Evolutionary Computation: An Investigation of Parameter Space

Moshe Sipper
Institute for Biomedical Informatics, University of Pennsylvania
Philadelphia, PA, USA 19104-6021
and Department of Computer Science, Ben-Gurion University, Israel
Beer Sheva, Israel 8410501
sipper@gmail.com

Weixuan Fu
Institute for Biomedical Informatics, University of Pennsylvania
Philadelphia, PA, USA 19104-6021
weixuanf@pennmedicine.upenn.edu

Karuna Ahuja
Institute for Biomedical Informatics, University of Pennsylvania
Philadelphia, PA, USA 19104-6021
karuna.ahuja1@gmail.com

Jason H. Moore
Institute for Biomedical Informatics, University of Pennsylvania
Philadelphia, PA, USA 19104-6021
jhmoore@upenn.edu

ABSTRACT

Through an *extensive* series of experiments over multiple evolutionary algorithm implementations and 25 problems we showed that parameter space tends to be rife with viable parameters, somewhat in contrast with common lore [6].

CCS CONCEPTS

•Mathematics of computing →Evolutionary algorithms; •Theory of computation →Evolutionary algorithms;

KEYWORDS

Evolutionary algorithms, Genetic programming, Meta-genetic algorithm, Parameter tuning, Hyper-parameter

ACM Reference format:
Moshe Sipper, Weixuan Fu, Karuna Ahuja, and Jason H. Moore. 2018. Evolutionary Computation:
An Investigation of Parameter Space. In *Proceedings of the Genetic and Evolutionary Computation Conference 2018, Kyoto, Japan, July 15–19, 2018 (GECCO '18 Companion),* 2 pages.
DOI: 10.1145/3205651.3208207

In [6] we examined key parameters in evolutionary computation (EC), asking whether we might find new insight into the ever-crucial parameter-seeking process. EC practitioners often employ commonly used parameters "selected by conventions, ad hoc choices, and very limited experimental comparisons" [1] (see also [2]). We sought parameters that met a reasonable minimal performance level over an entire set of several problems. We experimented with a large and variegated assortment of problems in what was arguably one of the most extensive EC experiments, concluding that parameter space, in fact, tends to be rife with viable parameters. This does not mean that the EC practitioner's job is over, given the many desiderata that still remain, including the representation of solutions in the search space, defining the fitness function, designing crossover and mutation operators, and more. However, our experiments suggested that one can at least find good parameters with a bit more ease.

We chose to work with two very different evolutionary-algorithm packages: Distributed Evolutionary Algorithms in Python (DEAP) [3]—which uses tree-based GP, and M4GP [4]—which is a stack-based evolutionary algorithm. We ran our experiments on a cluster of 224 cores (Intel® Xeon® E5-2650L), with 2 threads per core.

DEAP, available at `github.com/DEAP`, comes with five sample problems: Symbolic regression, Even-Parity, Multiplexer 3-8, Artificial Ant, and Spambase. M4GP is entirely different, based on stack-based GP. This served to reinforce our conclusions by running our experiments on two very different types of EC algorithms. We ran M4GP over problems from PMLB, a new publicly available dataset suite (accessibly hosted on GitHub) initialized with 165 real-world, simulated, and toy benchmark datasets for evaluating supervised classification methods [5]. Of the 165 datasets we selected two suites of 10 datasets each. Note that PMLB focuses on classification benchmarks, whereas of the DEAP sample problems above only Spambase involves classification. Thus, our study included different types of problems.

We began by experimenting with a meta-level genetic algorithm over the space of EC parameters, of which we identified five major ones: Population size ($\in \mathbb{N}$, $[100, 3000]$), number of generations ($\in \mathbb{N}$, $[100, 2000]$), crossover rate ($\in \mathbb{R}$, $[0, 1]$), mutation rate ($\in \mathbb{R}$, $[0, 1]$), and tournament size ($\in \mathbb{N}$, $[3, 100]$). We experimented with the meta-GA for approximately two months, performing *tens of thousands of evolutionary runs*. We noted that numerous good parameter sets kept emerging, quite often appearing at random generation zero. Eventually, we turned to study just how rife with

GECCO '18 Companion, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
DOI: 10.1145/3205651.3208207

good parameters parameter space is, through a random search over parameter space, i.e., by generating parameter sets at random and executing a full evolutionary run per set. All told, our experiments involved a total of *93,615 GP runs*, each with a population size and generation count that could *both be as high as 1000 or 2000*.

