Co-Optimization Algorithms

Travis C. Service
Missouri University of Science and Technology
325 Computer Science Building
500 West 15th Street
Rolla, Missouri 65409-0350, USA
tservice@acm.org

Daniel R. Tauritz
Missouri University of Science and Technology
324 Computer Science Building
500 West 15th Street
Rolla, Missouri 65409-0350, USA
dtauritz@acm.org

ABSTRACT

While coevolution has many parallels to natural evolution, methods other than those based on evolutionary principles may be used in the interactive fitness setting. In this paper we present a generalization of coevolution to co-optimization which allows arbitrary black-box function optimization techniques to be used in a coevolutionary like manner.

We find that the co-optimization versions of gradient ascent and simulated annealing are capable of outperforming the canonical coevolutionary algorithm. We also hypothesize that techniques which employ non-population based selection mechanisms are less sensitive to disengagement.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization

General Terms

Algorithms

Keywords

Coevolution, Simulated Annealing, Gradient Ascent

1. INTRODUCTION

Coevolution has shown promise in a variety of domains [3]. However, there are several problems inherent to coevolution described in literature, including: over-specialization [1], disengagement [1], inaccurate evaluation [2] and cycling [1].

Many of the problems observed in coevolutionary algorithms (CoEAs) are interrelated (i.e., over-specialization can result from inaccurate evaluation). While many additions to CoEAs have been proposed to address one or more of these deficiencies [5], the same underlying evolutionary model has been the base for all such additions.

In this paper we present a generalization of coevolution to co-optimization, where arbitrary black-box optimization algorithms may be employed in the interactive fitness setting. We compare the performance of the co-optimization versions of gradient ascent and simulated annealing to the canonical CoEA, under nominal conditions, as well as under the effects of disengagement. The results we present suggest that non-population based techniques may be less susceptible to disengagement than population based techniques.

2. CO-OPTIMIZATION

While coevolutionary processes have parallels to that of natural evolution, techniques other than evolutionary computation may be employed in the interactive fitness setting. Any method of generating new individuals to add to the current populations may be used in place of artificial evolution.

Rather than introducing a new add-on to coevolutionary algorithms to help address a deficiency observed on a particular class of problems, we instead suggest matching each problem with the technique best suited for it.

We propose generalizing coevolution to co-optimization, where arbitrary black-box optimization techniques may be used in place of artificial evolution. The basic principle of interactive fitness is still used in this larger class of algorithms. Individuals are judged by interactions with their peers. The method used to coevolve backgammon players in [4] is an example of this generalization of coevolution. The template algorithm presented in [2] can also be seen as an example of co-optimization where the generate and select functions may be filled in with arbitrary techniques.

This generalization allows for matching interactive problem domains to the optimization algorithms best suited for them. If a domain lends itself to a particular technique, such as simulated annealing, the co-optimization version of simulated annealing could be used in place of a CoEA.

A nice feature of the generalization of coevolution to co-optimization is that the majority of the growing body of theoretical results for coevolution and the numerous additions to the canonical CoEA also apply to the larger class of co-optimization algorithms, as the majority of such techniques are not dependent on an underlying evolutionary model.

3. TEST PROBLEMS & RESULTS

We compare the performance of the co-optimization versions of gradient ascent (CoGA) and simulated annealing (CoSA) to a canonical CoEA, both under nominal conditions as well as under disengagement.

In both test problems, two populations, consisting of 40 individuals each, are employed, one for test cases and one for candidate solutions. One-point crossover is used for recombination and mutation is always applied in the form of the

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

addition of a uniform random number in the interval $[-3, 3]$ to a randomly selected allele. Tournament selection is used for parent and survival selection. 40 offspring are created each generation for each population. For CoGA, a single child is generated for every individual by mutation and is selected to replace its parent if it is superior in the current context. For CoSA, children are successively generated for each individual by mutation until a generated individual is accepted, according to a linear cooling schedule.

3.1 Compare-On-One

The Compare-On-One problem is an example of a numbers game problem [2]. Candidate solutions and test cases are vectors of real numbers. A candidate solution's fitness is equal to the number of test cases which it solves. A candidate solution solves a test case if the candidate solution's value on the dimension on which the test case is greatest is greater than that of the test case. Thus, each test case promotes advances along a single dimension.

Formally, a candidate solution, C, passes a test, T, if

$$C_{T_{MAX}} \geq T_{T_{MAX}}$$

For the Compare-On-One problem both the candidate solution and test case populations are initialized such that each member is a uniform random number in the interval $[0, 10]$.

