An Empirical Study on the Accuracy of Computational Effort in Genetic Programming

David F. Barrero and María D. R-Moreno
Departamento de Automática
Universidad de Alcalá
Crta. Madrid-Barcelona Km. 33,6
Alcala de Henares, Madrid, Spain
Email: david,mdolores@aut.uah.es

Bonifacio Castaño
Departamento de Matemáticas
Universidad de Alcalá
Crta. Madrid-Barcelona Km. 33,6
Alcala de Henares, Madrid, Spain
Email: bonifacio.castano@uah.es

David Camacho
Departamento de Informática
Universidad Autónoma de Madrid
C/ Francisco Tomás y Valiente 11
Madrid, Spain
Email: david.camacho@uam.es

***Abstract*—Some commonly used performance measures in Genetic Programming are those defined by John Koza in his first book. These measures, mainly computational effort and number of individuals to be processed, estimate the performance of the algorithm as well as the difficulty of a problem. Although Koza's performance measures have been widely used in the literature, their behaviour is not well known. In this paper we study the accuracy of these measures and advance in the understanding of the factors that influence them. In order to achieve this goal, we report an empirical study that attempts to systematically measure the effects of two variability sources in the estimation of the number of individuals to be processed and the computational effort. The results obtained in those experiments suggests that these measures, in common experimental setups, and under certain circumstances, might have a high relative error.**

I. Introduction

Many different measures have been traditionally used by the Genetic Programming (GP) community. The intrinsic complexity of the behaviour of Evolutionary Algorithms has lead to a large number of measures that reflects the different sides of the phenomenon. Eiben and Smith distinguish two types of performance measures: effectivity and efficiency [1]. The former measures how good (or bad) is an algorithm finding an solution, while the latter deals with the cost of finding a solution, given that it was found.

Some popular performance measures widely used by GP researchers and practitioners were introduced by John Koza in the chapter 4 of his first book [2]. These metrics use an effectivity measure, the accumulated success probability, to generate two efficiency measures: the number of individuals to be processed and the computational effort, both closely related to each other. The *number of individuals to be processed*, or $I(M,i,z)$, is an estimation of how many individuals should be processed to obtain, at least, one success with a certain given probability z. The algorithm is suppsed to be run several times. $I(M,i,z)$ is given by the equation

$$I(M,i,z) = Mi \left\lceil \frac{ln(1-z)}{ln(1-P(M,i))} \right\rceil \quad (1)$$

where M is the population size and is supposed to remain constant along the execution of the algorithm; $i = 1, 2, ..., G$ is the generation number and is an independent variable; finally $P(M,i)$ is the accumulated success probability in generation i and is estimated as the ratio between the accumulated number of successful runs ($k(M,i)$) and the number of runs in the experiment (n), hence $P(M,i) = k(M,i)/n$. The operator $\lceil \ldots \rceil$ stands for the ceiling function, which rounds up its argument.

Since $I(M,i,z)$ is a function rather than a scalar, it is not well suitable to serve as a simple statistic. There is a simplification of $I(M,i,z)$ called *computational effort*, or E, which is simply the minimum value of $I(M,i,z)$, so

$$E = \min_i \left\{ Mi \left\lceil \frac{ln(1-z)}{ln(1-P(M,i))} \right\rceil \right\} \quad (2)$$

Many statistical issues arise from the difference between the definition of $I(M,i,z)$ and E and the estimation of those values that can be gathered empirically, $\hat{I}(M,i,z)$ and $\hat{E}$ [3]. This difference reduces the accuracy of these performance measures and, although this subject has been investigated before, we think that there is no understanding of the circumstances in which these measures are reliable.

To our knowledge, the first person noticing the statistical nature of computational effort was Angeline [4], he observed a remarkable variance in the measurement of E and suggested using statistical tools to manage the randomness. Some time after, Keijzer [5] bounded the estimation of with confidence intervals and observed that sometimes the width of the intervals are as large as the estimation of E [5]. Some studies followed, including the attempt made by Christensen *et al* to identify and characterize systematically the sources of variability in the measurement of Koza's performance measures [3]. He identified three sources of variability: the ceiling operator, the estimation error of the accumulated success probability and the minimum operator. Other works investigated how to use some statistical tools with computational effort, mainly confidence intervals [6], [7], toher authors studied the reliability [8], [9] of confidence intervals or proposed alternative performance measures [10].

This paper aims to quantify the error sources associated to the measurement of $I(M,i,z)$ as well as E from a pure empirical approach. We should emphasize that our goal is not to explain the error sources, but rather to identify and quantify

978-1-4244-7833-0/11/$26.00 ©2011 IEEE

the main factors that affect them. Inspired by Christensen, we systematically study two variability sources, the ceiling operator and the estimation error; however, in opposition to Christensen's work, we do not consider the minimum operator an error source, but the distinction between $I(M,i,z)$ and computational effort. Our initial hypothesis is that the error associated to $I(M,i,z)$ and E is affected by two and only two factors, the number of runs n, and the accumulated success probability $P(M,i)$. Nonetheless in this paper we only provide evidence in favor that n and $P(M,i)$ affects the quality of the measurement, the claim that only n and $P(M,i)$ affects the magnitude of the error is not demonstrated.

With these considerations, the paper has been structured as follows. We first perform an exploratory study. Section 3 begins the study of the error associated with $I(M,i)$ while section 4 studies the error of the computational effort. Both sections separates the study of how n and $P(M,i)$ affect the error. Finally, some conclusions and future work are outlined.