We performed what is arguably one of the most extensive EC experiments conducted, sample results of which are shown in Figure 1. We came to the following conclusions:

- Good parameters range over the entire spectrum, somewhat in contrast with common lore, which tends to focus on ad-hoc "good" values.
- While one is usually inclined to increase population size and generation count we concluded that this need not be so. At most, the two should not both be very low.
- A commonly used range for tournament size is 3-7, but our experiments showed that many more values work just as well.
- Crossover and mutation rates can take on widely diverse values, departing from the oft-used high or intermediate crossover rate and low mutation rate. Moreover, crossover-mutation pairs showed no tendency to aggregate anywhere. At most, such pairs should not both be low (which makes sense given that an evolutionary algorithm requires inter-generational variation).

So perhaps one need not always spend too much time and resources on tuning hyper-parameters, with random search being a good choice for such tuning (after which EC will use these random hyper-parameters). Robustness to hyper-parameter tuning is a desired quality of an evolutionary algorithm and if one's algorithm requires very specific parameters, the chance of finding them is slim; this would essentially be a needle-in-a-haystack situation in hyper-parameter space.

ACKNOWLEDGMENTS

This work was supported by National Institutes of Health grants AI116794, DK112217, ES013508, HL134015, LM010098, LM011360, LM012601, and TR001263.

REFERENCES

[1] Agoston E Eiben and Selmar K Smit. 2011. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation* 1, 1 (2011), 19–31.

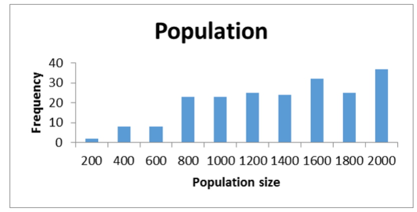
[2] A. E. Eiben and S. K. Smit. 2012. Evolutionary Algorithm Parameters and Methods to Tune Them. In *Autonomous Search*, Youssef Hamadi, Eric Monfroy, and Frédéric Saubion (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 15–36.

[3] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research* 13 (jul 2012), 2171–2175.

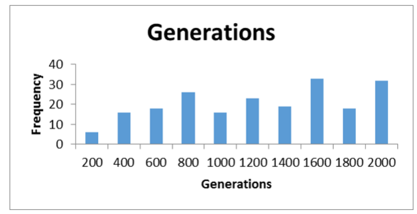
[4] William La Cava, Sara Silva, Leonardo Vanneschi, Lee Spector, and Jason Moore. 2017. Genetic programming representations for multi-dimensional feature learning in biomedical classification. In *European Conference on the Applications of Evolutionary Computation*. Springer, Berlin, Heidelberg, 158–173.

[5] Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. 2017. PMLB: a large benchmark suite for machine learning evaluation and comparison. *BioData Mining* 10, 1 (11 Dec 2017), 36.

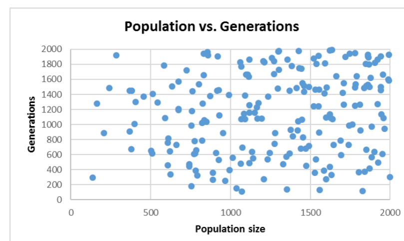
[6] Moshe Sipper, Weixuan Fu, Karuna Ahuja, and Jason H Moore. 2018. Investigating the parameter space of evolutionary algorithms. *BioData Mining* 11, 2 (2018), 1–14.



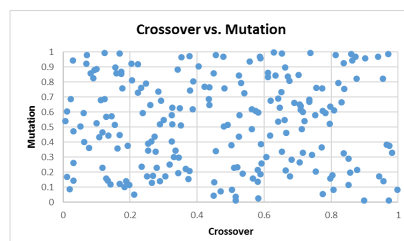
(a)



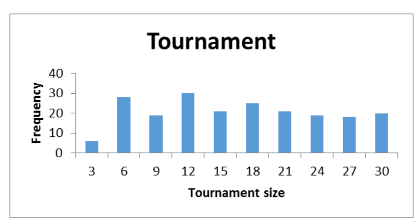
(b)



(c)



(d)



(e)

Figure 1: M4GP over the 10 problems of one of the suites of datasets we used. Shown are plots for the successful parameter sets found: population size (a), generation count (b), population size vs. generation count (c), crossover rate vs. mutation rate (d), and tournament size (e).