

# The Use of an Analytic Quotient Operator in Genetic Programming

Ji Ni, Russ H. Driberg, and Peter I. Rockett

**Abstract**—We propose replacing the division operator used in genetic programming with an analytic quotient (AQ) operator. We demonstrate that this AQ operator systematically yields lower mean squared errors over a range of regression tasks, due principally to removing the discontinuities or singularities that can often result from using either protected or unprotected division. Further, the AQ operator is differentiable. We also show that the new AQ operator stabilizes the variance of the intermediate quantities in the tree.

**Index Terms**—Analytic quotient (AQ), genetic programming (GP), protected division (PD), variance stabilization.

## I. INTRODUCTION

GENETIC programming (GP) has proved a valuable and flexible technique for a range of regression and classification problems [1]. Although some authors also used internal function nodes implementing transcendental functions, almost all reported results have employed the basic arithmetic operations of addition, subtraction, multiplication and some form of division, a function set that appears to date back to Koza [2].

The inclusion of a division-style operator giving a GP tree the ability to split some quantity into a number of “parts” provides an intuitively appealing richness to the function set, but conventional division can be problematic when the denominator is zero in which case the division operation is, strictly, undefined. This is in contrast to the common computer implementation, compliant with the IEEE754:1985 standard for floating-point arithmetic [3], where  $a/0$  returns  $\infty$  for  $a \neq 0$ , but  $0/0$  returns not-a-number (NaN), a special representation for indeterminate quantities.

Koza’s original concern was with operator *closure*—the desire for an operation on a real number to always map to another (proper) real number—although he also recognized the difficulties with the normal division operator and therefore introduced *protected* division whereby

$$PD(x_1, x_2) = \begin{cases} x_1/x_2 & x_2 \neq 0 \\ 1 & x_2 = 0. \end{cases}$$

Manuscript received July 28, 2011; revised December 16, 2011 and March 29, 2012; accepted April 9, 2012. Date of publication April 18, 2012; date of current version January 25, 2013. The work of R. H. Driberg was supported by an EPSRC Doctoral Training Award.

The authors are with the Department of Electronics and Electrical Engineering, University of Sheffield, Sheffield S1 3JD, U.K. (e-mail: j.ni@shef.ac.uk; r.driberg@shef.ac.uk; p.rockett@sheffield.ac.uk).

Digital Object Identifier 10.1109/TEVC.2012.2195319

The use of protected division (PD) appears “common” [1] although maybe not universal, and some authors appear to use an *unprotected* form of the division operation. A random survey of 20 recent GP papers suggests that 18 authors used PD although some report only the use of “division”; whether this is truly an unprotected division (UPD) or actually a PD which has been referred to with the unqualified shorthand of “division” is unclear.<sup>1</sup>

In practice, it is comparatively rare in floating-point calculations for a quantity to be exactly zero although small quantities are common. In particular, for  $x_1/x_2$ , when  $|x_2|$  is very small but strictly greater than zero and  $x_1 \neq 0$ , both the protected and unprotected division operations can produce “spikes” of very large values, which is highly undesirable for approximating continuous functions. The shortcomings of protected division were previously identified by Keijzer [4] who proposed using interval arithmetic to probe the regions around training points for discontinuities. Unfortunately, Keijzer’s work seems to have been largely ignored. Similarly, the option of simply omitting the problematic division operator has been explored (see [4]), but the resulting function set is far less expressive.

Additionally, PD embeds a discontinuity whenever  $x_2 = 0$  and therefore renders the function represented by the whole tree nonanalytic; implementations using UPD in IEEE754 arithmetic will similarly produce a singularity when  $x_2 = 0$ . This inability to differentiate the tree function restricts the range of operations which can be carried out on the tree. For example, the nonanalyticity due to (un)protected division [(U)PD] prevents the use of curvature as a complexity measure [5].

In this paper, we propose replacing the PD operator by an analytic quotient (AQ) operator defined by

$$AQ(x_1, x_2) = \frac{x_1}{\sqrt{1+x_2^2}} \quad (1)$$

which has the general properties of division, especially when  $x_2 \gg 1$ , but is everywhere differentiable. Given that the empirical modeling of data has been carried out using a very wide range of functionals [6], we can see no fundamental reason why the (U)PD operator is sacrosanct. Our exclusive focus in this paper is on the *empirical* modeling of data; we specifically exclude attempts to identify some “true” generating function of the data by symbolic regression.

<sup>1</sup>We have extended the coverage of this paper to UPD at the suggestion of an anonymous reviewer.