II. Overview of performance measures

There are two main problems concerning the experimentation that we have to face. First, since we are interested in the accuracy of the measures under study, there is a need to have something to compare with, to take as reference; ideally it should be the exact measure, but clearly it is not possible. Secondly, we need a high number of algorithm runs, with a high consumption of computing resources. These two problems can be solved using resampling methods.

Four classical GP study cases have been selected: Artificial ant with the Santa Fe trail, 6-multiplexer, even 4-parity and a linear regression [2]. They have been selected to represent a diversity of difficulties, from an easy problem (6-multiplexer) to a difficult one (4-parity), with two intermediate problems (artificial ant and regression). Each one of these domains was run a high number of times, $100,000$, with the exception of the 4-parity, that was only run $5,000$ times because its greater population size required more computational resources. The main advantage that it provides is that using all the runs it is possible to calculate an accurate estimation of the metrics under study. A second advantage is that once those runs are executed and stored, they can be resampled to avoid running again the algorithms, saving substancial computational resources and time.

The object of this study is not the algorithm itself, but rather the performance metrics, so the details of their implementation and the parameter tuning does not affect this study. Consequently, we have used the default implementation of the selected problems and parameters found in ECJ v18 [11], which are based on the original settings used by Koza [2]. The main parameters that we have used are reported in Table I, with just minimal corrections such as the population size[1].

The large number of runs executed yields a good estimation of the true values of Koza's metrics. Since they are the best estimation available for this study, we note them as $\hat{I}^{best}(M,i,z)$, $\hat{P}^{best}(M,i)$ and $\hat{E}^{best}$. The values of these estimations are shown in Table II.

[1]All the code, datasets and scripts needed to repeat the experiments are available in http://atc1.aut.uah.es/~david/cec2011

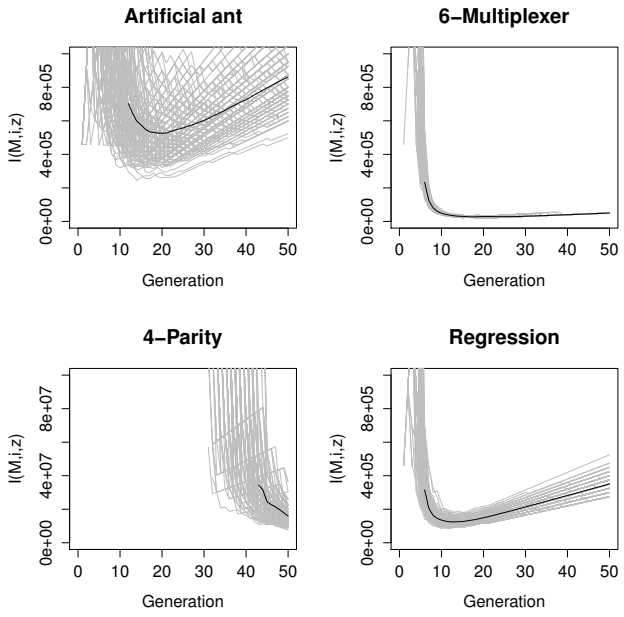

Fig. 1. Number of individuals to be processed of 200 pseudoexperiments composed by 50 runs. The mean value is plotted with a black solid line.

The variability of the estimation of $I(M,i,z)$ is depicted in Fig. 1. It contains the outcome of 200 simulated experiments (or pseudoexperiments) with its average value. Each experiment has been simulated taking 50 samples with replacement from the dataset. This figure shows that different pseudoexperiments usually yield different performance curves. Depending on the domain, the variability of the curves changes, for instance, if we compare the curves of the artificial ant and the 6-multiplexer, we find less variability in the latter than in the former. Notice that the scale used in the figure in both cases is the same.

At this point it makes sense for us to hypothesize that the problem difficulty plays a role, this hypothesis is based on the apparent correlation between the success rate of each problem and the dispersion of their $\hat{I}(M,i,z)$ curves. The two most difficult problems, the artificial ant and the 4-parity, are those with greater variability whereas the two easiest problems, 6-multiplexer and the regression, present less variability.

Fig. 2 shows the histograms of the computational effort calculated for the problem domains under study. Each histogram uses $5,000$ pseudoexperiments calculated using 50 (bottom row), 200 (middle row) and 500 (top row) runs sampled from the datasets of runs. Histograms do not suggest clearly a distribution function able to fit data in all the cases. Computational effort in the regression problem takes a triangular form while, for instance, the artificial ant seems to fit better in a lognormal or a Weibull distribution. There are also some outsider histograms, such us the parity problem

TABLE I

TABLEAU FOR THE PROBLEMS UNDER STUDY: ARTIFICIAL ANT WITH THE SANTA FE TRAIL, 6-MULTIPLEXER, EVEN 4-PARITY AND SYMBOLIC REGRESSION.

Parameter	Artificial ant	6-Multiplexer	4-Parity	Regression
Population	500	500	4,000	500
Generations	50	50	50	50
Terminal Set	Left, Right, Move, IfFoodAhead	A0, A1, A2, D0, D1, D2, D3, D4, D5	D0, D1, D2, D3, D4	X
Function set	Progn2, Progn3, Progn4	And, Or, Not, If	And, Or, Nand, Nor	Add, Mul, Sub, Div, Sin, Cos, Exp, Log
Success predicate	Best $fitness = 0$	Best $fitness = 0$	Best $fitness = 0$	Best$fitness \leq 0.001$
Initial depth	5	5	5	5
Max. depth	17	17	17	17
Selection	Tournament (size=7)	Tournament (size=7)	Tournament (size=7)	Tournament (size=7)
Crossover	0.9	0.9	0.9	0.9
Reproduction	0.1	0.1	0.1	0.1
Observations	Timesteps=600		Even parity	$y = x^4 + x^3 + x^2 + x$
	Santa Fe trail			$x \in [-1, 1]$

TABLE II

BEST ESTIMATION OF SUCCESS PROBABILITY FOR THE ARTIFICIAL ANT PROBLEM. IT REPORTS NUMBER OF RUNS (n), NUMBER OF SUCCESSFUL RUNS (k), BEST ESTIMATION OF SUCCESS RATE $\hat{P}^{best}(M, G)$, BEST ESTIMATION OF COMPUTATIONAL EFFORT ($\hat{E}^{best}$), BEST ESTIMATION OF COMPUTATIONAL EFFORT WITHOUT CEILING OPERATOR ($\hat{E}_c^{best}$) AND THEIR DIFFERENCE IN ABSOLUTE AS WELL AS RELATIVE VALUES.

