Analyzing Fault Tolerance on Parallel Genetic Programming by Means of Dynamic-Size Populations

Daniel Lombraña González
University of Extremadura
Mérida, Badajoz, Spain
e-mail: daniellg@unex.es

Francisco Fernández de Vega
University of Extremadura
Mérida, Badajoz, Spain
e-mail: fcofdez@unex.es

Abstract—This paper presents an experimental research on the size of individuals when dynamic size populations are employed with Genetic Programming (GP). By analyzing the individual’s size evolution, some ideas are presented for reducing the length of the best individual while also improving the quality. This research has been performed studying both individual’s size and quality of solutions, considering the fixed-size populations and also dynamic size by means of the plague operator. We propose an improvement to the Plague operator, that we have called Random Plague, that positively affects the quality of solutions and also influences the individuals’ size. The results are then considered from a quite different point of view, the presence of processors failures when parallel execution over distributed computing environments are employed. We show that results strongly encourage the use of Parallel GP on non fault-tolerant computing resources: experiments shows the fault tolerant nature of Parallel GP.

I. I NTRODUCTION

The growth of individuals in GP is considered one of the main drawbacks when applying the technique to optimization problem. When generations are computed the individuals increase their size (see for instance [2], [20], [8], [5]).

Given the inherent difficulty of many real-life problems, which require the evaluation of a large number of individuals for many generations, and the length’s growth of individuals occurring within populations, large computing resources and time are usually required. Therefore, researchers have tried to offer different techniques aimed at alleviating that problem.

One of the more recent approaches trying to fight computing effort by reducing population size was described first by Fernández, and consisted of employing dynamic size populations by using the plague operator (see [8]). The operator was inspired by the behavior of biological plagues in nature. Plagues work by eliminating a set of individuals, the worst ones. Although other authors have described similar ideas considering resources available in the population [23], the relationship between population-size and individual-size, to our best knowledge has not been studied yet when dynamic-size populations are considered.

This paper presents a detailed study of individuals’ size evolution, analyzing the size increase and studying how this increase is affected by plagues. We will also show how the quality of solutions can be slightly improved. Nevertheless and even more important, we will show that this previous study when applied to Parallel Genetic Programming is crucial: although a distributed computing environment may be not reliable, Parallel GP can circumvent faults and therefore provide fault tolerance.

II. D YNAMIC SIZE POPULATIONS AND THE PLAGUE OPERATOR

There are different studies dealing with individuals growth and their relationship with the quality of solutions (see [14], [21], [3], [15], [17]). One of the consequences of the growth phenomenon is that individuals complexity increases, making their evaluation very expensive in terms of memory and computing time.

As described above, the idea of employing a progressive reduction in population size in GP was first described in [8] and [10]. The idea was to control the global increment of population size by reducing progressively the number of individuals within the population as generations were computed. The idea was quite different than the traditional control of size at the individual level for avoiding Bloat. Similar ideas were later employed by other researchers (see for instance [18]).

Plague has been described as a new genetic operator that could be used every generation for removing a number of bad individuals, those featuring worst fitness values. If large values are employed, the population may get empty very quickly. Therefore, a good and frequency value will be an important choice when applying the plague operator.

Plague has shown worthy for reducing computing effort required for finding solutions not only for GP but also for GAs [4].
I. Introduction

Nevertheless, although a number of studies considering dynamic size populations have been published since then (see [16], [9], [19]), to our best knowledge, a deep study analyzing the evolution of every individual size in the population and its relationship with population size has not been yet attempted.

Our research thus focuses on the size of individuals and their evolution along generations when some individuals of the population, the worst ones, are selectively removed (which is actually performed by the plague operation). We are interested in studying whether this action, besides reducing the global size of the population, has some impact on the size of individuals within the population. We will see later how this study sheds light on the fault tolerance nature of Parallel GP and its capability for circumventing faults that may happen when running on a parallel or distributed system.

II. Experiments and Results

In this research, we have worked with the well-known Even Parity 5 (EP5) problem. This problem has been traditionally employed as a difficult benchmark for GP [13]. The problem treats to build a program capable of calculating the parity of a set of 5 bits. We have computed the fitness value as the number of errors over the 32 available cases. Therefore, when we get a fitness near to zero we will have a better solution. Another set of experiments have been performed with the Symbolic Linear Regression (SLR) problem [13]. The problem tries to find the Function (1), and we measure the quality of solutions as the error computed by a given evolved polynomial when compared with the previous function within the range [−1, 1].

\[ y = x^4 + x^3 + x^2 + x \] (1)

Each of the figures that are shown below are the average of 50 independent executions for every experiment. The figures show the size or fitness of individuals along generations.

