Online Discovery of Search Objectives
for Test-based Problems

Paweł Liskowski and Krzysztof Krawiec
Institute of Computing Science
Poznan University of Technology, 60965 Poznań, Poland
{pliskowski,krawiec}@cs.put.poznan.pl

ABSTRACT

In [13], we proposed DISCO, a method that automatically identifies the groups of tests for which the candidate solutions behave similarly. Each such group gives rise to a *derived objective*, which together guide the search algorithm in multi-objective fashion. When applied to several well-known test-based problems, the proposed approach significantly outperforms the conventional two-population coevolution. This opens the door to efficient and generic countermeasures to premature convergence for both coevolutionary and evolutionary algorithms applied to problems featuring aggregating fitness functions.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

Keywords

coevolution; test-based problems; multi-objective evolutionary computation; search driver

1. INTRODUCTION

Many optimization and learning problems approached in evolutionary computation involve objective functions that reward candidate solutions by counting the number of *tests* they pass. When evolving computer programs or controllers, passing a test requires producing the desired output for a given input. When learning game strategies, tests are embodied by opponents, and a candidate solution passes a test if it wins a game against it. In these problems, known in the context of coevolutionary algorithms [1, 6] as *test-based problems* [3, 4, 7], candidate solutions need to *interact* with multiple 'environments' in order to be evaluated.

Solving a test-based problem consists in finding a candidate solution with certain properties. In the most common case, it should maximize the *expected utility*, i.e., the average

outcome against all tests. Finding such a solution is challenging in many test-based problems, because the number of tests is usually large, and for some problems even infinite.

The algorithms that naturally match the class of test-based problems are two-population coevolutionary algorithms [1, 6]. A typical coevolutionary algorithm maintains a population of candidate solutions $S \subset \mathcal{S}$ and a separate population of tests $T \subset \mathcal{T}$. In every generation, each candidate solution $s \in S$ interacts with every test $t \in T$, producing an interaction outcome $g(s, t)$. The interaction outcomes are gathered in an *interaction matrix* G, from which the fitness values of individuals in S and T are calculated.

An objective function that counts the number of passed tests such as expected utility usually forms an inherent part of the problem and makes it amenable to many conventional algorithms that expect a scalar objective. However, aggregation of interaction outcomes inevitably leads to information loss, primarily due to *compensation*: two solutions that pass k tests each are considered equally valuable, no matter *which* particular tests they pass. Also, aggregation neglects the fact that some tests can be inherently more difficult than others, or more or less harder to reach for a given search algorithm. Algorithms that rely on aggregation are oblivious to these aspects and thus prone to inferior performance.

Compensation can be avoided by comparing the candidate solutions using the *dominance relation.* Dominance compares the behavior of solutions on particular tests and is in this sense more scrupulous than the expected utility. However, by being a partial relation, dominance fails to provide a useful search gradient whenever none of the compared solutions passes a superset of the tests solved by the other solution (cf. [8]). This is unfortunately common: for two unrelated solutions, it is much more likely that they are mutually non-dominated than that one of them dominates the other, and that likelihood grows with the number of tests.

Expected utility and dominance occupy thus two extremes in scrutinizing interaction outcomes. In [13], we proposed DISCO, a method that is a compromise between these extremes. We motivate DISCO's design by the following observations. First, tests are often characterized by different *difficulty*, which is not known a priori. Coming across a problem instance with all tests equally difficult is much less likely than difficulty varying across the tests. Second, any stochastic search algorithm (other than a purely random search) has a *search bias*, i.e., it is more likely to visit some candidate solutions than others. As a consequence of diverse test difficulty and search bias, a search algorithm driven by a scalar evaluation measure tends to converge to

candidate solutions that solve the tests that are easier and better 'reachable'. In parallel search techniques like EAs or coevolutionary algorithms, the probable aftermath of that is premature convergence.

DISCO mitigates this problem by deriving alternative search objectives via identifying the combinations of interaction outcomes that prevail in an interaction matrix. The derived objectives form a multi-objective characterization of the candidate solutions in the context of the current population of tests (cf. [12]) , and are subsequently fed into a multi-objective selection method NSGA-II [5].

