Increasing Genetic Programming Robustness using Simulated Dunning-Kruger Effect

Thomas D. Griffiths
grifftd1@aston.ac.uk
Aston Lab for Intelligent Collectives Engineering (ALICE)
Aston University, Birmingham B4 7ET, U.K.

Anikó Ekárt
a.ekart@aston.ac.uk
Aston Lab for Intelligent Collectives Engineering (ALICE)
Aston University, Birmingham B4 7ET, U.K.

ABSTRACT

Robustness is a key characteristic of genetic programming candidate solutions, providing the ability to maintain a satisfactory level of performance under dynamic and uncertain environments. In this paper we perform experiments on Tartarus problem instances[2] exploring the hypothesis that the introduction of a fitness distribution bias, inspired by the Dunning-Kruger effect [5], will lead to an increase in the diversity and robustness of candidate solutions.

CCS CONCEPTS

• **Theory of computation** → *Design and analysis of algorithms*; • **Computing methodologies** → *Genetic programming*.

KEYWORDS

Robustness, Diversity, Tartarus Problem, Dunning–Kruger

ACM Reference Format:
Thomas D. Griffiths and Anikó Ekárt. 2019. Increasing Genetic Programming Robustness using Simulated Dunning-Kruger Effect. In . ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3321885

1 INTRODUCTION

The prevalence of candidate solutions prematurely converging to sub-optimal performance is an ever-present issue in the field of Genetic Programming. A population is said to have converged when the level of diversity present in the population collapses to near-zero and there is no increase in fitness score throughout a generation. Without the proper application of genetic operators or heuristic intervention, it is unlikely that the population will recover from this drop in diversity [3]. It is hypothesised that through the manipulation of fitness value distributions, it will be possible to increase the level of diversity present in the population.

2 POPULATION ROBUSTNESS

Robustness is often referred to as a characteristic of a candidate solution whose performance is not diminished despite changes in environmental parameters or constraints.

A solution that does not lose its utility or performance quality under these changes is said to be robust [1]. Robustness can be broadly classified into two groups:

- *Phenotypic robustness* – the number of unique genotypes that map to a given phenotype, and,
- *Genotypic robustness* – the likelihood that a genotype will produce the same phenotype under single-point mutation.

Figure 1 illustrates the relationship between genotypes and phenotypes in a two-dimensional representation of a geno-pheno space, showing 6 phenotypes, P_{a-f} and 2 genotypes, G_{1-2}.

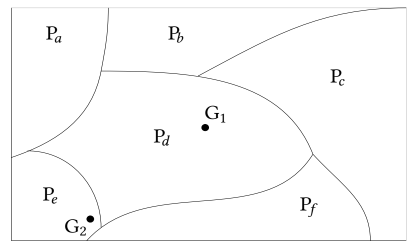

Figure 1: Hypothetical Geno-Pheno Space.

2.1 Phenotypic Robustness

The robustness of a phenotype is analogous to the size of its genotype network, defined as the number of unique genotypes present in the population that map to the given phenotype [4]. A phenotype which has a large genotype network is considered to be more robust than one with a smaller genotype network. This can be conceptualised as being analogous to the size of the area of the geno-pheno map occupied by the individual phenotype. It can be seen in Figure 1 that the phenotypes P_d and P_c occupy a much larger area of the geno-pheno space than the phenotypes P_e and P_a, and would therefore be considered to be more robust.

2.2 Genotypic Robustness

Canonical genetic programming genotypes can be compared to each other in terms of their *n*-neighbour relationship. For example, genotypes which have $n = 1$ chromosome difference between them are said to have a *1*-neighbour or *1*n relationship, and so forth.

Genotypes with a *1*n relationship are said to be adjacent to each other in the wider geno-pheno space, shown in Figure 1. A genotype is considered robust if it is able to map to the same phenotype under the effects of single-point mutation [4], which can be conceptualised in terms of its neighbour relationships. It is likely that for genotype G_1, the genotypes with a *1*n or *2*n relationship still map to the phenotype P_d, known as *neutral* neighbours.

However, for the genotype G_2, which lies closer to the boundaries between the phenotypes P_e, P_d and P_f, it is more likely that the genotypes with a *1*n or *2*n relationship may map to an entirely different phenotype, known as *non-neutral* neighbours. The relative robustness of a genotype can be measured by comparing the number of neutral and non-neutral neighbours [4].

GECCO '19 Companion, July 13-17 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07...$15.00
https://doi.org/10.1145/3319619.3321885

3 DUNNING-KRUGER EFFECT

The Dunning-Kruger effect is a form of cognitive bias observed in populations [5]. It is described that individuals with a low level of ability mistakenly over-estimate their performance and conversely, individuals with a high level of ability will often mistakenly under-estimate their performance, as illustrated in Figure 2.