TABLE I  
 TEST FUNCTIONS USED IN THIS PAPER

$F_1$	$f(x) = 4.26(e^{-x} - 4e^{-2x} + 3e^{-3x})$	$x \in [0 \dots 3.25]$	Automatic French curve [7]
$F_2$	$f(x) = 3\cos(3\cos^{-1}x)$	$x \in [-1 \dots +1]$	Chebyshev polynomial
$F_3$	$f(x) = 5\sin(x)/x$	$x \in (0 \dots 10]$	Scaled <i>sinc</i> function
$F_4$	$f(x_1, x_2) = (x_1 - 3)(x_2 - 3) + 2\sin(x_1 - 4)(x_2 - 4)$	$x_i \in [0 \dots 5], i = 1, 2, 3$	“Ripple” function [8]
$F_5$	$f(x_1, x_2, x_3) = 30 \frac{(x_1-1)(x_3-1)}{x_2^2(x_1-10)}$	$x_1, x_3 \in [0 \dots 5], x_2 \in [1 \dots 2]$	“RatPol3D” function [8]
$F_6$	$f(x_1, x_2, x_3, x_4, x_5) = \frac{10}{5 + \sum_{i=1}^5 (x_i - 3)^2}$	$x_i \in [0 \dots 5], i = 1 \dots 5$	“UBall5D” function [8]

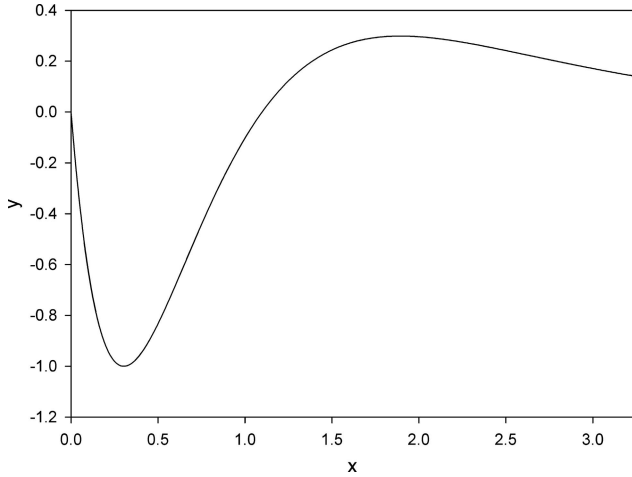


Fig. 1. Automatic French curve of Wahba and Wold [7].

This paper is structured as follows. Section II describes the experimental methodology we have employed and in Section III we report results which demonstrate that the AQ operator gives consistently lower generalization errors over a range of regression problems, an observation which we explore further in Section IV.

## II. METHODOLOGY

### A. Datasets

We have considered six regression problems ranging from one to five dimensional. We have generated 100 independent training sets from each function comprising 30 data per dimension, randomly selected from the domain, and added zero-mean Gaussian noise with a variance of 0.01 to each training instance. The six functions are listed in Table I. For convenient comparison with what follows, the well-known French curve of Wahba and Wold [7] is shown in Fig. 1.

Independent test sets comprising 100 000 randomly-drawn instances were used to assess generalization performance.

### B. GP Algorithms

To ensure our results are not dependent on evolutionary strategy we have employed both single-objective (SO) and multiobjective (MO) genetic programming in the experimental work. For both SOGP and MOGP strategies we have used both generational GP with 50% elitism, and steady-state algorithms. We have used rank-based selection for both SO and MOGP strategies. For the steady-state MOGP experiments we have

 TABLE II  
 GP PARAMETERS USED IN THIS PAPER

Population size	100
No. of evaluations (steady-state)	20 000
No. of generations (generational)	398
Crossover	Point mutation [1]
Mutation strategy	Point mutation [1], full depth of 4
Node types	Unary minus Addition, subtraction Multiplication UPD, PD or analytical quotient

used the PCGP algorithm [9]. The GP parameters are summarized in Table II. The steady-state algorithms were run for a constant 20 000 tree evaluations per run and the generational algorithms for 398 generations (which corresponds to an equivalent 20 000 tree evaluations).

An enduring problem in GP is *bloat*, the tendency for trees to continue increasing in size without any improvement in fitness [1]. For the single-objective GP experiments we have used the dynamic depth-control method of Silva and Almeida [10] to control bloat. For the multiobjective GP experiments, we have controlled bloat by simultaneously minimizing: 1) mean squared error (MSE) over the training set, and 2) tree node count, including terminals, within a Pareto framework [11]. See [12] for further details.

## III. RESULTS

For a given generating function (see Table I) and for each of the 100 training sets, we have conducted the paired experiment of minimizing the training set MSE (SOGP) or training set MSE/node count (MOGP) using trees with either PD, UPD, or AQ operators. For both SO and MOGP methods, on the termination of each GP run, we have selected the individual with the best MSE over the test set and compared the values arising from the use of the AQ, PD, and UPD operators.