	Artificial ant	6-Multiplexer	4-Parity	Regression
n	100,000	100,000	5,000	100,000
k	13,168	95,629	305	29,462
$\hat{P}^{best}(M, G)$	0.13168	0.95629	0.061	0.29462
$\hat{E}^{best}$	490,000	24,000	14,800,000	117,000
$\hat{E}_c^{best}$	487,276	22,805	14,633,571	116,468
Difference	2,724 (0.5%)	1,195 (4.98%)	166,429 (1.13%)	536 (0.49%)

for $n = 50$ or the multiplexer with $n = 50$, nonetheless, the latter can be explained by the grouping of the categories in the histogram.

The lack of an obvious distribution able to describe $\hat{E}$ confirms the previous result reported by Walker *et al* in [7], who also failed in finding a probability distribution able to model $\hat{E}$. We conjecture that there is an underlying random variable associated to the accumulated success probability, and this random variable is modified by several non-lineal operations such us logarithms and the minimum operator, so $\hat{E}$ in some sense follows the same distribution but it has been "contaminated" by those operations. From another point of view, differences in the distribution of $\hat{E}$ with different levels of n suggest the presence of a sampling bias [12], and thus the presence of other factors that influence E. We suspect these factor are the non-linear operations made by (1) and (2). More research is needed to provide evidence in favor or against this conjecture.

An important property of any estimator is its variability. Fig. 2 illustrates a relationship between the variability of the estimator and the number of runs: the higher is n, the narrower is the distribution of $\hat{E}$. Let us, for instance, observe the artificial ant, when $n = 50$ most of the estimators are placed between 0 and $1.5E6$ individuals, if we increase the number of runs to 200, most of $\hat{E}$ take values between $200,000$ and $800,000$; higher values of n yield even less variability of the estimations of computational effort, when $n = 500$ $\hat{E}$ is mostly placed in the range of $300,000$ and $700,000$. This behaviour is observed also in the rest of problem domains. Since the estimation of $I(M, i, z)$ and E depends on the estimation of the accumulated probability, and its quality is highly dependent on the number of runs [9], it makes sense to suppose that they are related to each other. This fact is analyzed in more detail in sections III-B and IV-B.

In any case, it is clear that performance measures contain a variability that comes from the intrinsic stochastic nature of experimentation. However the exact nature of the variability and the factors that influence its magnitude is not yet clear, so we move on to try to answer under which circumstances Koza's performance measures contain more variability and quantify it.

III. ACCURACY OF THE NUMBER OF INDIVIDUALS TO BE PROCESSED

We consider two sources of randomness in (1) associated to the estimation of the number of individuals to be processed: the ceiling operator and the estimation of the accumulative success probability. Christensen identified a third source of variability in the minimum operator, nevertheless, in our opinion, the effects of this operator should be studied in the context of computational effort because the minimum operator is strictly associated to this metric. So far the minimum operator is not included in the study of the number of individuals and is included in the study of computational effort done in section IV.

