# Revisiting Interval Arithmetic for Regression Problems in Genetic Programming*

Grant Dick
Department of Information Science
University of Otago
Dunedin, New Zealand
grant.dick@otago.ac.nz

## ABSTRACT

Traditional approaches to symbolic regression require the use of protected operators, which can lead to perverse model characteristics and poor generalisation. In this paper, we revisit interval arithmetic as one possible solution to allow genetic programming to perform regression using unprotected operators. Using standard benchmarks, we show that using interval arithmetic within model evaluation does not prevent invalid solutions from entering the population, meaning that search performance remains compromised. We extend the basic interval arithmetic concept with 'safe' search operators that integrate interval information into their process, thereby greatly reducing the number of invalid solutions produced during search. The resulting algorithms are able to more effectively identify good models that generalise well to unseen data.

## CCS CONCEPTS

•**Computing methodologies → Genetic programming; Model verification and validation;** Supervised learning;

## KEYWORDS

genetic programming; symbolic regression; interval arithmetic

**ACM Reference format:**
Grant Dick. 2017. Revisiting Interval Arithmetic for Regression Problems in Genetic Programming. In *Proceedings of the Genetic and Evolutionary Computation Conference 2017, Berlin, Germany, July 15–19, 2017 (GECCO '17),* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3076107

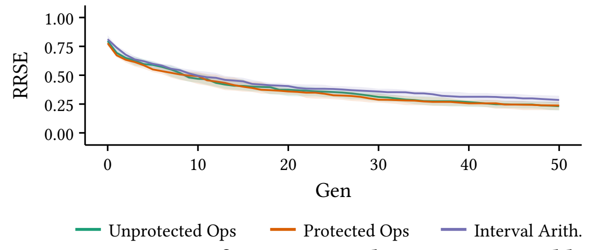

**Figure 1: Training performance on the *Keijzer-10* problem.**

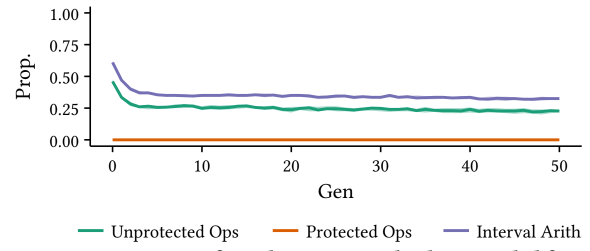

**Figure 2: Proportion of popluation marked as invalid for the *Keijzer-10* problem.**

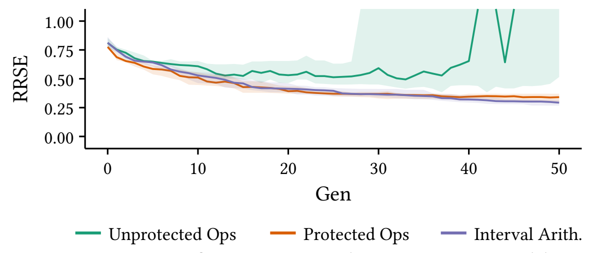

**Figure 3: Test performance on the *Keijzer-10* problem.**

## 1 INTRODUCTION

Using genetic programming (GP) to evolve models via symbolic regression simultaneously searches for a suitable model structure and its corresponding parameters. This frees the user from a priori decisions pertaining to model structure and may provide useful feedback and understanding of the problem domain. A long-known issue with symbolic regression is the need for closure [4]. By far, the most common solution to this is to use 'protected' operators with built-in exception handling, such as for division by zero. An alternative has been proposed that applies interval arithmetic to identify safe and valid use of operators, and previous work has demonstrated some success in applying interval arithmetic over protected operators [3]. However, most work incorporating interval arithmetic into GP uses it purely in an evaluative framework, and so the dynamics of interval arithmetic within the population remain largely unexplored. The goal of this paper is to explore how interval arithmetic may be used to change the behaviour of GP by using interval arithmetic to guide safe and effective use of search operators. New search operators are presented in this paper that attempt to honour the intervals presented by the problem domain to ensure that offspring remain valid during search. This increases the rate of search within the population, and leads to more rapid evolution towards fit models.

*A full version of this paper is available [2]