Obtaining the UPD results, which we include for completeness, proved problematic. During evolution, a large fraction of the trees using completely UPD evaluated to NaN [3], i.e., indeterminate values. This posed a problem with our tree sorting algorithm since any binary comparison of a floating-point number with a NaN always returns false [3]. On further investigation, the source of these NaNs was found to be zero-divided-by-zero operations. We thus modified our UPD

TABLE III  
SUMMARY OF BEST TEST ERRORS FOR SINGLE OBJECTIVE GENERATIONAL GP FOR VARYING DIVISION OPERATORS

	AQ	PD	UPD
<i>F1</i>	0.0109 ± 0.0106	22.98 ± 147.4	2.8224 ± 15.14283394
<i>F2</i>	0.1382 ± 0.2190	2 453 666 ± 24 534 512	36.436 ± 188.9573505
<i>F3</i>	0.2061 ± 0.1697	684.85 ± 4312	2.5611 × 10 <sup>10</sup> ± 2.56115 × 10 <sup>11</sup>
<i>F4</i>	3.040 ± 1.096	150 584 ± 1 293 396	705 988 ± 5 130 451
<i>F5</i>	3.169 ± 2.058	46.145 ± 172.39	95.221 ± 318.33
<i>F6</i>	0.0338 ± 0.0071	0.7288 ± 3.5013	4.7056 ± 842.24

Errors are shown ± one SD.

TABLE IV  
SUMMARY OF BEST TEST ERRORS FOR SINGLE OBJECTIVE STEADY-STATE GP FOR VARYING DIVISION OPERATORS

	AQ	PD	UPD
<i>F1</i>	0.01165 ± 0.01279	2.0904 × 10 <sup>11</sup> ± 1.7908 × 10 <sup>12</sup>	1.8279 × 10 <sup>21</sup> ± 1.8279 × 10 <sup>22</sup>
<i>F2</i>	0.08509 ± 0.19400	2.6514 × 10 <sup>13</sup> ± 2.3224 × 10 <sup>14</sup>	3.6211 × 10 <sup>22</sup> ± 3.6211 × 10 <sup>23</sup>
<i>F3</i>	0.40464 ± 0.88382	7.7702 × 10 <sup>11</sup> ± 7.3240 × 10 <sup>12</sup>	4.7844 × 10 <sup>79</sup> ± 4.7844 × 10 <sup>80</sup>
<i>F4</i>	5.4449 ± 15.307	9.2040 × 10 <sup>11</sup> ± 6.4689 × 10 <sup>12</sup>	8.0715 × 10 <sup>23</sup> ± 8.072 × 10 <sup>24</sup>
<i>F5</i>	2.5774 ± 3.5450	1.4455 × 10 <sup>24</sup> ± 1.4455 × 10 <sup>25</sup>	1.2375 × 10 <sup>43</sup> ± 1.2375 × 10 <sup>44</sup>
<i>F6</i>	0.03300 ± 0.00617	3.6094 × 10 <sup>9</sup> ± 3.6061 × 10 <sup>10</sup>	1.6648 × 10 <sup>21</sup> ± 1.6648 × 10 <sup>22</sup>

TABLE V  
SUMMARY OF BEST TEST ERRORS FOR MULTIOBJECTIVE GENERATIONAL GP FOR VARYING DIVISION OPERATORS

	AQ	PD	UPD
<i>F1</i>	0.01145 ± 0.00643	0.02486 ± 0.03464	0.03276 ± 0.04220
<i>F2</i>	0.23613 ± 0.25591	0.55210 ± 0.59903	0.58528 ± 0.58265
<i>F3</i>	0.28542 ± 0.11390	1.0011 ± 1.0591	0.80542 ± 0.89571
<i>F4</i>	3.40132 ± 0.83564	6.1088 ± 1.3939	6.1632 ± 1.4100
<i>F5</i>	5.2364 ± 3.2134	6.8237 ± 3.8258	6.3948 ± 3.4893
<i>F6</i>	0.03350 ± 0.00470	0.04631 ± 0.00470	0.04605 ± 0.0055

TABLE VI  
SUMMARY OF BEST TEST ERRORS FOR MULTIOBJECTIVE STEADY-STATE GP FOR VARYING DIVISION OPERATORS

	AQ	PD	UPD
<i>F1</i>	0.00971 ± 0.00748	0.01950 ± 0.03674	0.01959 ± 0.03215
<i>F2</i>	0.09154 ± 0.16742	0.28373 ± 0.59593	0.33525 ± 0.63592
<i>F3</i>	0.18396 ± 0.12380	0.81131 ± 1.0199	0.69990 ± 0.77649
<i>F4</i>	2.6046 ± 0.73180	5.0935 ± 1.8264	5.4817 ± 1.8136
<i>F5</i>	2.62175 ± 1.9555	3.3771 ± 2.3554	3.2286 ± 1.7834
<i>F6</i>	0.03028 ± 0.00459	0.04127 ± 0.00615	0.04002 ± 0.00589

procedure to assign any tree evaluating to NaN to have a very large fitness, making it unlikely to be selected for breeding in subsequent iterations [1]. (We speculate that anybody using truly UPD in GP must be implementing sorting or tournament selection in a way that silently discards NaN-evaluating trees.)

