let the test archives obtained over time be written as $T^1, T^2, \ldots T^t$. Then progress is monotonic if for any $t' > t$:

$$1.\ \forall TS \subseteq T^t : \quad [\exists L \in L^t : solves(L, TS) \implies \exists L' \in L^{t'} : solves(L', TS)]$$
$$2.\ \exists TS \subseteq T^{t'} : \quad [\nexists L \in L^t : solves(L, TS) \ \wedge\ \exists L' \in L^{t'} : solves(L', TS)]$$

From these definitions, it follows that any archive guaranteeing monotonic progress avoids cycling. This can be seen as follows. Since each test set in the sequence must contain a subset of tests that is not solved by a single learner in any previous learner set yet is solved by some learner in the current learner set, the learner set must therefore be different from any previous learner set. Therefore, cycling in the sense of a recurring population state is ruled out. In fact, the definition of monotonic progress is much stricter than the avoidance of cycling; it guarantees that actual improvements must be made between subsequent versions of the archive.

2 Experimental Results

As discussed in the previous section, cycling (whether due to intransitivity or not) can be avoided. This suggests that intransitivity should present no problem to coevolutionary algorithms with a progress guarantee. To test this conclusion, we apply the IPCA algorithm to two intransitive problems.

The first problem is the Intransitive Numbers Game introduced in [9]. We employ the version of the problem defined in [15], and use a value of 0.05 for the ϵ parameter. The problem is discretized with a granularity of 0.05. Mutation randomly adds a constant from [-.065,.035], and when used is applied to randomly selected dimensions twice. This mutation bias makes it more likely that mutation results in regress than progress, and is meant to model the situation in problems of practical interest where this is generally the case.

The generator that supplies candidate learners and tests to IPCA produces offspring using crossover (50%) and mutation (50%). With probability 0.1, it uses an archive member as a parent. For IPCA, archive members are chosen for this purpose based on recency, where $x = \frac{index+1}{archivesize}$ is used as the relative probability of an archive member in a Boltzmann distribution with temperature 1, i.e. $P_{rel} = e^x$. The generator maintains a learner and test populations, both of size 10. The objectives for learners are their outcomes against tests and the distinctions between tests. The learner objectives are based on the union of the current population and new generation of tests. The objectives for the tests are analogous, namely their outcomes against and distinctions between individuals in the current population and new generation of learners, resulting in a symmetric setup.

For each objective achieved by an individual, a score is assigned that equals one over the number of other individuals that achieve the objective, as in *competitive fitness sharing* [14]. The weighted sum of an individual's scores on the n outcome objectives (weight 0.75 for learners and 0.25 for tests) and on the n^2 distinction objectives, where n is the size of the population plus the new generation, are added to yield a single total score for the individual.

The highest scoring individuals of the new generation are lined up with the lowest scoring individuals of the current population. Then k is determined as the highest number for which the summed scores of the first k generation members is still at least as high as that of the first k population members. The lowest scoring k population members are discarded and replaced by the k highest scoring individuals from the new generation.

As a control experiment, we also measure the performance of the generator by itself, without the use of the IPCA archive. In addition to these two methods, the LAPCA method is used. This is a layered variant of IPCA design to achieve reasonable (but not guaranteed) reliability while maintaining limited archive sizes. LAPCA is described in detail elsewhere [17]. Briefly, the learner archive in LAPCA maintains a number n of layers of learners, analogous to the NSGA algorithm for EMOO [20], while the test archive maintains tests that *separate* these layers from one another; for each distinction [7] between learners in the same or subsequent layers, tests are kept that maintain the distinction.

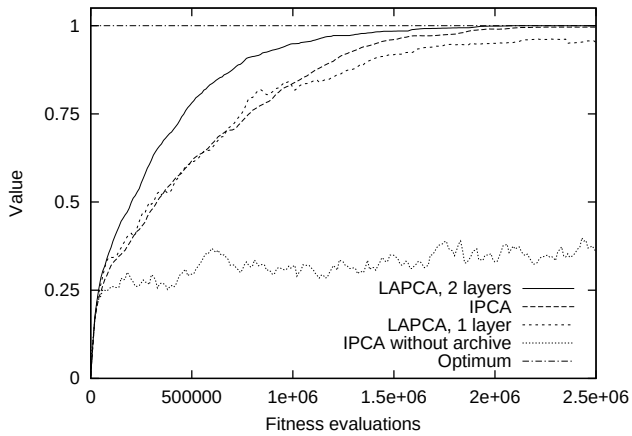

Fig. 1. Performance on the Intransitive Numbers game. IPCA and LAPCA converge to the optimum of 1, and LAPCA with two layers is most efficient in doing so. The archive methods perform substantially better than the method that does not use an archive.

The performance criterion is the lowest value among all dimensions of an individual; if this value increases, progress is made on all dimensions. Performance is plotted as a function of the number of actual generations, and averaged over 50 runs.

Figure 1 shows the performance of the different methods on the two-dimensional version of the Intransitive Numbers Game with mutation bias. The results validate the expectation that monotonic progress can be made.

The second test problem is the LINT game, introduced by Richard Watson [21]. LINT stands for Locally INTransitive game, and can be defined as follows.

$$G(a,b) = \begin{cases} a < b & \text{if} \quad |a-b| < d \\ a > b & \text{otherwise} \end{cases} \tag{9}$$

Figure 2 shows the results of the methods on the LINT problem. Again, all archive methods achieve sustained progress, and the two-layered LAPCA is most efficient in doing so.