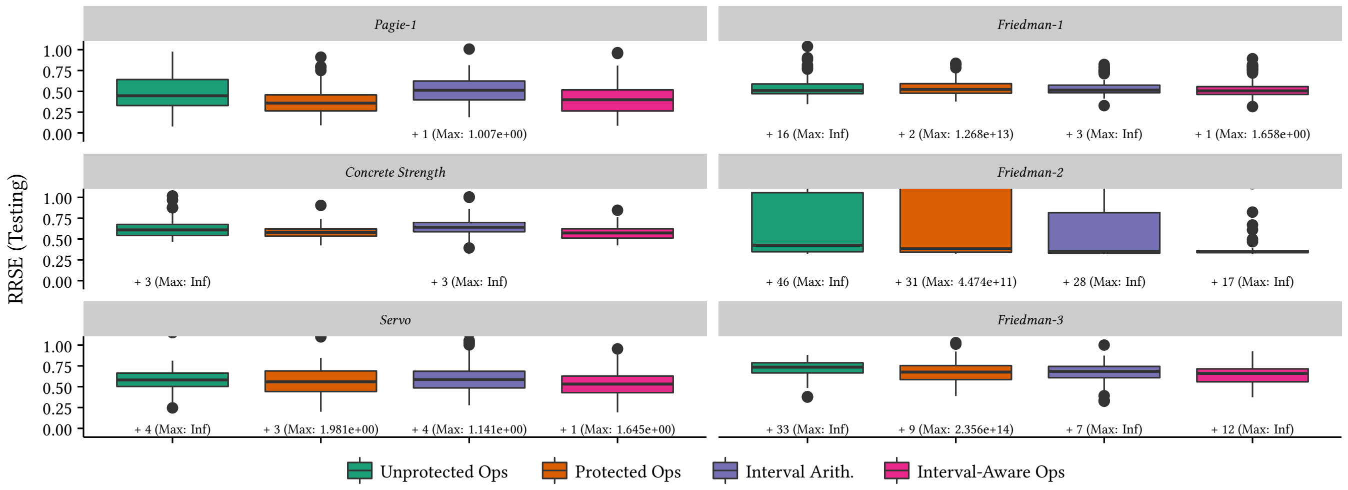

**Figure 4: Test performance on benchmark problems.**

## 2 BEHAVIOUR OF INTERVAL ARITHMETIC

We start by explaining the behaviour of interval arithmetic through exploring the *Keijzer-10* problem from previous work [3]: this problem proved a challenge for GP, which without interval arithmetic failed to find a meaningful solution in 98% of runs. Here we explore this problem again, using a standard implementation of GP using protected and unprotected operators, and then augmented with interval arithmetic to perform static evaluation.

The results of the initial analysis using *Keijzer-10* are shown in Figures 1–3. The results suggest that both protected and unprotected operators appear to allow the GP system to evolve at a faster rate than when using interval arithmetic and static analysis. A possible reason for this is that interval analysis fails to prevent invalid individuals from entering the population (see Figure 2), and this effectively reduces the population size. While the integration of interval arithmetic into GP appears to compromise training performance, it is offset by stronger generalisation performance.

## 3 INTERVAL-PRESERVING OPERATORS

The results in the previous section suggest that including interval arithmetic in GP can improve its generalisation performance but may also reduce the effective size of the population and thus require larger populations and more generations to scale to harder problems. However, this may be a consequence of considering interval arithmetic solely for use in the assignment of fitness to individuals. Previous work identified that interval arithmetic could also be used to preserve working bounds of solutions developed through geometric semantic genetic programming [1]. The so-called ‘safe initialisation’ method modifies the tree initialisation process of GP such that the actual choice of operator at a given node is delayed until all the necessary child nodes have been created. Given that the tree initialisation process in GSGP is not substantially different from standard GP, it should be reasonably straightforward to integrate here. Additionally, ensuring valid solutions during tree initialisation is only one aspect of operator protection that can be considered: the results in the previous section suggest that the search operators themselves may also contribute to the generation of invalid solutions. Therefore, this paper extends the work done exploring safe initialisation: crossover and mutation operators are modified so that they produce individuals that maintain useful intervals. Following a normal crossover or mutation operation, the parent node of the site in the offspring that is modified is checked for valid execution interval. If the interval at this parent node is invalid, then a search is performed to find an operator that produces a valid output interval using the the child intervals as inputs. Once a valid operator is determined, the interval check then proceeds up the tree until the root node is encountered. The impact of these interval-aware operators on several benchmark problems can be seen in Figure 4, where these interval-aware operators can be seen to offer a general improvement in performance over both static analysis through interval arithmetic and protected operators.

## 4 CONCLUSION

Symbolic regression has the potential to be a useful method for machine learing and data science. Previous work has advocated the use of interval arithmetic to eliminate the requirement for protected operators. This paper has extended previous work to uncover the dynamics of interval arithmetic during the evolutionary process. In the process, new operators have been developed that greatly reduce the number of invalid solutions that are generated during a run, which allows evolution to proceed at a greater rate then by using interval arithmetic solely in the evaluation of individuals.

Extended analysis of the results in this paper can be found in [2].

## REFERENCES

[1] Grant Dick. 2015. Improving Geometric Semantic Genetic Programming with Safe Tree Initialisation. In *European Conference on Genetic Programming*. Springer International Publishing, 28–40.

[2] Grant Dick. 2017. Interval Arithmetic and Interval-Aware Operators for Genetic Programming. (2017). arXiv:arXiv:1704.04998

[3] Maarten Keijzer. 2003. Improving Symbolic Regression with Interval Arithmetic and Linear Scaling. In *Genetic Programming*, Conor Ryan, Terence Soule, Maarten Keijzer, Edward Tsang, Riccardo Poli, and Ernesto Costa (Eds.). Lecture Notes in Computer Science, Vol. 2610. Springer Berlin Heidelberg, 70–82.

[4] John R. Koza. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, USA.