The Effects of Training Set Size and Keeping Rules on the Emergent Selection Pressure of Learnable Evolution Model

Mark Coletti*
Computer Science Department
George Mason University
4400 University Drive
Fairfax, VA 22030
mcoletti@gmu.edu

ABSTRACT

Evolutionary algorithms with computationally expensive fitness evaluations typically have smaller evaluation budgets and population sizes. However, smaller populations and fewer evaluations mean that the problem space may not be effectively explored. An evolutionary algorithm may be combined with a machine learner to compensate for these smaller populations and evaluations to increase the likelihood of finding viable solutions. Learnable Evolution Model (LEM) is such an evolutionary algorithm (EA) and machine learner (ML) hybrid that infers rules from best- and least-fit individuals and then exploits these rules when creating offspring. This paper shows that LEM introduces a unique form of emergent selection pressure that is separate from any selection pressure induced by parent or survivor selection. Additionally this work shows that this selection pressure can be attenuated by how the best and least fit subsets are chosen, and by how long learned rules are kept. Practitioners need to be aware of this novel form of selection pressure and these means of adjusting it to ensure their LEM implementations are adequately tuned. That is, too much selection pressure may mean premature convergence to inferior solutions while insufficient selection pressure may mean no sufficient solutions are found.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Concept learning*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

performance

Keywords

evolutionary computation, machine learning, learnable evolution model, function optimization

*Also affiliated with the Evolutionary Computation Laboratory and the Krasnow Institute for Advanced Study Adaptive Systems Laboratory

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

1. INTRODUCTION

Some optimization problems to which evolutionary algorithms (EA) are applied are computationally burdensome with attendant lengthy fitness evaluations. For example, evaluating an individual may entail running an involved simulation or solving a complex high dimensional problem. This means that for an EA to be practically applied to such problems that the population sizes and number of evaluations must be accordingly small. Unfortunately, this also means that the problem space will not be adequately covered as the algorithm progresses such that viable solutions are missed. One approach for mitigating this problem is to augment an EA in some way with a Machine Learner (ML).

Algorithm 1 Simplified Learnable Evolution Model

$i \leftarrow 0$
$P_i \leftarrow initial()$ ▷ Create initial population
repeat
  $learn(best(P_i), worst(P_i))$ ▷ Learn from best and worst
  for $j = 0 \rightarrow \lambda - 1$ **do**
    $I \leftarrow applyRules(mutate(\sigma, \rho, clone(P_i[j])))$
    $P_{i+1}[j] \leftarrow I$
  end for
  $i \leftarrow i + 1$ ▷ Offspring become new parents
until $halt() = true$

Learnable Evolution Model (LEM) is just such an (EA) / (ML) hybrid [1]. Algorithm 1 shows a simplified version of LEM used for this paper. For each generation the machine learner infers rules describing good gene value ranges corresponding to the selected "best" individuals. Then each parent creates one offspring by cloning itself and perturbing the clone's genes with a Gaussian mutation operator with σ scale and ρ mutation rate; then this clone is passed to $applyRules()$, which further perturbs genes cited by a rule. That is, the learner may infer zero or more rules from the current "best" and "worst" parents; one of these rules is selected and applied to the offspring. The rules are conjunctions of the form $[x_l, x_u]_i$ where x_l is a lower bound for a given gene, x_u the upper, and i the referred gene. Each gene covered by a rule gets a new value from a uniform random distribution for the given range dictated by that rule's bounds. For example, the rule $[-1.08, 2.92]_1 \wedge [5.92, 6.13]_3$ would mean that offspring would have their first gene values generated in U(-1.08,2.92) and the third gene in U(5.92,6.13). In cases where the machine learner inferred more than one set of rules a single rule is arbitrarily selected and then similarly applied.

Parameter	Value
Parents	50
Brood size	1
Initial domain range	[-4,4]
Genome size	2
Mutation	Gaussian
Mutation rate (ρ)	0.5
Mutation scale (σ)	0.1
Runs	30
Fitness function	Spheroid
Machine learner	C4.5 [3]

Table 1: Run-time parameters for experiments

Selection pressure is the means by which an EA converges to a solution, and is articulated through the selection of parents to generate offspring and the selection of survivors. However, LEM may induce a novel kind of selection pressure that is entirely separate from that found in legacy EA operators — in a sense, a sort of selection pressure *emerges* from the interaction of selected training sets, the machine learner, and the exploited rules.