In order to study the evolution of individuals we have stored each of the features of every individual in the population every generation, so that we can establish a ranking of individuals within graphs according to their fitness values (actually every individual value –a point in the graph– is the average over 50 runs, so that the best-individual fitness or length within a graph, for instance, is thus the average of the 50 best-individuals at a given generation from the 50 runs).

Most of the graphs presented in this paper are 2D gray-color maps that represent, –using a color box–, the length-fitness of individuals. In all the graphs, every individual is sorted according to their fitness value; we will thus have at zero position the best individual and latter positions will represent worse fitness values.

A given individual that subsist for a number of generation will not occupy the same rank at different generations: their fitness-position will depend on the rest of individuals quality every generation. The importance of this ranking of individuals is that we can analyze the relationship between the relative quality of individuals within the population and their length.

III. Experiments and Results

In this research, we have worked with the well-known Even Parity 5 (EP5) problem. This problem has been traditionally employed as a difficult benchmark for GP [13]. The problem treats to build a program capable of calculating the parity of a set of 5 bits. We have computed the fitness value as the number of errors over the 32 available cases. Therefore, when we get a fitness near to zero we will have a better solution. Another set of experiments have been performed with the Symbolic Linear Regression (SLR) problem [13]. The problem tries to find the Function (1), and we measure the quality of solutions as the error computed by a given evolved polynomial when compared with the previous function within the range [−1, 1].

\[ y = x^4 + x^3 + x^2 + x \] (1)

Each of the figures that are shown below are the average of 50 independent executions for every experiment. The figures show the size or fitness of individuals along generations.

In order to study the evolution of individuals we have stored each of the features of every individual in the population every generation, so that we can establish a ranking of individuals within graphs according to their fitness values (actually every individual value –a point in the graph– is the average over 50 runs, so that the best-individual fitness or length within a graph, for instance, is thus the average of the 50 best-individuals at a given generation from the 50 runs).

Most of the graphs presented in this paper are 2D gray-color maps that represent, –using a color box–, the length-fitness of individuals. In all the graphs, every individual is sorted according to their fitness value; we will thus have at zero position the best individual and latter positions will represent worse fitness values.

A given individual that subsist for a number of generation will not occupy the same rank at different generations: their fitness-position will depend on the rest of individuals quality every generation. The importance of this ranking of individuals is that we can analyze the relationship between the relative quality of individuals within the population and their length.

IV. Analysis of Individuals Length

All the experiments have the same set up of basic parameters described in the Table I. As explained above, fitness will represent individual’s error when solving the problem, (zero will be the best possible value).

Two versions of every experiment has been run: the Classic Algorithm and the Classic Algorithm with plague. The idea is to compare evolution of individuals’ size when plagues act and when they are not present. We have to take into account, that this analysis have been performed while simultaneously applying a maximum depth for individuals (see Table I), which can be considered the standard for GP.

A. Analysis of Individuals Length

Fig. 1(a) shows the results obtained for the EP5 problem with a population of 1000 individuals, and executing the experiment for 100 generations with the classic GP algorithm. The figure shows each of the individuals’ size (average of 50 runs) ordered according to their fitness values (individual 0 being the best one). We can observe how individuals grow (from dark side to the brighter one) every generation until the end of the execution. Although the graph doesn’t show explicitly fitness values, and shows instead sizes, this way of showing data allows us to understand the relationship between fitness, size and evolution time. Fig. 1(b) shows the results for the SLR problem with the classic algorithm. All the parameters that we have used for this problem are described in the Table I(b) (Protected / and log operations). Similar conclusions can be drawn.