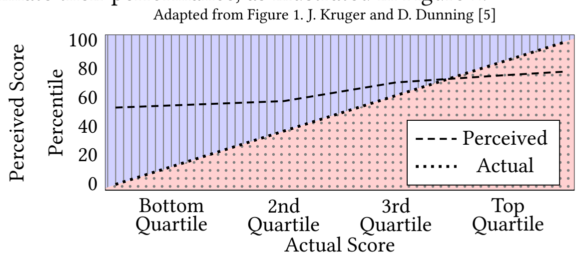

Figure 2: Comparison Between Actual & Perceived Scores

3.1 Implementing Dunning–Kruger Bias

We propose that the introduction of a Dunning-Kruger style bias (DK) into the fitness distribution of a population will enable us to maintain a higher level of population diversity over time. This would be achieved by means of modifying the fitness scores of individuals in the distribution based on their performance relative to the rest of the population. This is achieved with the lower performing individuals having their reported fitness scores artificially increased and the higher performing individuals having their reported fitness scores decreased. The reported fitness score of each individual i is modified at the end of the generation, prior to the execution of the genetic operators, using the following function:

$$\mathrm{DK}_i = F_i * 50 - 0.75p * \frac{F_{max} - F_{min}}{2},$$

where DK_i represents the new biased fitness score and F_i the original fitness score for individual i, the constants are linear approximations of Figure 2, p is the percentile fitness rank of the individual in the population, F_{min} and F_{max} are the minimum and maximum fitness scores found in the population fitness distribution.

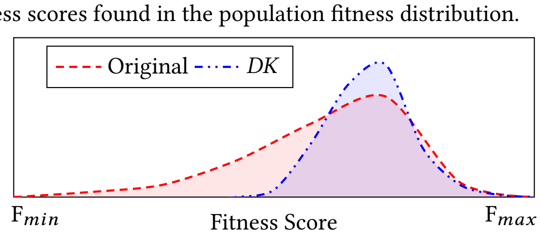

Figure 3: Comparison Between Fitness Distributions

Figure 3 a comparison between the original and DK fitness distributions is shown. In the original distribution, there is a significant minority of individuals whose performance lies around the median value, but whom are unlikely to be chosen for recombination; contributing, in part, to the long-term loss of overall population diversity. We postulate that the modification of the fitness distribution will reduce the evolutionary pressure present in the population, leading to a higher level of long-term diversity. The proposed approach is similar to that of fitness sharing. However, in contrast to fitness sharing where the fitness of individuals is modified in relation to a distance metric or neighbourhood, the fitness value of DK_i is modified based on global performance in comparison to the rest of the population.

4 RESULTS

A selection of tartarus problem instances, of size 8×8 were generated [2]. A series of experiments were conducted where the instance was changed during execution, at regular generation intervals: $\{10,\ 20,\ 50\}$. This change in environment and instance was used to assess the robustness of candidate solution to change, the results of which are shown in Table 1.

Table 1: Average Fitness Comparison for Canonical and DK.

	G=10		G=20		G=50	
	+1	+5	+1	+5	+1	+5
Original	-0.959	-0.740	-1.264	-1.087	-1.366	-1.412
DK	**-0.764**	**-0.572**	**-1.258**	**-0.944**	**-1.272**	**-1.173**

The change in average population fitness for *one generation*: **+1** and *five generations*: **+5**, after the instance change is shown. It can be seen that the system utilising DK is able to recover fitness performance faster than the canonical system, supporting the supposition that the introduction of DK leads to an increase in robustness.

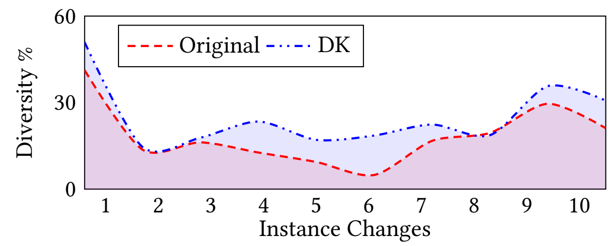

Figure 4: Average Levels of Phenotypic Diversity

Figure 4 shows the average level of phenotypic diversity present in the population for the canonical and DK systems, calculated utilising the fitness spread technique, using the *true* fitness values present in the population, opposed to the *biased* DK fitness values.

5 CONCLUSION

In this paper we present a novel approach for increasing the diversity of a genetic programming population, utilising the DK fitness distribution bias to modify the reported fitness scores of individuals in the population. The results indicate that the modification of the reported fitness values leads to a small but significant increase in the overall level of diversity present in the population. We demonstrated that this increase in diversity within the genetic programming population leads to an increase in the robustness of the candidate solutions generated. The candidate solutions from the system utilising the DK bias were able to recover from changes in the environment, faster and more effectively than the ones without.

REFERENCES

[1] D.F. Dunham. 2015. Robustness of Genetic Algorithm Solutions in Resource Levelling. In *IEEE Systems and Information Engineering Design Symposium.*
[2] T.D. Griffiths and A. Ekárt. 2017. Improving the Tartarus Problem as a Benchmark in Genetic Programming. In *Proceedings of the 20th European Conference on Genetic Programming - EuroGP'17.* 278–293.
[3] T.D. Griffiths and A. Ekárt. 2018. Improving the Effectiveness of Genetic Programming using Continuous Self-Adaptation. In *Proceedings of the 2nd Artificial Life and Intelligent Agents Symposium - ALIA'16.* 97–102.
[4] T. Hu and W. Banzhaf. 2018. Neutrality, Robustness, and Evolvability in Genetic Programming. In *Genetic Programming Theory and Practice XIV.* 101–117.
[5] J. Kruger and D. Dunning. 1999. Unskilled and Unaware of it: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Awareness. *Journal of Personality and Social Psychology* 77(6), 1121–1134.