

Large Scale Empirical Analysis of Cooperative Coevolution

Sean Luke

Department of Computer Science
George Mason University
sean@cs.gmu.edu

Keith Sullivan

Department of Computer Science
George Mason University
ksulliv1@cs.gmu.edu

Faisal Abidi

Department of Computer Science
George Mason University
fabidi@gmu.edu

ABSTRACT

We present a study of cooperative coevolution applied to moderately complex optimization problems in large-population environments. The study asks three questions. First: what collaboration methods perform best, and when? Second: how many subpopulations are desirable? Third: is it worthwhile to do more than one trial per fitness evaluation? We discovered that parallel methods tended to work better than sequential ones, that “shuffling” (a collaboration method) predominated in performance in more complex problems, that more subpopulations generally did better, and that more trials performed marginally better.

Categories and Subject Descriptors

I.2.m [Artificial Intelligence]: Miscellaneous—*Evolutionary Computation*

General Terms

Algorithms, Experimentation

Keywords

Coevolution

1. INTRODUCTION

Cooperative coevolution is an evolutionary algorithm framework which breaks a population into several subpopulations, each optimizing a sub-part of the complete solution. Individuals are tested in *trials* by grouping them with *collaborators* from each of the other subpopulations to form a complete candidate solution, which is then assessed. The fitness of the individual is then *evaluated*, often involving the maximum over its trials. The promise of cooperative coevolution is that it can take advantage of problems which are partially decomposable, that is, breakable into separate subproblems which can be somewhat, though not entirely, optimized independently due to low linkage between variables across subproblem boundaries.

Cooperative coevolution entails many parameters beyond those common to evolutionary computation. This study focuses on four critical ones: (1) What *update timing* should be applied? In *sequential* update timing, only one subpopulation is assessed and bred per generation; in *parallel* update

Problem	Description	Bounds for x_i
Sum	$\sum_{i=1}^n x_i$	$[-5.12, 5.12]$
Median	$\text{median}(x_1, \dots, x_n)$	$[-5.12, 5.12]$
Min	$\min(x_1, \dots, x_n)$	$[-5.12, 5.12]$
Rastrigin*	$10n \sum_{i=1}^n x_i^2 - 10 \cos(2\pi x_i)$	$[-5.12, 5.12]$
Schwefel*	$\sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	$[-512.03, 511.97]$
Rosenbrock*	$\sum_{i=1}^{n-1} (1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2$	$[-2.048, 2.048]$
Rot. Rastrigin*	$\text{Rastrigin}(\text{rotate}(x_1, \dots, x_n))$	$[-5.12, 5.12]$
Rot. Schwefel*	$\text{Schwefel}(\text{rotate}(x_1, \dots, x_n))$	$[-512.03, 511.97]$

Table 1: Test problems. * indicates minimization.

timing, subpopulations are assessed and bred simultaneously. (2) What *collaboration scheme* should be used? This is the procedure by which collaborators are gathered together to perform trials. (3) How many subpopulations should be used, assuming the number is not constrained by the problem? (4) Is it worthwhile to base fitness on more than a single trial per individual? We test these parameters against several problems of varying complexity, all with a moderately large genome size (100), with large numbers of trials (one million per run), and with large populations (1000).

This paper is a summary of a fuller technical report [1].

2. EXPERIMENTS

We used a large population: $p = 1000$ individuals, divided into $s = 1, 2, 4, 10, 50,$ and 100 subpopulations to perform coevolution. All problems had $n = 100$ variables. Individuals in each subpopulation thus had genome lengths of n/s .

We tested using both parallel and sequential update timing. Parallel update timing runs lasted for $g = 1000$ generations, except in “shuffling” as discussed next. Sequential update timing runs lasted for $g \times s$ generations. We compared eight collaboration schemes, based on the following three criteria:

- Individuals were tested with the *single best* collaborators of the prior generation from other subpopulations.
- Individuals were tested with collaborators chosen at *random*, with replacement, from the current individuals of other subpopulations.
- Individuals in each subpopulation were randomly *shuffled*, then paired off. If the update timing was parallel, one shuffling would result in p/s total trials: the surplus in trials could be redistributed using *larger subpopulation sizes* (p individuals each), *more generations* ($g \times s$ generations), or *more shuffled trials* (s per individual).

This yielded eight combinations of update timing and collaboration: Sequential Single Best (SB), Sequential Random (SR), Sequential Shuffled (SS), Parallel Single Best (PB),

Problem	Number of Subpopulations										
	2	4	10	50	100						
Sum	SB PB	SB PB	PB SB	SB PB	PP PG	PB SB	PP PG				
Median	PB SB	PP	SB PB	SB PB	(all)	(all)					
Min	PP PG	PG	PP	PG	PG						
Rosenbrock	PP	PP	PP	PP	PP						
Rastrigin	PP	PP	PP	PP	PP						
Schwefel	PP	PP	PP	PP	PP						
Rot. Rastrigin	PP	PP	PP	PP	PP						
Rot. Schwefel	PT PR	PG*	PG	PG	PG						

Table 2: First Experiment. Shown are the best-performing schemes without statistically significant differences. Note * where PR and SB are not significantly different; and also PG, SB, PP, and PB.