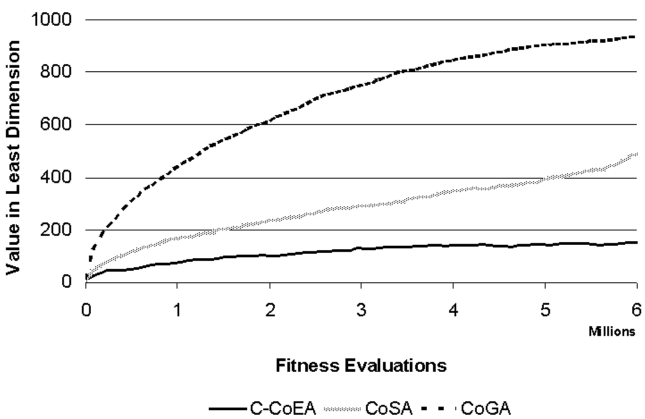

Figure 1: Performance of CoEA, CoSA and CoGA on the Compare-On-One problem.

Figure 1 shows the performance of each algorithm averaged over 60 runs. Of the three test algorithms, CoGA performs the best followed by CoSA and finally by the CoEA.

3.2 Compare-On-All

Compare-On-All is identical to Compare-On-One, except that tests promote advances on all dimensions [2].

Formally, a candidate solution, C, passes a test, T, if

$$\forall i \in \{1, 2, \cdots, n\} \ \ C_i \geq T_i$$

The Compare-On-All problem is used to investigate the effects of disengagement on the three algorithms. The members of the test case population are initialized to be between 190 and 200 and the members of the candidate solution population are initialized to be between 50 and 60. Figure 2 shows the performance of all three algorithms. Both CoSA and CoGA were able to reengage and make significant progress, while CoEA was stuck in a period of random drift for the majority of the runs.

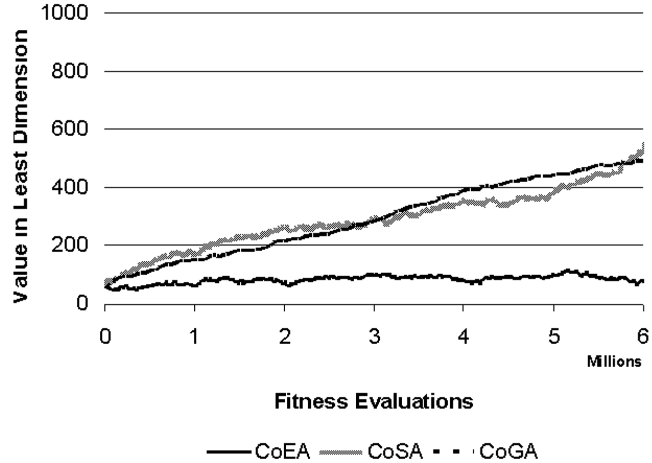

Figure 2: Performance of CoEA, CoSA and CoGA on the disengaged Compare-On-All problem.

4. CONCLUSIONS & FUTURE WORK

The primary contribution of this work is the introduction of the class of co-optimization algorithms. This class provides a natural generalization of coevolution to arbitrary optimization techniques and employs the same interactive fitness environment as does coevolution.

We compare the performance of CoSA, CoGA and CoEA on two test problems under both nominal conditions as well as conditions of disengagment. We find that in both instances CoSA and CoGA outperformed CoEA. Furthermore, under conditions of disengagement, CoSA and CoGA were able to effectively reengage and make significant progress, whereas CoEA remained in a state of random drift for the entirety of most runs. These results promote the hypothesis that non-population based selection methods are less susceptible to the effects of disengagement.

To strengthen the results presented in this paper, the performance of the co-optimization algorithms described in this paper need to the compared to the many sophisticated extensions of CoEAs in existence, such as competitive fitness sharing [5] and reduced parasite virulence [1].

5. REFERENCES

[1] John Peter Cartlidge. *Rules of Engagement: Competitive Coevolutionary Dynamics in Computational Systems*. PhD thesis, University of Leeds, 2004.

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

[3] Daniel Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D Nonlinear Phenomena*, 42:228–234, June 1990.

[4] Jordan Pollack and Alan Blair. Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, 32(3):225–240, 1998.

[5] Christopher Darrell Rosin. *Coevolutionary Search Among Adversaries*. PhD thesis, University of California - San Diego, 1997.