A "well tuned" EA balances the opposing forces of selection and perturbation; too much of one or the other will lead to suboptimal performance [2]. Practitioners may need to take any such additional selection pressure into consideration when implementing LEM. Failure to do so may lead to EAs that are "out of tune" and thus perform suboptimally, perhaps by converging too quickly on inferior optima. This research shows that this type of LEM-specific selection pressure is real and it responds to the size of the best and worst subpopulations used to train the ML, and to how long inferred rules are allowed to persist.

2. METHODOLOGY

The objective of the first set of experiments was to show that LEM does have inherent selection pressure. This was done by using a "control" EA where selection pressure has been removed by using deterministic parent selection and non-overlapping generations, and then comparing that with an almost identical EA that uses LEM. Essentially the difference between the two sets of experiments is that the former did not have *applyRules*() whereas the latter did. Hypothetically, if LEM has inherent selection pressure then the "control" should exhibit random walk behavior because there is no selection pressure, whereas the LEM runs should converge because it does. Table 1 shows run-time parameters used for all experiments.

The previous LEM experiments used the top 15 and bottom 15 of the population by rank for the "best" and "worst" ML training sets, respectively. To explore the influence of training set size on LEM's selection pressure, experiments were run for the top and bottommost 5, 10, and 25 individuals as machine learner "best" and "worst" training sets.

Another set of experiments were run that kept rules for up to three generations for training set sizes of 5, 15, and 25 individuals of the best and least out of the total population of 50 to gauge the impact of making rules persistent for multiple generations. As for the previous LEM runs, if there was more than one rule, a single rule was selected from the pool of rules with equal chance regardless of its age.

3. RESULTS AND CONCLUSIONS

This research shows that Learnable Evolution Model has an emergent and unique selection pressure. This was shown when LEM consistently converged to the global optima for a simple problem when compared to a similarly configured EA that does not converge due to not having selection pressure, as depicted in that scatterplots of all the individuals from all the runs in Fig.1. It was also demonstrated that this selection pressure responds to the machine learner training set size and to allowing rules to linger for more than a generation. Here the results were counterintuitive. That is, instead of demonstrating greedy behavior with smaller training sets — that is, biasing the system in favor of a smaller set of the fittest — LEM showed less greedy behavior by taking longer to converge and higher variance by generation.

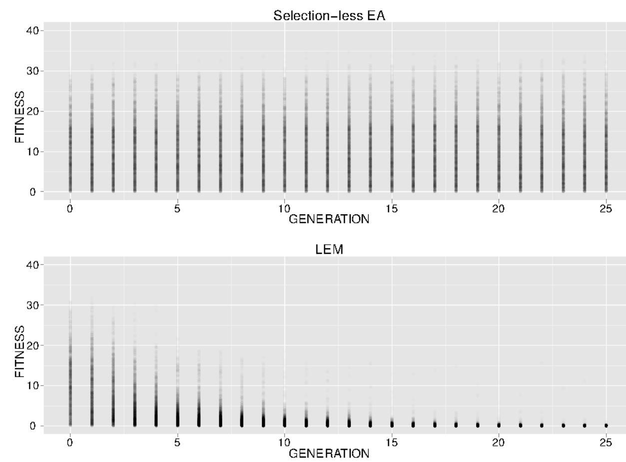

Figure 1: No selection pressure EA vs. LEM

More specifically, care must be taken to ensure that the training sets are not too small, else the machine learner is unable to infer rules and the system reverts to regular EA behavior. Smaller training sets can be somewhat compensated for by keeping rules for more than one generation so that even if the machine learner does not have enough information to learn anything, there may still be some rules that can be used to create offspring.

Keeping rules longer also had an affect on LEM selection pressure. As expected, the longer rules were allowed to stay, there was correspondingly less selection pressure.

4. REFERENCES

[1] R. S. Michalski. *Machine Learning*, volume 38, chapter LEARNABLE EVOLUTION MODEL: Evolutionary Processes Guided by Machine Learning, pages 9–40. Kluwer Academic Publishers, 2000.

[2] Kenneth De Jong. *Evolutionary Computation: A Unified Approach*. The MIT Press, The MIT Press, 55 Hayward St., Cambridge, MA 02142, 2006.

[3] J. R. Quinlan. *C4.5, Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.