All the mean test error results are summarized in Tables III–VI. Some of the mean test errors, particularly for SOGP (Tables III, IV), are very large. This reflects not any lack of convergence to small training errors but the fact that over 100 trials, although some runs produced very small test errors, some produced extremely large test errors resulting in an overall very large mean test error. [The large spread of test errors also results in large values of standard deviation

(SD), as can be seen in Tables III and IV.] This trend is most marked for the steady-state single-objective algorithm. An unambiguous general trend emerges that for any given evolutionary setup (e.g., single objective+generational), trees using the AQ operator always yielded the smallest mean test errors. Comparing PD and UPD, no clear pattern emerges—sometimes PD is better, sometimes UPD. Overall, the steady-state MOGP algorithm yielded the lowest mean test errors over the set of functions explored here.

An equally important observation is that, comparing within a given evolutionary setup (i.e., by table), AQ yields the smallest SD of any operator (with the single exception of PD for *F6* under generational MOGP where the SD is the same as for AQ). This implies that AQ consistently produces more compact distributions of test errors across different runs leading to greater repeatability and a far higher probability of obtaining a good generalization error from a limited number of runs.

We have computed the test errors for each of the 100 training runs by averaging the squared errors over the relevant test set of 100K data. We have performed nonparametric one-sided sign tests on each set of 100 paired differences under the null hypothesis that the median is less than or equal to zero. The necessary binomial probability can be well-approximated by using a normal distribution leading to *p*-values supporting the null hypotheses [13]. The results of the statistical comparisons are shown in Tables VII–X from which it can be seen that for all strategies (generational and steady-state, single and multiple objectives) and for all functions (*F1*–*F6*), the *p*-values indicate that the AQ outperforms both protected and UPDs with around 99% confidence or greater. (Due to the limited power of the nonparametric test, a few of the tests using 100 paired samples yielded *p*-values that provided no clear evidence either way about the null hypothesis. We repeated these tests using 500 training sets

TABLE VII

STATISTICAL COMPARISONS OF DIFFERENCES IN BEST MSE FOR SINGLE-OBJECTIVE GENERATIONAL GP

	AQ versus PD		AQ versus UPD	
	Z score	<i>p</i> -value	Z score	<i>p</i> -value
<i>F1</i>	3.4	0.000337	4.4	$5.4 \times 10^{-6}$
<i>F2</i>	5.6	$1.1 \times 10^{-8}$	3.8	$7.2 \times 10^{-5}$
<i>F3</i>	9.0	0	8.6	0
<i>F4</i>	9.4	0	9.6	0
<i>F5</i>	5.4	$3.3 \times 10^{-8}$	6.4	$7.8 \times 10^{-11}$
<i>F6</i>	9.4	0	9.0	0

TABLE VIII

STATISTICAL COMPARISONS OF DIFFERENCES IN BEST MSE FOR SINGLE-OBJECTIVE STEADY-STATE GP

	AQ versus PD		AQ versus UPD	
	Z score	<i>p</i> -value	Z score	<i>p</i> -value
<i>F1</i>	9.6	0	9.6	0
<i>F2</i>	9.8	0	9.2	0
<i>F3</i>	9.4	0	9.8	0
<i>F4</i>	10	0	10	0
<i>F5</i>	9.4	0	9.2	0
<i>F6</i>	9.6	0	9.8	0

TABLE IX

STATISTICAL COMPARISONS OF DIFFERENCES IN BEST MSE FOR MULTIOBJECTIVE GENERATIONAL GP, 100 PAIRED SAMPLES

	AQ versus PD		AQ versus UPD	
	Z score	<i>p</i> -value	Z score	<i>p</i> -value
<i>F1</i>	3.2	0.00687	5.6	$1.07 \times 10^{-8}$
<i>F2</i>	4.2	$1.33 \times 10^{-5}$	4.4	$5.41 \times 10^{-6}$
<i>F3</i>	8.4	0	8.4	0
<i>F4</i>	8.8	0	9.2	0
<i>F5</i>	3.4	0.000337	2.4	0.008198
<i>F6</i>	9.6	0	9.6	0

to give the figures shown in bold in Table X.) The AQ is thus much more likely to produce individuals exhibiting statistically better generalization.