We may notice how individuals that grow more are habitually those with the best fitness values: the figure clearly shows a brighter color towards the 0 individual and the last generation: better individuals are the ones that possess a larger number of nodes. We observe that the global trend is that larger fitness value are related to larger increases in

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initsize</td>
<td>Yes</td>
</tr>
<tr>
<td>Crossover Probability</td>
<td>0.90</td>
</tr>
<tr>
<td>Reproduction Probability</td>
<td>0.10</td>
</tr>
<tr>
<td>Mutation Probability</td>
<td>0.33</td>
</tr>
<tr>
<td>Selection</td>
<td>Fitness Proportional</td>
</tr>
<tr>
<td>Functions</td>
<td>∧, ∨, /, NOR, NAND</td>
</tr>
<tr>
<td>Terminals</td>
<td>D0, D1, D2, D3, D4</td>
</tr>
<tr>
<td>Max Depth in Cross</td>
<td>15</td>
</tr>
<tr>
<td>Max Depth in Mutation</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initsize</td>
<td>Yes</td>
</tr>
<tr>
<td>Crossover Probability</td>
<td>0.90</td>
</tr>
<tr>
<td>Reproduction Probability</td>
<td>0.10</td>
</tr>
<tr>
<td>Mutation Probability</td>
<td>0.33</td>
</tr>
<tr>
<td>Selection</td>
<td>Fitness Proportional</td>
</tr>
<tr>
<td>Functions</td>
<td>+, −, ×, sin, cos, exp, rlog</td>
</tr>
<tr>
<td>Terminals</td>
<td>X, 0</td>
</tr>
<tr>
<td>Max Depth in Cross</td>
<td>15</td>
</tr>
<tr>
<td>Max Depth in Mutation</td>
<td>15</td>
</tr>
</tbody>
</table>
Nevertheless, it is also remarkable in Fig. 1(b) that worst individuals present an exception with an increase in their size, which contrast with the global trend. This behavior was not observed within Fig. 1 (a) corresponding to the Even Parity 5 problem. Nevertheless, the reason may be due to the way data are presented within the graph. For better analyzing the results Fig. 2 shows the same results on a 3D-Graph for both problems. We see that the same phenomena is present on the EP5 problem.

In the next experiment we use the plague operator. Fig. 3 shows the results for the EP5 and SLR problem with an initial population of 1000 individuals and 100 generations. Plague acts by removing the 87 worst individuals from the population every 10 generations (although similar results are obtained with different parameters for the plague operator). In this figure we can see how the latter individuals are the ones which are eliminated every 10 generations. These eliminations produce two effects within the graph: firstly, the sawtooth in the figure, because we are not eliminating individuals every generation; and secondly the black area corresponding to the individuals that have been eliminated. Black areas in length graphs will represent individuals with sizes equal to zero, therefore each deleted individual will be represented as an individual with a length equal to zero.

Given that plague deletes the worst individuals of the population (notice that for every generation, worst individuals are again exceptionally large along classic plague experiment), a collateral effect is being produced: plague eliminates exceptionally large individuals that were not going to produce
new material in next generations, because of their low fitness values (their probability to be selected is low). We could consider an alternative: the elimination of better individuals that have a higher impact on new individuals generated. The memory consumption and computing time might improve for obtaining the best individual in the experiment.

B. Random Plagues

The improvement that we propose for the plague operator is the following one: remove a number of \( f \) randomly selected individuals, instead of the worst ones. Therefore, we will sometimes affect individuals with good fitness values, and also with large sizes. Given that worse individuals—which feature large size—will have low probability of reproduction, and therefore of transmitting their large size to descendants; the effect of random plague will thus be an extra help for reducing size. We haven’t considered here the possibility of also adding individuals to the population for improving plagues: we are only comparing different policies for removing individuals.

The new operator has been configured for working each 10 generations (as the classic plague, for comparison purposes), randomly deleting \( f \) individuals from the population. If we choose a high \( f \) we will get an empty population soon, so it is very important the value that we will choose for this parameter. We kept the operator from removing the best individual in the population, so that the algorithm is set up with the elitism option to ensure convergence (see [11]).

Both problems have the same set of parameters for GP (see Table I).

We can see in Fig. 4 (a) the results obtained for the EP5 problem when applying the random plague operator eliminating every 10 generation 87 individuals, and beginning with an initial population’s size of 1000 individuals. The black area of the figure corresponds to individuals that have been eliminated. Fig. 4 (b) shows the same results for SLR problem. We see that individuals growth is directly related with fitness values.

Again brighter individuals –larger ones– appear in the group of the worst individuals. Even when the random plague is applied, worst individuals continue being exceptionally larger when compared with the whole population (the same happens with regular plague, see Fig. 3). This feature can therefore be considered a constant for GP populations regardless its fix or dynamic size.

