Control of Bloat in Genetic Programming by Means of the Island Model

Francisco Fernández de Vega[1], German Galeano Gil[1], Juan Antonio Gómez Pulido[2], and Jose Luis Guisado[1]

[1] Centro Universitario de Mérida, University of Extremadura
C/ Calvario, s/n. 06800 Mérida, Spain
{fcofdez,ggaleano}@unex.es
[2] Escuela Politécnica, University of Extremadura,
Cáceres, Spain
jangomez@unex.es
http://atc.unex.es/pacof

Abstract. This paper presents a new proposal for reducing bloat in Genetic Programming. This proposal is based in a well-known parallel evolutionary model: the island model. We firstly describe the theoretical motivation for this new approach to the bloat problem, and then we present a set of experiments that gives us evidence of the findings extracted from the theory. The experiments have been performed on a representative problem extracted from the GP field: the even parity 5 problem. We analyse the evolution of bloat employing different settings for the parameters employed. The conclusion is that the Island Model helps to prevent the bloat phenomenon.

1 Introduction

When an evolutionary algorithm is applied to a difficult problem, a large number of individuals is usually required for making up the population, and very frequently, a large number of generations have to be computed in order to find a successful solution for the problem. Therefore, the computational effort required for solving difficult problems is sometimes prohibitive.

It is also well known that in Genetic Programming (GP) – one of the members of EAs' family – individuals tend to increase their size as population evolves. This growth is not necessarily correlated with increases in the fitness of the evolved programs, and many times individuals increase their size while fitness doesn't improve. The problem is that individuals require computing time to be evaluated; and given that individuals undergo the problem of increasing their size as they are evolved, generations will take progressively longer time to be computed, which is a big concern for GP researchers.

The above describe problem is usually known as the bloat phenomenon, and has been frequently studied during the last few years [3, 4, 5, 6, 13, 14]. As said above, bloat has a large impact in the search process.

Besides presenting several studies that aims at offering reasons for the bloat [13, 14], researchers try to offer alternatives for controlling that problem. In [3] some of these proposals are described: firstly, by placing a universal upper limit either on tree depth or program length; secondly, by incorporating a penalty which is proportional to program size; and finally, tailoring the genetic operations.

On the other hand, given that any EA requires long time for solving difficult problems, some approaches taken from parallel computing field have also been applied to alleviate the problem. These approaches try to apply some degree of parallelization to the basic algorithm. There are important differences in the way it can be done, and the kind of EA employed. For instance, Cantú-Paz [10] studied how to apply parallelism to Genetic Algorithms (Gas), and also presented some theoretical results involved in the parallel version of the algorithm. Also Fernández et al [7, 8, 9] have studied Parallel GP, and the importance of some key parameters that are only employed in the parallel algorithm, such as synchrony, migration frequency, granularity and so on.

All of the above referred studies employ the well-known Island model for the parallel algorithm. The idea behind this model is simple: Divide the entire population of individuals into smaller ones, so that each sub-population is computed in a different processor. Populations may sometimes exchange good individuals. The results have shown that the Island Model improves fitness quality of individuals while also saves computing time because of the use of several processors [7, 8, 10].

Nevertheless, for the time being, the Island model has always been analysed with the idea of improving quality of solutions and for saving computing time, and only once a report on the evolution of bloat when using the model for GP has been presented [9], although no hypothesis for the reason of the behaviour observed was provided.

In this paper we continue the study on the bloat phenomenon, employing this non-traditional point of view: Instead of focusing on the kind of genetic code that causes bloat (such as non-operating instructions in programs, subprograms functionally equivalent to instructions, etc. [14]) we show that the island based algorithm can also help to prevent the bloat of individuals in GP, while we also provide a reason for this behaviour. The island models could be thus considered as a new technique for preventing bloat when looking for solutions by means of GP.

A set of experiments is presented for supporting the theoretical motivation that describes the advantages of the Island Model.

This paper is structured in the following way: Section 2 describes the theoretical motivation that justify why Island Models are of interest when fighting bloat. Section 3 briefly describes the problem studied, while Section 4 shows the results we have obtained in a set of experiments. Finally, section 5 presents our conclusions and future work.

2 Motivation

Langdon and Poli have established that programs size increases on average at less than $O(time^{\alpha})$ with $\alpha \in [1.2,2]$, and it will approach a square power law, $O(t^2)$, as the programs get bigger (see [5,6]) In this section, for simplicity, we assume the later $O(t^2)$ for the increase of programs size (although other α values within the range will not change the main conclusion). We also consider that programs sizes are measured be counting the number of nodes contained in the tree.

When a generational evolutionary algorithm is employed, time can be measured by means of the number of generations computed, so that time and generations have similar meanings. When generations are computed time runs. Therefore, we could formulate $O(g^2)$ as an equivalent expression to $O(t^2)$ (g being generation computed). We can then write:

$$O(t^2)=O(g^2) \quad (1)$$

'=' sign according to big-Oh notation.

We also know that individuals grow when crossover and mutation operators are applied. A GP algorithm lacking both operands -that only apply selection and copy of individuals- will evolve to a population of identical individuals, all of them copies of any of the individuals making up the initial population. This is because no variation operator is employed. In this condition, the bloat phenomenon is not present, because individuals can not change their size (the only difference between the size of individuals at the end of the algorithm is due to the difference between the individual that has dominated the population and the remaining ones). Nevertheless this cannot be considered bloat.