IV. DISCUSSION

To further explore the reasons for the superiority of the AQ operator, we show in Fig. 2 an example of a GP fit to the French curve function [7] (*F1*) for one particular training set instance using PD. The filled circles represent the training data and the solid line shows the evolved mapping over the domain. Note the change of scale compared to Fig. 1. Although the evolutionary pressure has ensured that the fitted function’s deviation is small *at* the training data, there is nothing to constrain the function away from the training points. For  $x \approx 0.3$ , the fitted function has a discontinuity rising up to  $\sim 70$  and falling down to  $\sim -100$  before returning to a smooth curve; the AQ does not produce such discontinuities. Although we show only one example for the sake of brevity, and although not every fit using (U)PD exhibits discontinuities, such behavior is not uncommon. Indeed, this phenomenon has previously been reported by Keijzer [4] who gave other examples.

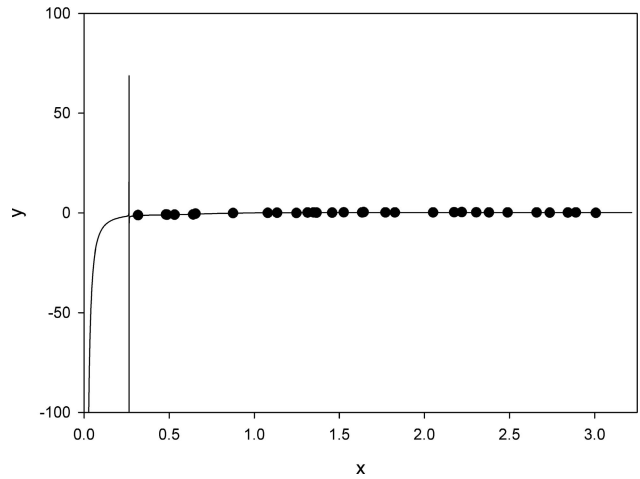


Fig. 2. Fitted automatic French curve. Filled points are the training data and the solid line the fitted model. Note the scale change compared to Fig. 1.

TABLE X

STATISTICAL COMPARISONS OF DIFFERENCES IN BEST MSE FOR MULTIOBJECTIVE STEADY-STATE GP, 100 PAIRED SAMPLES APART FROM RESULTS IN BOLD FACE (SEE THE TEXT FOR FURTHER DETAILS)

	AQ versus PD		AQ versus UPD	
	Z score	<i>p</i> -value	Z score	<i>p</i> -value
<i>F1</i>	<b>3.0</b>	<b>0.00135</b>	<b>3.4</b>	<b>0.00034</b>
<i>F2</i>	3.2	0.00687	4.2	$1.33 \times 10^{-5}$
<i>F3</i>	8.8	0	9.4	0
<i>F4</i>	9	0	9.2	0
<i>F5</i>	2.4	0.008198	<b>6.3</b>	<b><math>1.5 \times 10^{-10}</math></b>
<i>F6</i>	7.8	$3.11 \times 10^{-15}$	8.6	0

In Fig. 3, we plot two typical examples of the cumulative probability distributions of the mean squared test errors over the sets of 100 runs for both the PD and AQ trees for the individuals with the best training errors in their run. The distributions for the AQ trees (shown dashed) are compact indicating that there is a good correlation between low training error and low test error. For the PD trees (shown as solid lines), although there are some instances of highly performing individuals, it is clear that many of the GP runs are producing very large best test errors despite these individuals having the lowest training errors in their respective runs. This frequent association of low training error with very large test error is a result of the instabilities introduced by (U)PD, as illustrated in Fig. 2, but is completely absent from trees using the AQ operator. This phenomenon of some GP runs producing best-performing individuals with large test errors has frequently (but incorrectly) been ascribed in the literature to overfitting or a “bad run,” whereas we suspect it is, in fact, caused by the instabilities of the (U)PD operator. Although using interval arithmetic to probe the regions around the training points [4] can mitigate the instabilities due to (U)PD, interval arithmetic cannot completely remove the problem—unless a discontinuity falls within the interval around a training datum it will not be identified.

All the results reported above have used the functional form for the AQ given by (1). It is possible to envisage a more