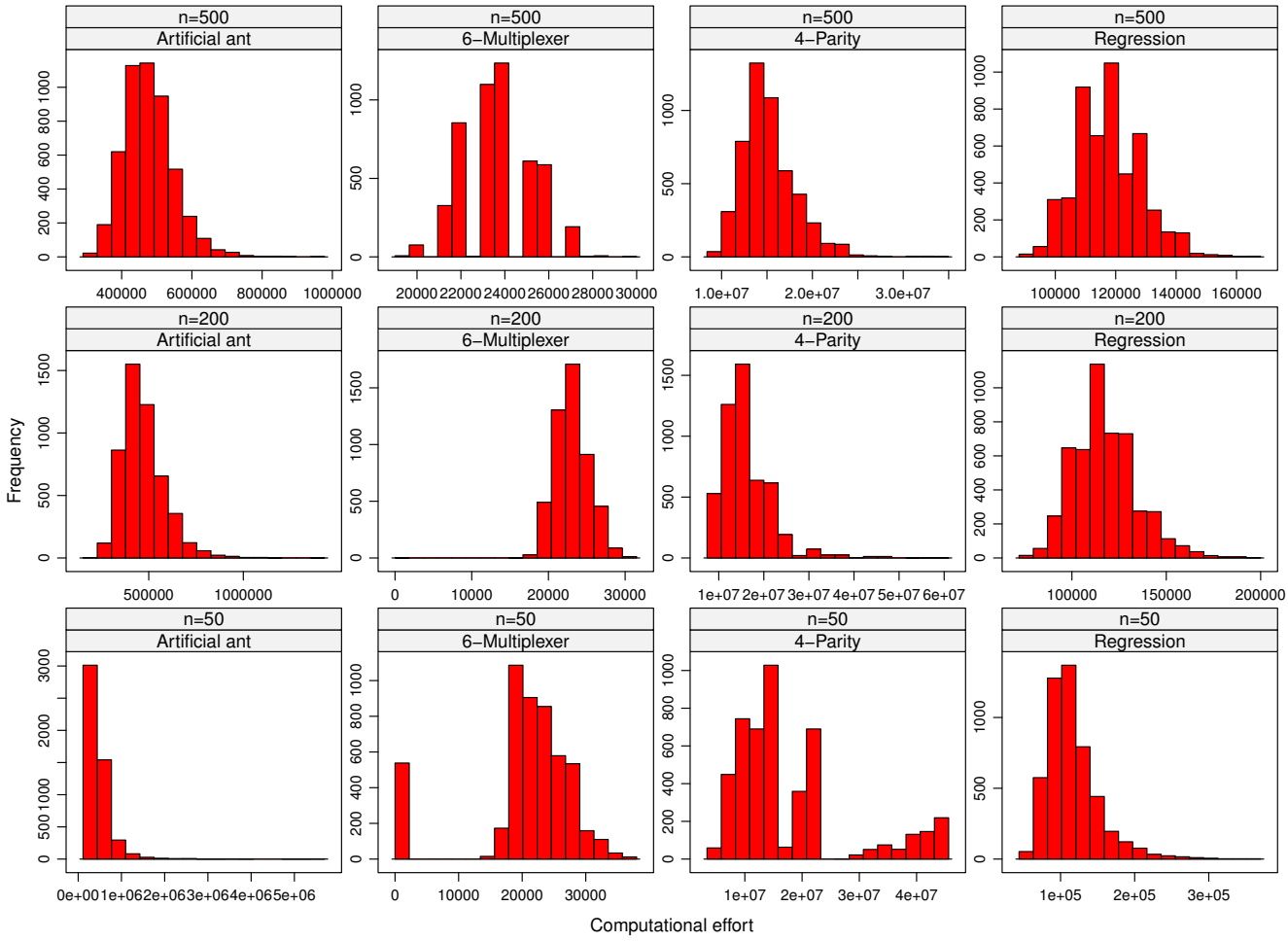

Fig. 2. Histograms of the computational effort for the four problems under consideration. Each histogram represents the computational effort of $5,000$ experiments that were simulated subsampling 50, 200 and 500 runs from the datasets.

A. *Ceiling error in* $\hat{I}(M,i,z)$

We begin the empirical study looking at the ceiling error. Strictly speaking, the ceiling operator is not a randomness source because it is a deterministic operator, but it removes information yielding discontinuities that increase the variability of the estimation, and thus introducing an error in practical terms.

In order to study the effects of the ceiling operator in the estimation of $I(M,i,z)$, we define a new measure $I_c(M,i,z)$ such as

$$I_c(M,i,z) = Mi\frac{ln(1-z)}{ln(1-P(M,i))} \tag{3}$$

which is (1) without the ceiling operator. Using (3) makes the estimation of the ceiling error straightforward, it is just the difference $I(M,i,z) - I_c(M,i,z)$.

In this way we can measure the ceiling error just comparing the number of processed individuals with and without the ceiling operator. We have performed this comparison using all the runs in the datasets and the results are shown in Fig. 3. This figure represents the best estimation of the number of individuals to be processed with ($\hat{I}^{best}(M,i,z)$) and without ($\hat{I}_c^{best}(M,i,z)$) ceiling operator. The most obvious difference is the sawtooth shape that $\hat{I}^{best}(M,i,z)$ has in some problem domains, such as the multiplexer. This shape is also found in the rest of the problems, nonetheless in different magnitude. In the case of the parity problem it seems that there are no discontinuities, however there are, but they so reduced that only a zoom over the figure shows it. In any case, $\hat{I}^{best}(M,i,z)$ is strictly higher than $\hat{I}_c^{best}(M,i,z)$, so, as Christensen reported, the ceiling error is biased and tends to increase the value of $I^{best}(M,i,z)$.

Interestingly, there seems to be a correlation between the problem difficulty and the magnitude of the discontinuity; the ceiling operator introduces more discontinuities in the multiplexer problem ($\hat{P}_{best}(M,G) = 0.96$), followed by the regression ($\hat{P}_{best}(M,G) = 0.29$), artificial ant ($\hat{P}_{best}(M,G) = 0.13$) and finally the parity problem ($\hat{P}_{best}(M,G) = 0.06$). This experiment confirms the relationship between the ceiling error and the problem difficulty found by Christensen and Oppacher using a synthetic expression of $P(M,i)$ [3]. Ex-

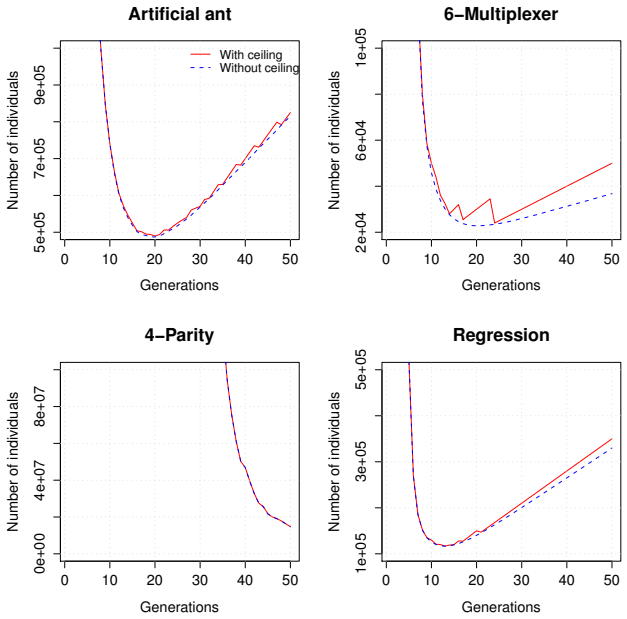

Fig. 3. Comparison between the number of individuals to be processed calculated using ceiling operator (solid red line) and not using it (dashed blue line). $I(M,i,z)$ curves have been calculated using all the samples in the dataset and setting $z = 0.99$.