3 Conclusions

Intransitivity has long been seen as a substantial obstacle to progress in coevolution, as it can lead to cycling. Recent developments in coevolution clarify how reliable progress can be guaranteed. Informed by these developments, we review and investigate the remaining significance of intransitivity and cycling for coevolution research.

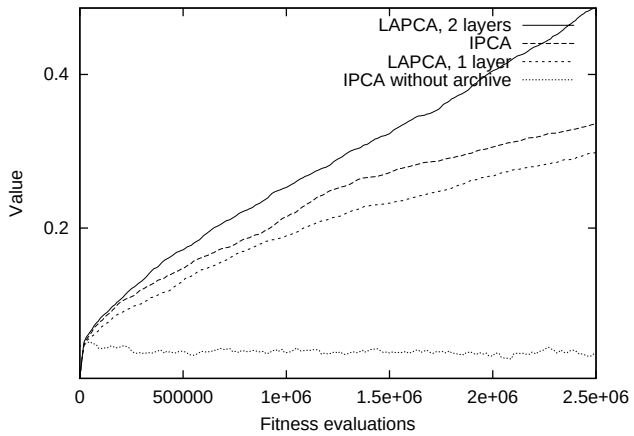

Fig. 2. Performance of the same methods on the LINT game. Again, all archive methods achieve sustained progress.

Pareto-Coevolution transforms intransitive problems into transitive problems, as known and as shown with a concise proof. Apart from intransitivity, the loss of informative tests can also lead to cycling. Several recent coevolution archives provide a guarantee of permitting progress only. Since cycling is thereby avoided, it is expected that intransitivity no longer presents an insurmountable obstacle to coevolution methods. This expectation is confirmed in experiments with two intransitive problems, where the IPCA and LAPCA archive methods achieved sustained progress and the same method without an archive fails to do so. A main further question regarding reliable algorithms for coevolution is how methods can be made more efficient while retaining reliability.

References

1. Barricelli, N.A.: Numerical testing of evolution theories. Part I: Theoretical introduction and basic tests. Acta Biotheoretica **16** (1962) 69–98
2. Axelrod, R.: The evolution of strategies in the iterated prisoner's dilemma. In Davis, L., ed.: Genetic Algorithms and Simulated Annealing. Research Notes in Artificial Intelligence, London, Pitman Publishing (1987) 32–41
3. Hillis, D.W.: Co-evolving parasites improve simulated evolution in an optimization procedure. Physica D **42** (1990) 228–234
4. Paredis, J.: Coevolutionary computation. Artificial Life **2** (1996)
5. Pagie, L., Hogeweg, P.: Evolutionary consequences of coevolving targets. Evolutionary Computation **5** (1998) 401–418
6. Juillé, H.: Methods for Statistical Inference: Extending the Evolutionary Computation Paradigm. PhD thesis, Brandeis University (1999)
7. Ficici, S.G., Pollack, J.B.: Pareto optimality in coevolutionary learning. In Kelemen, J., ed.: Sixth European Conference on Artificial Life, Berlin, Springer (2001)

8. Cliff, D., Miller, G.F.: Tracking the Red Queen: Measurements of adaptive progress in co-evolutionary simulations. In Morán, F., Moreno, A., Merelo, J.J., Chacón, P., eds.: Proceedings of the Third European Conference on Artificial Life: Advances in Artificial Life. Volume 929 of LNAI., Berlin, Springer (1995) 200–218
9. Watson, R.A., Pollack, J.B.: Coevolutionary dynamics in a minimal substrate. In Spector, L., Goodman, E., Wu, A., Langdon, W., Voigt, H.M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M., Burke, E., eds.: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-01, San Francisco, CA, Morgan Kaufmann (2001) 702–709
10. Nolfi, S., Floreano, D.: Co-evolving predator and prey robots: Do 'arms races' arise in artificial evolution? Artificial Life **4** (1998)
11. De Jong, E.D., Pollack, J.B.: Ideal evaluation from coevolution. Evolutionary Computation **12** (2004) 159–192
12. Ficici, S.G., Pollack, J.B.: A game-theoretic approach to the simple coevolutionary algorithm. In Schoenauer et al., M., ed.: Parallel Problem Solving from Nature, PPSN-VI. Volume 1917 of LNCS., Berlin, Springer (2000)
13. Watson, R.A., Pollack, J.B.: Symbiotic combination as an alternative to sexual recombination in genetic algorithms. In Schoenauer et al., M., ed.: Parallel Problem Solving from Nature, PPSN-VI. Volume 1917 of LNCS., Berlin, Springer (2000)
14. Rosin, C.D.: Coevolutionary Search among Adversaries. PhD thesis, University of California, San Diego, CA (1997)
15. Ficici, S.G., Pollack, J.B.: A game-theoretic memory mechanism for coevolution. In Cantú-Paz et al., E., ed.: Genetic and Evolutionary Computation – GECCO-2003. Volume 2723 of LNCS., Chicago, Springer-Verlag (2003) 286–297
16. De Jong, E.D.: The Incremental Pareto-Coevolution Archive. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-04. (2004)
17. De Jong, E.D.: Towards a bounded Pareto-Coevolution archive. In: Proceedings of the Congress on Evolutionary Computation, CEC-04. (2004)
18. De Jong, E.D., Pollack, J.B.: Principled Evaluation in Coevolution. Technical Report CS-02-225; May 31 2002, Brandeis University (2002)
19. Bucci, A., Pollack, J.B.: Focusing versus intransitivity. Geometrical aspects of coevolution. In Cantú-Paz et al., E., ed.: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-03, Berlin, Springer (2003)
20. Srinivas, N., Deb, K.: Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation **2** (1994) 221–248
21. Watson, R.A. Personal communication (2003)