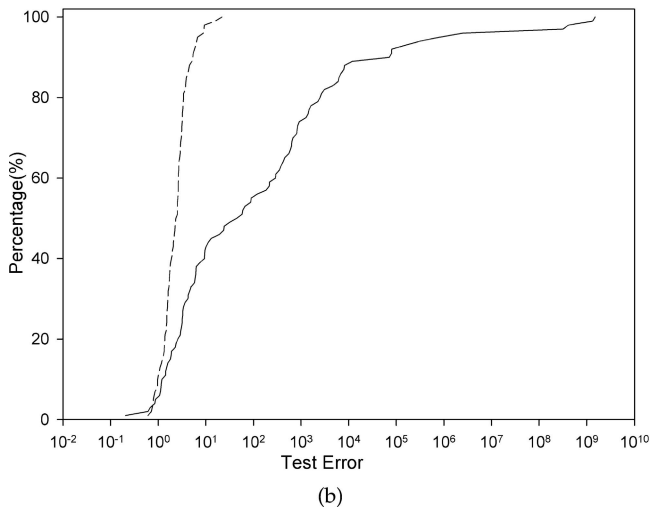
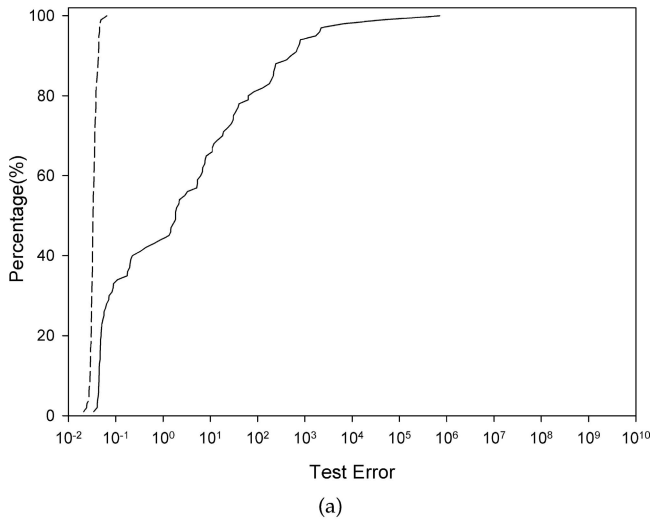


Fig. 3. Cumulative probability distributions of the test errors for (a) generational MOGP and function  $F5$  and (b) steady-state MOGP and function  $F6$ . The dashed lines are for trees with AQ operators and the solid lines are for trees with PD.

general AQ form as follows:

$$AQ(x_1, x_2) = \frac{x_1}{\sqrt{a + x_2^2}}.$$

We have investigated the effect of varying the value of  $a$  to see if it can be “tuned” to give better results. Broadly, as  $a \rightarrow 0$ , the distribution of test errors (unsurprisingly) tends to the same distribution as the (U)PD operator. For values of  $a$  larger than unity, the distribution of test errors becomes more compact, but its mean shifts to larger values. For  $|a| \gg \|x_2^2\|$ , the AQ tends to  $x_1$  divided by a large, positive constant which presumably lacks the expressive power of a division-type operator [4]. Typical data for generational SOGP are shown in Fig. 4 in the form of cumulative probability distributions, for clarity. A value of  $a = 1$  seems to be a good compromise for the data used in this paper; whether it might be beneficial to tune  $a$  for other datasets would need to be investigated on a case-by-case basis.

From a statistical point of view, the inputs to a GP model are invariably random variates. Considering two normally

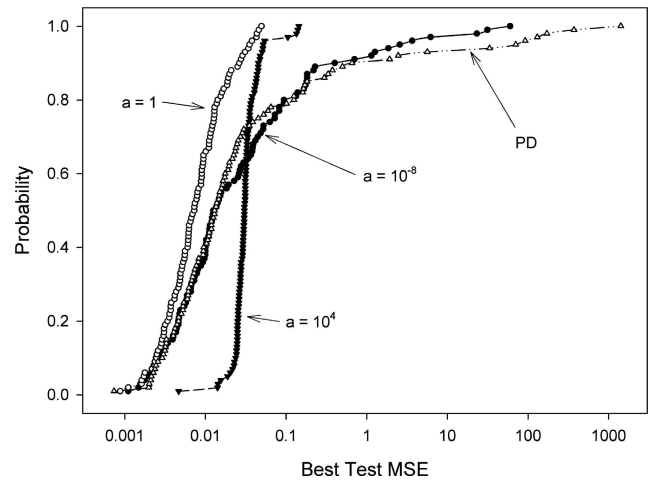


Fig. 4. Typical cumulative probability distributions of the test errors for varying values of  $a$ , generational single-objective GP.