Table II shows detailed results. The effort (computed as described in [7]) that we have chosen for comparing results corresponds to the last generation of the random plague (the end of the experiment). Therefore it is the total effort required for obtaining the best individual in the experiment. Table II (a) shows that the new random plague operator has obtained the best fitness value when compared with the other two experiments. But the most interesting information comes from the comparison between the classic plague and the new one. We can see how the new operator gets a smaller best individual for the same effort, even though the average length of population is bigger in the random plague compared with the other experiments. This behavior could be explained if we see that random plague is always in the last generation while the other two experiments are in earlier ones, so the other two experiments are not so evolved as the random plague. Similar conclusions can be drawn from Table II (b) for the Symbolic Linear Regression problem.

Again the effort was chosen from the final generation of new random plague operator. With the Symbolic Linear Regression problem the random plague operator does not get the best individual fitness, although differences are not significant. Nevertheless the differences for best individual length are of interest: a smaller best individual is found again. As with the Even Parity 5 problem the population average length is bigger in the new operator due to the fact that
Fig. 5. New Plague: 87 Random Individuals Removed without Elitism. Population Size 1000.

(a) Even Parity 5

(b) Symbolic Linear Regression

TABLE III

<table>
<thead>
<tr>
<th>Effort / Problem</th>
<th>Fitness</th>
<th>Effort / Problem</th>
<th>Fitness</th>
<th>Effort / Problem</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.15 x 10^5</td>
<td>9.14</td>
<td>0.02</td>
<td>0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.86</td>
<td>8.98</td>
<td>0.04</td>
<td>0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.82</td>
<td>8.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22.12</td>
<td>22.12</td>
<td>21.74</td>
<td>21.74</td>
<td></td>
<td></td>
</tr>
<tr>
<td>288.86</td>
<td>288.86</td>
<td>25.66</td>
<td>25.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>273.84</td>
<td>273.84</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>156.64</td>
<td>156.64</td>
<td>159.21</td>
<td>159.21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>196.21</td>
<td>196.21</td>
<td>189.26</td>
<td>189.26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>99</td>
<td></td>
<td>99</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 6. Fine Grain Parallel Model

C. Random Plague without Elitism

The idea now is quite simple: we will not exclude the best individual from being deleted in the new plague operator. Given that the best individual is among the larger ones we will analyze whether its possible elimination affects largely the fitness or length evolution. We will run the same experiments as before and with the same parameters.

In Fig. 5 we can observe the results of this new approach. The figure shows two images: one for the EP5 problem (a) and another one for the SLR problem (b). In both cases we are representing with a gray scale the values of length. If we compare this figure with the previous one Fig. 4(a) we will not found large differences.

In Fig. 5(b) we can see the results for the SLR problem. If we compare this new experiments with the Fig. 4 (b) we will not found again significant differences.

Table III shows the detailed values obtained for this new experiments. In the table we compare as before the results obtained for a given effort. Corresponding to the last generation of the random plague operator. In the Even Parity 5 problem we observe that plague operator without elitism continues obtaining the best individual fitness, and the smaller one if we compare it with the classic plague operator. For the Symbolic Linear Regression problem we have the same fitness value when comparing it with the classic plague operator. In this case the best individual is bigger when compared with the classic plague operator.

We can conclude that random plague without elitism provides individuals of the same quality as classic plague –Symbolic Linear Regression experiments– or even better ones –like in Even Parity 5 experiments–. On the other hand the regular experiment without plague always provides worst results for the EP5 –harder problem–, while similar quality results for SLR –easier problem–.

This new results lead us to consider the random plague from a very different point of view. When the fine grain parallel model is applied to GP (see [24]), individuals from the population can be simultaneously evaluated on different processors/computers; Fig. 6 shows the model.

When global computing facilities are employed, volatility is a key feature that must be circumvented by applying special techniques, such as check-pointing. Nevertheless, computers failures could be understood as a random plague: if a computer fails, the individual is removed from the population. The only difference is that the number of computers that might fail will not be a fixed number –as with individuals removed in previous experiments–. Random plagues have taught us that loosing some individuals is not important for the global experiment. The experiment can continue and still good solutions (even better) will be found.
D. Fault Tolerance with Fine Grain Parallel GP

In the previous section we have shown GP continues providing solutions of quality in the presence of random plagues, even when the best individual may be removed.