Problem	Collaboration Scheme	Subpopulations
Sum	Sequential Single Best	2 4 10 50 100
Median	Sequential Single Best	2 4 10 50 100
Min	Parallel Shuffled Gens	2 4 100 50 10
Rosenbrock	Parallel Shuffled Pops	2 4 10 50 100
Rastrigin	Parallel Shuffled Pops	2 4 10 50 100
Schwefel	Parallel Shuffled Pops	2 4 10 50 100
Rot. Rastrigin	Parallel Shuffled Pops	2 4 10 50 100
Rot. Schwefel	Parallel Shuffled Gens	2 4 10 100 50

Table 3: Second Experiment. Subpopulations are ordered in increasing performance. Overbars indicate statistically insignificant differences.

Parallel Random (PR), Parallel Shuffled Pops (PP), Parallel Shuffled Gens (PG), and Parallel Shuffled Trials (PT).

We tested against the eight test problems in Table 1. Some problems were rotated, using a random orthonormal basis.

Breeding used tournament selection (of size 2), then one-point crossover, then gaussian mutation ($\sigma = 0.01$), reselecting mutation until the new gene was within valid gene boundaries. We ran each experiment for 100 independent runs, and used one-way ANOVAs each set to $p = 0.00025$ in order to retain slightly better than $p = 0.05$ for the paper as a whole. Experiments were performed using the ECJ library.

First Experiment: Collaboration Scheme Choice. We began by comparing the eight collaboration schemes in the context of all combinations of problem type and number of subpopulations (including 1: a standard GA). We asked: which collaboration schemes worked best and in which situations? Abridged results are summarized in Table 2. We had hypothesized that single-best strategies would work well for simple, linearly separable problems, but less well for others. In fact this is what occurred: Parallel Single Best and Sequential Single Best had the best results for the two easiest problems (Sum and Median), but did not impress elsewhere.

Parallel Shuffled Pops and Parallel Shuffled Gens dominated the other problems. The one exception was Schwefel at 2 subpopulations, where Parallel Shuffled Trials and Parallel Random performed best. In nearly all problems, the baseline (1-population) GA was middling. Sequential Shuffled and Sequential Random were poor performers. Never did sequential schemes statistically significantly outperform parallel ones.

Second Experiment: Number of Subpopulations. We then compared across different numbers of subpopulations. We hypothesized that as the number of subpopulations grew, cooperative coevolution would perform well on separable problems but not high-linkage ones (such as rotated ones).

Problem	Collaboration Scheme	Number of Subpopulations				
		2	4	10	50	100
Sum	PB	PB	PB	PB	PB	PB
Median	PB	PB	PB	PB	PB	PB 2G
Min	PG	PG	2P PG	2P PG	2G	2G
Rosenbrock	PP	PP 2G	2G PP	2G	2G	2G
Rastrigin	PP	PP 2G	2G PP	2G	2G	2G
Schwefel	PP	PP	PP	PP	PP 2G	PP 2G
Rot. Rastrigin	PP	2G PP	2G	2G	2G	2G
Rot. Schwefel	PG	2G 2P*	2P PG	2P PG	PG 2P 2G	2P 2G PG

Table 4: Third Experiment. Shown are the best-performing schemes without statistically significant differences. Note * where 2P and PG are not significantly different.

The results were very surprising. As the number of subpopulations increased, the performance order of the collaboration schemes stayed roughly the same, but the variance increased. The poorer-performing collaboration schemes generally performed *much worse*, while the highest-performing collaboration schemes consistently performed *even better*. Table 3 summarizes the results using the highest-performing collaboration schemes, illustrating part of this phenomenon.

Third Experiment: Extra Trials. The prevailing wisdom regarding cooperative coevolution is that it may be worthwhile to perform multiple trials to compute a fitness evaluation, because individuals are evaluated in the context of other individuals who may or may not prove to be worthwhile collaborators. To test this prevailing wisdom, we selected the best collaboration method for each problem. These methods originally used one trial. We then built two-trial variations of the methods and compared them to the originals. For each variation, we tried both halving the subpopulation size and halving the total number of generations. In Table 4 these are referred to as the “2P” and “2G” subvariations.

For Sum and Median, we selected Parallel Single Best, and compared it against a variation where an individual is tested against the single best collaborators and also against randomly-chosen collaborators (the so-called “CCEA-2” method). For Min and Rotated Schwefel, we chose Parallel Shuffled Gens, and compared it against a variation where two randomly shuffled trials were performed. For the remaining problems we chose Parallel Shuffled Pops, again compared against a variation with two randomly shuffled trials.

Table 4 shows the results. We expected that for easily separable problems (such as Sum or Median) more trials would provide no benefit: and this was the case. Interestingly, for Schwefel and Rotated Schwefel more trials weren’t useful either. For the other problems, it was useful to have more trials with more subpopulations. This made sense, as more subpopulations meant more collaborators to add contextual noise to the fitness evaluation. Even so, the improvements were rarely large. It was always as good or better to halve the generations rather than the population size.

3. ACKNOWLEDGMENT

This research was supported by NSF Grant 0916870.

4. REFERENCES

- [1] S. Luke, K. Sullivan, and F. Abidi. Large scale empirical analysis of cooperative coevolution. Technical Report GMU-CS-TR-2011-2, Dept. Computer Science, George Mason University, 2011. Available at <http://cs.gmu.edu>.