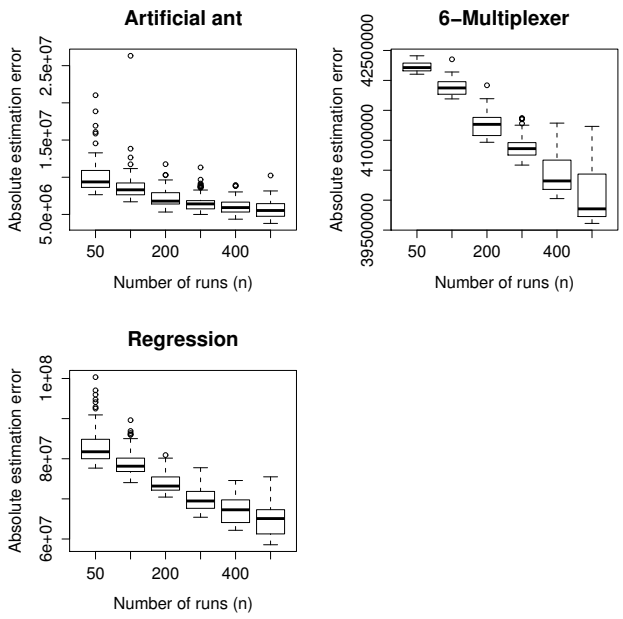

Fig. 4. Boxplot of the absolute estimation error of $\hat{I}(M,i,z)$ with several values of number of runs. Each box represents the sum of the average estimation error of 5.000 pseudoexperiments.

periments show that measuring $I(M,i,z)$ in easy problems tends to have more ceiling error than in hard problems.

Despite the potentially high impact that the ceiling operator might have in the estimation, there is an easy solution, just removing the operator. Koza introduced this operator to reflect that it is not possible to carry out a fractional number of experiments [2], however it is actually not supposed to be interpreted physically, so the ceiling error can be removed without any evident drawback. Nonetheless, the another source of variability under study, the estimation error, is intrinsic to the measure and thus cannot be removed.

B. Estimation error in $\hat{I}(M,i,z)$

If we look in more detail (1), we can identify two fixed parameters, M and z, and one independent variable, i. All these values are known, and thus they do not generate uncertainty. Usually, the only element in (1) that is not perfectly known is $P(M,i)$, that is an unknown probability and must be estimated empirically. The error associated to the estimation of $P(M,i)$ is actually the only true source of error since this is the only element in (1) that introduces uncertainty.

$\hat{P}(M,i)$ is the estimation of a probability, and, if we do not consider its variation in time, this probability in a fixed generation i_0 can be described using a binomial distribution, which is a well known problem [13]. Irrespective of the problem under study, the quality of the estimation of any success probability only depends on the number of trials (or runs in our case) and the magnitude of the probability [13], this result let us limit our study to only those factors.

We begin investigating the influence of the number of runs with the following experiment. Given the four datasets, we have calculated $5,000$ values of $\hat{I}_c(M,i,z)$ using n runs resampling with replacement from each dataset. The ceiling function is removed to isolate the effects of the estimation error. For each value of $\hat{I}_c(M,i,z)$, its distance to $\hat{I}_c^{best}(M,i,z)$ has been calculated using the following formula

$$\overline{\xi} = \sum_{j=0}^{R} \sum_{i=0}^{G} \frac{\hat{I}_c^{best}(M,i,z) - \hat{I}_c^{j}(M,i,z)}{R} \tag{4}$$

where $\overline{\xi}$ is the statistic that measures the average distance between $\hat{I}_c^{best}(M,i,z)$ and $\hat{I}_c^{j}(M,i,z)$, which is the j^{th} curve of the number of individuals to be processed. R the number of pseudo experiments. All the experiments were carried out with $R = 5,000$ and $G = 50$. Of course, it is an error measure and therefore low values means good estimations.

The boxplots of the estimation error calculated using the method described earlier are depicted in Fig. 4. A glance to this figure clearly suggests a strong relationship between the number of runs and the average estimation error, more runs yield better estimations of $I(M,i,z)$. The estimation error of the 4-parity problem is not shown because it was found that the low number of generations where $I(M,i,z)$ is defined (see Fig. 1) generates an erratic behaviour of the statistic we use to measure the distance.

Experimentation with the other factor under study, the accumulated success probability, is more tricky. $P(M,i)$ is not an independent variable, but a dependent one and, unless we used a synthetic $P(M,i)$, we cannot manipulate it to carry out the experiment. Additionally, $P(M,i)$ is a function rather than a scalar. These two facts difficult experimentation,

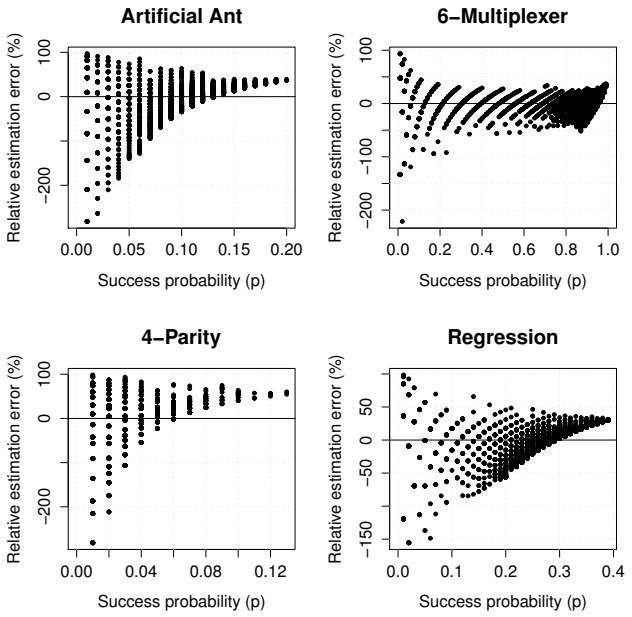

Fig. 5. Scatterplot of the relative estimation error of $I(M,i,z)$ with several values of success probability. 200 pseudoexperiments with $n = 100$ were carried out for each problem domain.