DISCO broadens thus the bottleneck of evaluation in characterizing the candidate solutions with k objectives rather than with a single one. On the other hand, keeping k small ensures that the dominance relation in the derived space is dense enough to provide a reasonably strong search gradient (as opposed to the dominance on all tests). Candidate solutions that feature different 'skills' can coexist in a population even if some of them are clearly better than others in terms of scalar evaluation.

In [13], we conducted scrupulous experimental analysis on the suite of diversified benchmarks consisting of Iterated Prisoner's Dilemma [2], Numbers Games [14] and Density Classification Task, and demonstrated that DISCO is able to identify meaningful derived objectives that are often internally cohesive and mutually non-redundant. The method autonomously adjusts the number of derived objectives to the problem characteristics and the dynamics of evolutionary search, and systematically improves the performance in comparison to the conventional coevolutionary algorithm driven by the scalar evaluation.

The heuristic character of DISCO is advantageous in several respects. Firstly, it entails only moderate computational overhead. Secondly, the derived objectives evolve along the candidate solutions and may adapt to their capabilities, creating a suitable search gradient while avoiding over-specialization.

DISCO offers means to widen the *evaluation bottleneck* between the fitness function and a search algorithm. As we postulated in [11], treating fitness function as a black box is unjustified, especially when more detailed information on solution's characteristics, like interaction outcomes, is easily available. Such information may deserve more effort in conceptual analysis, implementation and computational expense to harness it, but, as we showed in this study, these costs may pay off with a more effective search method.

In replacing the original objective function with heuristic and transient derived objectives, DISCO subscribes to the concept of *search driver* [10, 9]. A single derived objective is a search driver in the sense that it conveys only partial information about the quality of candidate solution. In a rugged and multimodal fitness landscape, the original objective may turn out to be more deceptive than an imperfect search driver. This becomes particularly true in DISCO, where multiple diversified search drivers are used simultaneously and so mitigate premature convergence.

2. REFERENCES

[1] R. Axelrod. The evolution of strategies in the iterated prisoner's dilemma. *The dynamics of norms*, pages 1–16, 1987.

[2] R. Axelrod and W. D. Hamilton. The evolution of cooperation. *Science*, 211(4489):1390–1396, 1981.

[3] A. Bucci, J. B. Pollack, and E. de Jong. Automated extraction of problem structure. In K. D. et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 3102 of *GECCO 2004*, pages 501–512, Seattle, WA, USA, 26-30 June 2004. Springer-Verlag.

[4] E. D. de Jong and J. B. Pollack. Ideal Evaluation from Coevolution. *Evolutionary Computation*, 12(2):159–192, Summer 2004.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182 –197, apr 2002.

[6] W. D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial life II*, volume 10 of *Sante Fe Institute Studies in the Sciences of Complexity*, pages 313–324, Redwood City, Calif., 1992. Addison-Wesley.

[7] W. Jaśkowski. *Algorithms for Test-Based Problems*. PhD thesis, Institute of Computing Science, Poznan University of Technology, Poznan, Poland, 2011.

[8] K. Krawiec. Genetic programming-based construction of features for machine learning and knowledge discovery tasks. *Genetic Programming and Evolvable Machines*, 3(4):329–343, 2002.

[9] K. Krawiec. *Behavioral Program Synthesis with Genetic Programming*, volume 618 of *Studies in Computational Intelligence*. Springer International Publishing, 2015.

[10] K. Krawiec and U.-M. O'Reilly. Behavioral programming: a broader and more detailed take on semantic GP. In C. Igel, editor, *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO 2014, pages 935–942, Vancouver, BC, Canada, 12-16 July 2014. ACM. Best paper.

[11] K. Krawiec and J. Swan. Pattern-guided genetic programming. In C. Blum, editor, *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO 2013, pages 949–956, Amsterdam, The Netherlands, 6-10 July 2013. ACM.

[12] P. Liskowski and K. Krawiec. Discovery of implicit objectives by compression of interaction matrix in test-based problems. In *Parallel Problem Solving from Nature–PPSN XIII*, pages 611–620. Springer, 2014.

[13] P. Liskowski and K. Krawiec. Online discovery of search objectives for test-based problems. *Evolutionary computation*, 2016.

[14] R. A. Watson and J. B. Pollack. Coevolutionary dynamics in a minimal substrate. In L. S. et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO 2001, pages 702–709, San Francisco, California, USA, 7-11 July 2001. Morgan Kaufmann.