TABLE XI  
MEAN NODE COUNTS FOR SINGLE-OBJECTIVE GENERATIONAL GP FOR DIFFERENT DIVISION OPERATORS

	AQ	PD	UPD
$F1$	$822.0 \pm 428.9$	$298.7 \pm 190.7$	$285.59 \pm 228.7$
$F2$	$1049 \pm 802.9$	$363.1 \pm 238.1$	$342.57 \pm 184.2$
$F3$	$775.8 \pm 361.1$	$292.8 \pm 211.4$	$304.84 \pm 231.2$
$F4$	$828.3 \pm 326.9$	$272.3 \pm 286.7$	$277.16 \pm 234.2$
$F5$	$739.8 \pm 301.6$	$348.4 \pm 223.3$	$315.12 \pm 190.1$
$F6$	$772.3 \pm 359.8$	$245.1 \pm 269.2$	$199.94 \pm 192.1$

TABLE XII  
MEAN NODE COUNTS FOR SINGLE-OBJECTIVE STEADY-STATE GP FOR DIFFERENT DIVISION OPERATORS

	AQ	PD	UPD
$F1$	$3347 \pm 2479$	$1464.31 \pm 951$	$1219 \pm 932.4$
$F2$	$3412 \pm 1970$	$1571.41 \pm 1104$	$1308 \pm 637.6$
$F3$	$2903 \pm 2055$	$1572.92 \pm 1061$	$1551 \pm 1096$
$F4$	$3542 \pm 1932$	$1483.17 \pm 1120$	$1373 \pm 960.4$
$F5$	$2550 \pm 1514$	$1371.68 \pm 899.0$	$1139 \pm 729.1$
$F6$	$2940 \pm 1587$	$1369.2 \pm 1288$	$1225 \pm 981.6$

distributed random variates,  $y_1$  and  $y_2$ , the quotient  $y_1/y_2$  is Cauchy-distributed [14] and thus has an unbounded variance and heavy tails extending to  $\pm\infty$ . (A value of  $\infty$  is, of course, the case where the denominator is zero.) PD removes the points at  $\pm\infty$ , but the very large values of the quotient  $< \infty$  remain and give rise to the unstable behavior described above.

Considering the proposed AQ, Fig. 5 shows the Q–Q plot [15] for the simulated AQ transformation under the assumption of variate normality. Apart from the few data which lie in the tails of the distribution, most of the data fall on the straight line indicating that  $AQ(y_1, y_2)$  is very close to normally distributed. Thus, the AQ stabilizes the variance of the intermediate values within the tree and consequently the overall tree response.

Finally, we have examined the distributions of tree sizes which result from using the AQ operator. Since the AQ function is somewhat smoother than (U)PD, we might expect the expressiveness of the AQ function to be lower leading to larger

TABLE XIII  
MEAN NODE COUNTS FOR MULTIOBJECTIVE GENERATIONAL GP FOR DIFFERENT DIVISION OPERATORS

	AQ	PD	UPD
F1	124.9 ± 74.42	86.85 ± 52.26	73.17 ± 47.52
F2	186.4 ± 77.05	96.32 ± 50.00	88.27 ± 42.78
F3	142.8 ± 65.52	61.66 ± 56.45	69.71 ± 49.34
F4	169.1 ± 71.49	42.19 ± 34.11	40.85 ± 38.17
F5	124.7 ± 66.39	86.28 ± 43.90	79.01 ± 43.01
F6	135.8 ± 78.06	59.01 ± 41.76	59.53 ± 44.95

TABLE XIV  
MEAN NODE COUNTS FOR MULTIOBJECTIVE STEADY-STATE GP FOR DIFFERENT DIVISION OPERATORS

	AQ	PD	UPD
F1	154.8 ± 98.13	75.05 ± 46.77	78.67 ± 52.14
F2	198.4 ± 91.94	108.52 ± 62.19	102.73 ± 60.53
F3	183.8 ± 81.04	72.55 ± 58.97	70.07 ± 54.02
F4	219.1 ± 118.49	58.17 ± 48.94	52.4 ± 44.64
F5	169.5 ± 80.48	96.62 ± 55.45	100.73 ± 47.00
F6	138.4 ± 69.25	78.97 ± 55.67	82.94 ± 57.90

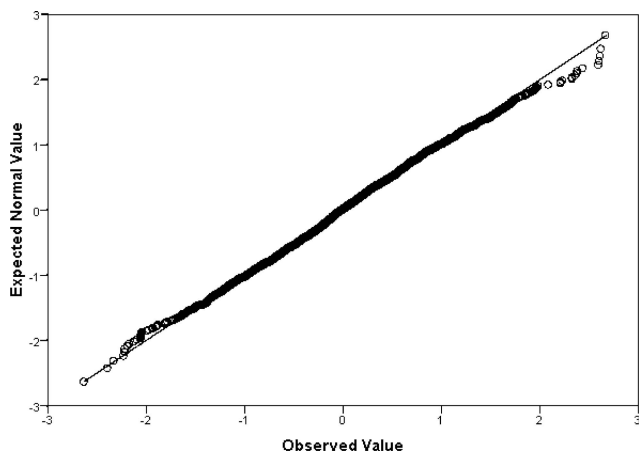


Fig. 5. Q-Q plot for samples transformed by the AQ.

tree sizes. The mean node counts for each of the experiments are shown in Tables XI–XIV. It is indeed the case that, within any given evolutionary setup, AQ produces larger trees. Interestingly, Šprogar [16] also observed a significant variation in the mean size of trees due to using different versions of the division operator. Comparing between evolutionary strategies, multiobjective GP seems to produce smaller trees than single-objective GP. Despite tending to produce larger trees, the AQ operator has the overriding advantage of delivering consistently and statistically significant smaller test errors.