however, we can still perform an experiment to observe the behaviour of the estimation error for different values of the accumulated success probability. For each problem domain, we have run 200 pseudoexperiments with $n = 100$ following the same procedure described above, but we have done a different manipulation of data. Instead of measure how close is $\hat{I}_c^{best}(M,i,z)$ from $\hat{I}_c(M,i,z)$, we have stored the tuple $(P(M,i), \varepsilon_{est}(i))$, where $i = 1, ..., G$ and

$$\varepsilon_{est}(i) = 100\frac{\hat{I}_c^{best}(M,i,z) - \hat{I}_c(M,i,z)}{\hat{I}_c^{best}(M,i,z)}$$

is the relative estimation error. In this way we obtain $G = 50$ tuples from each pseudorun, and we used 200, so there are $10,000$ tuples in each problem domain.

The tuples that we have obtained are shown in the scatterplot depicted in Fig. 5. This figure shows a surprising behaviour of the estimation error: it is not symmetrical and it is biased. Overestimating $P(M,i)$ yields an underestimation of $I(M,i,z)$, on the contrary, an overestimation of $P(M,i)$ generates an underestimation of $I(M,i,z)$. Fig. 5 shows that the effects of overestimating or underestimating $P(M,i)$ are not the same. An overestimation of $P(M,i)$ induces a higher error in $I(M,i,z)$ than a underestimation, it is specially notorious in the case of the artificial ant and the 4-parity problems. This asymmetry varies with the success probability, while the minimum error tends to reduce with the probability, the maximum error is almost constant. In any case, there is an an asymptotic behaviour of the estimation error with very low success probability that makes the estimation highly imprecise in that region.

The magnitude of the maximum estimation error depends on the success probability. Low probabilities yield higher estimation error and higher success probabilities tend to generate less estimation error. Nonetheless the error is biased in the end of the execution of the algorithm (higher success rates), with the only exception of the multiplexer, which is the only one that achieve a success rate close to 1. It leads us to conjecture that high success probabilities have associated higher estimation error, however we feel unable to claim it with the evidence shown, it should be confirmed by further research. In any case, the magnitude of the bias seems to be rather significant in almost all the cases, around 30% and 50%, with the exception of the 6-multiplexer. We should remark that this experiment used 100 runs, which is a relatively high number of runs; it is quite easy to find literature that reports experiments with fewer number of runs, so we can expect that the estimation error in those experiments were higher.

In average, the relative estimation error is notable and the estimator is biased significantly, depending on the problem domain. Nonetheless, these error might be, or not, significant when the computational effort is calculated, which is the objective of the next subsection.

IV. Accuracy of computational effort

Common sense suggests that a good estimation of $I(M,i,z)$ should also yield a good estimation of the computational effort; this apparent correlation should link the factors of $I(M,i,z)$ with the factors of E. However, common sense might fail, therefore we have performed some experiments to verify this hypothesis. We should point out that in this section we only study one factor, the number of runs. There are reasons to think that the magnitude of the accumulated success probability plays an important role, however we must face that this probability is not fixed with the generation time and it is not an independent variable. Moreover, the variation of $P(M,i)$ plays an essential role in the measurement of computational effort, and it is not possible to treat it as a punctual estimator, like we did in the previous section. For these reasons, in the following, this factor is excluded from the study.

A. Ceiling error in E

Firstly it is worth to compare the computational effort with and without ceiling operator when it is calculated using all the samples. These values, as well as their absolute and relative difference, can be found in Table II. We found earlier that easy problems -those with high success probability- generated more ceiling error in the estimation of $I(M,i,z)$. Table II shows that our experiments partially verify this behaviour in the estimation of E. The easiest problem, the 6-multiplexer ($\hat{P}_{best}(M,G) = 0.96$), generated the biggest difference between $\hat{E}^{best}$ and $\hat{E}_c^{best}$, 4.98%, while the rest of the domains achieve intermediate values of ceiling error: the artificial ant ($\hat{P}_{best}(M,G) = 0.13$) with a difference of 0.5%, the 4-parity ($\hat{P}_{best}(M,G) = 0.06$) with 1.13% and finally the regression problem ($\hat{P}_{best}(M,G) = 0.29$) with 0.49%.

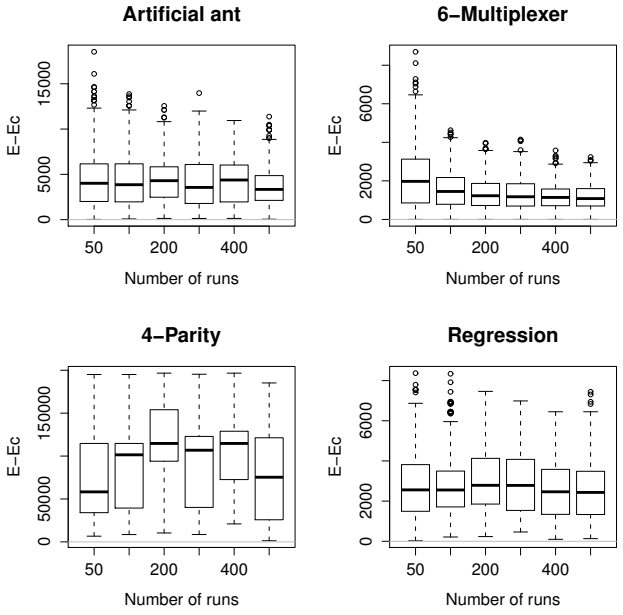

Fig. 6. Absolute ceiling error of computational effort with several number of runs. Each box represents $2,000$ pseudoexperiments.