When the fine grain parallel model is applied to GP individuals are evaluated simultaneously on different nodes or computers. In both cases, when a multiprocessors computer or a network of computers is employed there exists the possibility of losing some of the fitness values due to processors or networks failures: power-off, hangs, buffer overrun, etc. If this happen some special techniques for solving this problem should be employed: an error handling mechanism must be employed so that our algorithm or system becomes Fault Tolerant. Some of these available techniques are check pointing, n-version, redundancy, etc. [6], [22], [12], [1]. Any of these techniques consume extra time for handling the errors. They also add complexity to the algorithm.

The idea now is to apply conclusions obtained before when considering the fine grain parallel GP algorithm, studying whether the results of random plagues allow us to avoid any of those techniques and still provides fault tolerance. Each deleted individual will thus correspond to a fault in the parallel or distributed environment. Instead of applying random plague, we consider now that when the experiment is performed, random faults will appear which implies that some of the results are lost, and the corresponding individual is removed from the population.

Obviously the experiments performed above don’t fit in the exact scenario of a parallel environment so we have redefined the experiments to delete randomly individuals according to a percentage of computing/communication failures. This percentage will simulate the random lost of resources in a parallel environment. Our scenario will be more aggressive than before, because we will assume a continue lost of resources (we will consider faults every generation causing a lost of $n\%$ of individuals).

The experiments will simulate a Master/Slave scenario where the master node creates the new populations and sends to slave nodes the individuals that have not been yet evaluated. We also consider that we begin the experiments with the same number of slave nodes as the initial population size: if we have an initial population of 1000 individuals we will have 1000 slave nodes.

The experiments have been run with a percentage of failure equal to 2% and 10%. The experiments have been set up with the same parameters as before for comparing the results subsequently. The same as before, every experiment has been performed 50 times and results showed are the average.

In Fig. 7(a) and 7(b) we can see the effects of loosing each generation a 2% of the resources. Similarly as in previous sections, the black areas that we obtain for each figure are due to the eliminated individuals, in this case will represent death computer nodes.

Table IV shows a summary of the results obtained for this new experiments. Results are compared with the classic algorithm (free of faults, no parallelization is employed) and the classic plague (the source of the idea for considering fault tolerance). In this Table we don’t have the Best Individual Average Length because here we are only interested in maintaining the quality of solutions. In sub-table (a) we have the results for the Even Parity 5 problem with a 2% of failure and in (b) with a greater percentage of failure 10% for the Symbolic Linear Regression problem.

In both cases we have not obtained worse solutions than the fault-free environment. Therefore we have achieved what we proposed at the beginning of the section: obtaining at least solutions of the same quality in a non fault-free environment. In some cases we have even improved the quality of solutions, but the most important conclusion is that we can run a fine grained Parallel Genetic Programming algorithm.
without adding any special technique for handling the errors that could occur, at least for the couple of problems we have studied. We have shown that it will be only necessary to ignore them and continue the execution of the algorithm, as the random plague taught us.

IV. CONCLUSIONS

This paper has shown a detailed study on the evolution of individuals size for dynamic-size GP populations. By analyzing all individuals from experiments employing the Even Parity 5 and also Symbolic Linear Regression problems, we have found some interesting features. We have presented results in a 2D and 3D fashion that allows us to find more easily the relationship between length and size evolution, and exceptions in the global trend. Firstly, we have seen that while the general assumption that establishes a direct relationship between fitness and length is confirmed when using dynamic size populations, we have seen that not only for plagues but also for classic fixed-size populations, worst individuals feature exceptionally large size. Secondly, this has allowed us to design a new kind of plague, the random plague, which has provided better or similar results in a couple of benchmark problems. Although the size of solutions are larger than that obtained with regular GP, it improves over the classic plague.

Finally, results obtained from random plagues has allowed us to consider faults that are present when running experiments on distributed computing systems. Experiments performed has shown that the master/slave version of Parallel GP (being individuals evaluated on different processors) is fault-tolerant by nature. Even when a number of results – individuals– are lost, the quality of solutions remains.

In the future work we will confirm the latter results on an actual distributed environment, and we will also study more deeply the implications of bad-large individuals within the evolutionary process.

REFERENCES