V. CONCLUSION

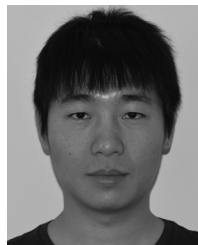
In this paper, we proposed an AQ as a replacement for the commonly used protected and unprotected division operations. The problems related to PD have previously been identified in the literature [4] and although these can be mitigated by using interval arithmetic, the fundamental problem of instability remains. We showed that the AQ produces statistically lower

mean test errors on a range of regression problems due to the elimination of the unstable fitted functions, which can result from the use of (U)PD. We also demonstrated the variance-stabilizing property of the AQ transformation. Further, this transformation is differentiable.

We thus proposed the AQ as a superior replacement for division in genetic programming.

REFERENCES

- [1] R. Poli, W. B. Langdon, and N. F. McPhee. (2008). *A Field Guide to Genetic Programming* [Online]. Available: <http://lulu.com> and <http://www.gp-field-guide.org.uk>
- [2] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.
- [3] *IEEE Standard for Binary Floating-Point Arithmetic*, IEEE Standard 754, 1985.
- [4] M. Keijzer, “Improving symbolic regression with interval arithmetic and linear scaling,” in *Proc. EuroGP*, 2003, pp. 70–82.
- [5] M. Castelli, L. Manzoni, S. Silva, and L. Vanneschi, “A quantitative study of learning and generalization in genetic programming,” in *Proc. 14th EuroGP*, 2011, pp. 25–36.
- [6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Berlin, Germany: Springer-Verlag, 2008.
- [7] G. Wahba and S. Wold, “A completely automatic French curve: Fitting spline functions by cross validation,” *Commun. Stat.*, vol. 4, no. 1, pp. 1–17, 1975.
- [8] E. J. Vladislavleva, G. F. Smits, and D. den Hertog, “Order of nonlinearity as a complexity measure for models generated by symbolic regression via Pareto genetic programming,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 333–349, Apr. 2009.
- [9] R. Kumar and P. I. Rockett, “Improved sampling of the Pareto-front in multiobjective genetic optimizations by steady-state evolution: A Pareto converging genetic algorithm,” *Evol. Comput.*, vol. 10, no. 3, pp. 283–314, 2002.
- [10] S. Silva and J. Almeida, “Dynamic maximum tree depth: A simple technique for avoiding bloat in tree-based GP,” in *Proc. GECCO*, 2003, pp. 1776–1787.
- [11] A. Ekárt and S. Z. Németh, “Selection based on the Pareto non-domination criterion for controlling code growth in genetic programming,” *Genet. Program. Evol. Mach.*, vol. 2, no. 1, pp. 61–73, Mar. 2001.
- [12] Y. Zhang and P. I. Rockett, “A generic optimizing feature extraction method using multiobjective genetic programming,” *Appl. Soft Comput.*, vol. 11, no. 1, pp. 1087–1097, 2011.
- [13] P. Sprent and N. C. Smeeton, *Applied Nonparametric Statistical Methods*, 4th ed. Boca Raton, FL: Chapman and Hall/CRC, 2007.
- [14] M. D. Springer, *The Algebra of Random Variables*. New York: Wiley, 1979.
- [15] NIST/SEMATECH. (2010). *e-Handbook of Statistical Methods* [Online]. Available: <http://www.itl.nist.gov/div898/handbook>
- [16] M. Šprogar, “A study of GP’s division operators for symbolic regression,” in *Proc. 7th ICMLA*, 2008, pp. 286–291.



Ji Ni received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2006, and the M.Sc. degree from the University of Sheffield, Sheffield, U.K., in 2009, where he is currently pursuing the Ph.D. degree with the Department of Electronic and Electrical Engineering, University of Sheffield.

His current research interests include model generalization in multiobjective genetic programming, focusing on function complexity and stability analysis.

Russ H. Driberg, photograph and biography not available at the time of publication.



**Peter I. Rockett** was born in London, England. He received the B.Sc. degree in electronic engineering in 1976, the M.Sc. degree in solid-state electronics in 1977, and the Ph.D. degree in semiconductor physics in 1980, all from the University of Manchester Institute of Science and Technology, Manchester, U.K.

After holding various fellowships and research positions, he was appointed to a tenured faculty position in electronic systems with the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K., in 1990. His current research interests include image processing, statistical pattern recognition, and multiobjective genetic programming.