There is no direct correlation between problem difficulty and ceiling error when estimating E. There may be two possible explanations behind this fact. First, the ceiling operator introduces discontinuities in $I(M,i,z)$ that might increase variance when the minimum is calculated. From another perspective, we observe that the 4-parity is the hardest problem but it has the second highest ceiling error. There is an important issue with this problem domain, as can be seen in Fig 1: the number of generations given to the algorithm is too scarce, so it could have affected the result of this experiment. So far, there seems to be a tight correlation between the success probability of a problem and the ceiling error associated to $\hat{E}$. As we did earlier, we pass though to study whether the number of runs influences the ceiling error.

Fig. 6 shows a boxplot that represents the difference $\hat{E}-\hat{E}_c$ of $2,000$ pseudoexperiments calculated with different values of n. The use of $\hat{E}^{best}$ has been avoided to isolate the ceiling error from the estimation error. We first observe that the difference is always positive, meaning that $E > E_c$, which is not surprising because the ceiling operator always increases its argument, unless it were an integer, which is rather unlikely. No notable differences in the mean value of the difference $\hat{E}-\hat{E}_c$ are appreciated, only when n is small, around 50 runs, the tail of the distribution seems to be longer, with more outsiders, but the median, as well as the first and third quantiles, remains almost constant, regardless of the number of runs.

This result is confirmed with a one-way ANOVA test, whose result is shown in Table III. The ANOVA was calculated for six levels of n (50, 100, 200, 300, 400 and 500) using the square root of $\hat{E}-\hat{E}_c$ as independent variable. Using 50 pseudoexperiments for each level, two problems (multiplexer and parity) yielded statistical significance with $\alpha = 0.01$ while two did not (artificial ant and regression). However, the residuals of the multiplexer and the parity problems did not pass the normality test, and therefore we cannot accept their test as valid. The residuals of the other two problems did pass the normality test, which are the two that did not found differences, so, with these evidence, we conclude that the number of runs does not affect the ceiling error when estimating computational effort.

TABLE III
ANALYSIS OF VARIANCE FOR SIX LEVELS OF FACTOR n, THE INDEPENDENT VARIABLE IS THE SQUARE ROOT OF THE DIFFERENCE $E_c - E$. RESIDUALS OF PROBLEMS MARKED WITH * DID NOT PASS THE NORMALITY TEST. P-VALUES WITH SIGNIFICANCE ($\alpha = 0.01$) ARE MARKED IN BOLD.

Problem	df	Sum. sq.	Mean sq.	F-value	p-value
Artificial ant	5	2445	489.03	0.9689	0.437
6-Multiplexer	5	5145	1029	5.1462	**0.0001529***
4-Parity	5	236380	47276	4.9595	**0.0002247***
Regression	5	4349	869.79	2.9602	0.01266

Experiments shown in this subsection were designed to avoid the effects of the estimation error, which is just the factor that we move forward to study.

B. Estimation error in E

Finally, we study the effects of the estimation error. This study follows a procedure similar to the one used previously. Given the datasets of the four selected problem domains, 100 experiments were simulated resampling n runs with replacement from the datasets. For each simulated experiment, the error between the estimation and the best estimation of computational effort was calculated. Two methods to calculate computational effort were used, using the ceiling operator and not. In this way, we are able to measure the estimation error as well as compare both methods of calculating computational effort, so the statistic of relative estimation error is given by

$$\varepsilon_{est}^{E}(\%) = \frac{E_{best} - E}{E_{best}};\ \varepsilon_{est}^{E_c}(\%) = \frac{E_c^{best} - E_c}{E_c^{best}}$$

The variation of the estimation error with n is shown in Fig. 7. It shows some interesting behaviors. Probably, the most important one from a practical point of view is the high relative estimation error found in our experiments. Depending on the problem, when the number of runs is not too high, an estimation error of computational effort up to 50% is found. Error decreases rapidly with the number of runs, however there is a point that a small reduction of the error requires a very remarkable increment of the number of runs. Depending on the context, incrementing the number of runs might not pay off.

Another interesting property that Fig. 7 shows is the asymmetry of the estimation error. It was previously shown that estimation error of $I(M,i,z)$ is asymmetrical and we can observe now that this behaviour is transferred to the estimation error of E. The maximum overestimation of E is bounded

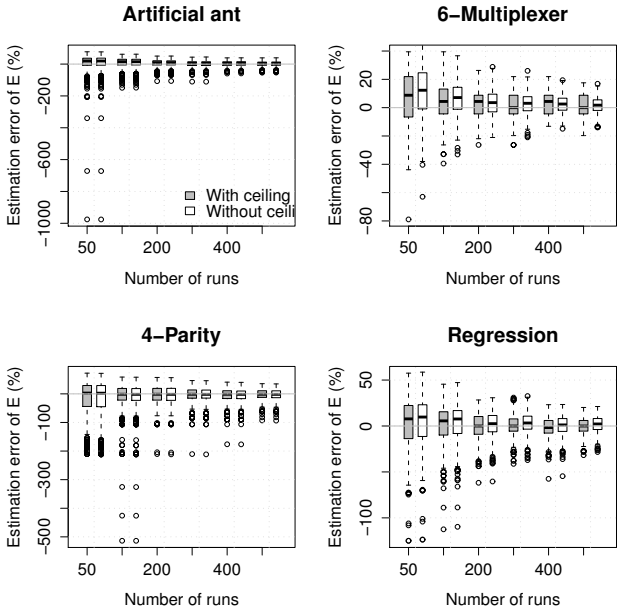

Fig. 7. Estimation error of computational effort with several values of number of runs. Each box represents 100 pseudoexperiments.

TABLE IV
ANALYSIS OF VARIANCE FOR SIX LEVELS OF FACTOR n, THE INDEPENDENT VARIABLE IS THE SQUARE ROOT OF THE ESTIMATION ERROR OF E_c. RESIDUALS OF PROBLEMS MARKED WITH * DID NOT PASS THE NORMALITY TEST. P-VALUES WITH SIGNIFICANCE ($\alpha = 0.01$) ARE MARKED IN BOLD.

Problem	df	Sum sq	Mean sq	F value	p-value
Artificial Ant	5	126.44	25.2874	12.579	**1.035e-10**
6-Multiplexer	5	95.09	19.0180	14.654	**3.272e-12**
4-Parity	5	98.72	19.7441	8.4623	**4.308e-07***
Regression	5	106.12	21.2248	12.264	**3.414e-10**

and it tends to reduce its value as n increases. Unfortunately, when E is underestimated, it tends to produce much higher errors, nonetheless this difference tends to disappear when the number of runs is increased. Finally, the ceiling operator does not seem to influence the estimation error, the distribution of the estimation error with and without ceiling operator is similar, with the only exception of the 6-multiplexer, which is also the most sensitive problem to the ceiling operator.

Although the variation of the estimation error shown in Fig. 7 is rather clear, it is better support this conclusion with a statistical test. We performed a one-way ANOVA of the square root of the estimation error for the six levels of n previously shown, the result can be seen in Table IV. The test found differences in the levels of the factor for the four problems using a significance level $\alpha = 0.01$, however one problem, the 4-parity, did not pass the normality test of its residues.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have carried out an empirical study about the accuracy of Koza's performance measures. Two sources of variability were analyzed, the ceiling operator and the estimation of the accumulated success probability. The former introduces an error that does not depend on the number of runs but varies with $P(M, i)$. Our experiments showed that this operator may introduce up to 50% of variability in the measure of $I(M, i, z)$ in an easy problem, nevertheless this operator may be removed without any evident drawback. The estimation of $P(M, i)$ introduces an intrinsic error that cannot be removed, just reduced increasing the number of runs. The experiments reported in this paper makes us doubt about the reliability of $I(M, i, z)$ and E under certain circumstances. A natural step is try to understand why the error sources have the behaviour that we observed and provide an analitycal estimation of the error.

ACKNOWLEDGMENT

This work was partially supported by the MICYT project ABANT (TIN2010-19872) and Castilla-La Mancha project PEII09- 0266-6640.

REFERENCES

[1] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Springer-Verlag, 2009, ch. Working with Evolutionary Algorithms, pp. 241–258.

[2] J. Koza, *Genetic Programming: On the programming of Computers by Means of Natural Selection*. Cambrige, MA: MIT Press, 1992.

[3] S. Christensen and F. Oppacher, "An analysis of koza's computational effort statistic for genetic programming," in *EuroGP '02: Proceedings of the 5th European Conference on Genetic Programming*. London, UK: Springer-Verlag, 2002, pp. 182–191.

[4] P. J. Angeline, "An investigation into the sensitivity of genetic programming to the frequency of leaf selection during subtree crossover," in *GECCO '96: Proceedings of the First Annual Conference on Genetic Programming*. Cambridge, MA, USA: MIT Press, 1996, pp. 21–29.

[5] M. Keijzer, V. Babovic, C. Ryan, M. O'Neill, and M. Cattolico, "Adaptive logic programming," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*. San Francisco, California, USA: Morgan Kaufmann, 7-11 Jul. 2001, pp. 42–49.

[6] M. Walker, H. Edwards, and C. H. Messom, "Confidence intervals for computational effort comparisons," in *EuroGP*, 2007, pp. 23–32.

[7] M. Walker, H. Edwards, and C. Messom, "The reliability of confidence intervals for computational effort comparisons," in *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2007, pp. 1716–1723.

[8] J. Niehaus and W. Banzhaf, "More on computational effort statistics for genetic programming," in *Genetic Programming, Proceedings of EuroGP'2003*, ser. LNCS, C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, and E. Costa, Eds., vol. 2610. Essex: Springer-Verlag, apr 2003, pp. 164–172.

[9] D. F. Barrero, D. Camacho, and M. D. R-Moreno, "Confidence intervals of success rates in evolutionary computation," in *GECCO '10: Proceedings of the 12th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2010, pp. 975–976.

[10] M. Walker, H. Edwards, and C. H. Messom, "Success effort and other statistics for performance comparisons in genetic programming," in *IEEE Congress on Evolutionary Computation (CEC 2007)*, Singapore, 2007, pp. 4631–4638.

[11] "A Java-based Evolutionary Computation Research System (ECJ Libraries) home page," http://cs.gmu.edu/˜eclab/projects/ecj/.

[12] P. R. Cohen, *Empirical methods for artificial intelligence*. Cambridge, MA, USA: MIT Press, 1995.

[13] L. D. Brown, T. T. Cai, and A. Dasgupta, "Interval estimation for a binomial proportion," *Statistical Science*, vol. 16, pp. 101–133, 2001.