Therefore, bloat only exists when variation operators are applied to produce new individuals, and may thus appear larger ones. The more new individuals we produce (crossover and mutation operators are then applied), the more opportunities for bloat. We could thus say that bloat approach $O(i^2)$ *i* being the number of new individuals produced per generation, and according to equation (1):

$$O(t^2)=O(g^2)= O(i^2) \quad (2)$$

If the number of individuals created per generation is proportional to the number of individuals in the population, which is usually the case, we could instead employ the expression $O(n^2)$ -*n* being the number of individuals in the population- for the limit on bloat growth. This can only be stated if mutation and crossover are applied a number of times proportional to the size of populations. Summarising we have the following expression:

$$\text{Bloat}(n) = O(t^2) = O(g^2) = O(i^2) = O(n^2) \quad (3)$$

The first idea obtained from the above expression is that different population sizes should produce different bloat values.

By analysing these results and those presented for the island model in [9], and considering that $bloat(n)= O(n^2)$, being *n* the size of the population, we can ask the following question: what would happen if we distribute all of the individuals among several subpopulations? If we employ *x* subpopulations, we would have *n/x* individuals per subpopulation. What we will try to analyse now is whether such distribution change the bloat evolution.

The big-Oh notation from equation 3 tells us that the bloat equation for a population with *n* individuals is a second degree polynomial function:

$$bloat(n) = O\left(n^2\right) \Rightarrow bloat(n) = an^2 + bn + c \quad (4)$$

By *bloat(n)* we mean a function that measure the number of nodes obtained with a population of size *n* on a given problem. If the number of individuals in the population *n* is not very small, then we can write:

$$bloat(n) \approx an^2 + bn \quad (5)$$

We are studying the bloat evolution when individuals are distributed among a set of populations, so that the total number of individuals *n* remains unaltered, but we employ *n/x* individuals in each of the *x* populations. The total bloat for each of the sub-

populations is *bloat(n/x);* if we consider that the bloat phenomenon occurs similarly in every subpopulation, employing equation 5 we can add up to obtain the total bloat:

$$\sum_{i=1}^{x} bloat(\tfrac{n}{x}) = x \cdot bloat\left(\tfrac{n}{x}\right) \tag{6}$$

Let's now make an assumption, and then we will try to check if it can be satisfied. Let's consider that the island model does not affect the evolution of bloat, i.e. the total bloat obtained when adding the bloat for each of the subpopulation correspond with the bloat obtained when using only one population. If this is true, the following expression must be true:

$$x \cdot bloat\left(\tfrac{n}{x}\right) = bloat(n) \tag{7}$$

But, if we substitute in both terms of the expression employing equation 5, we obtain:

$$x\left(a\frac{n^2}{x^2} + b\frac{n}{x}\right) = an^2 + bn \tag{8}$$

and equivalently:

$$\frac{an^2}{x} = an^2 \tag{9}$$

And this can only be true for $x=1$ i.e. employing the panmictic model. Nevertheless we first stated that we employ the island model, so that $x>1$. So, the left part of expression (9) will become smaller as new populations are added to the model. Given this contradiction, we infer that the initial assumption of an equivalent global rate of bloat in the Island Model is false.

Once we have seen that bloat rate is different when using the Island Model, and given the left part of equation (9), we conclude that as more subpopulations we use, a smaller bloat we will obtain. This statement is in agreement with results obtained in [9], in which the Island Model was applied to study bloat, although no clue for the behaviour observed was provided.

In the following sections we revisit the Island Model, for analysing again bloat evolution in a very well-know problem employed in GP studies: the even parity 5 problem. We try to see if the predictions from the theory confirmed by experimentation.

3 Experiments

We have employed a well-known GP problem, the even parity 5 problem [2], with several settings in order to make comparisons. The goal in the even parity 5 problem is to generate a Boolean function that gives us the parity of a set of 5 bits. The Parallel GP tool employed is described in [12].

Experiments have been performed in a cluster of PCs running a distribution of Linux specially suited for clusters [1]. We show the averages values obtained over 50 runs for each of the experiments.

Table 1 shows the setting used in the experiments with the evenp-5 problem. The maximum depths have been established identically in all of the experiments, so that differences we may found will not be due to differences in this parameter. On the

other hand, Table 2 provides the parameter specifically employed when using the Island Model (See [7] for a whole description of the model). The number of subpopulations is different for each of the experiments performed, and this information is provided within the graphs.

Table 1. Parameters for the Evenp Problem.

Crossover Probability	98.0
Creation Probability	1.0
Max. Depth For Creation	6
Max. Depth For Crossover	17
Swap Mutation Probability	1.0
Selection	10 individuals per Tournament

Table 2. Parameters for the Island Model.

Generations between Migration	10
% of the population that migrate	10%
Synchrony	Asynchronous model

4 Results

4.1 Panmictic Model

Before analysing results obtained with the Island Model, we have performed a couple of experiments to check the validity of Equation (3). The equation tell us that the bloat phenomenon must change with the size of the population, so we have firstly performed an experiment employing different population sizes when using the evenp-5 problem and the panmictic model.

Figure 1 presents the results that we have obtained for the evenp-5 problem. We have performed 50 experiments for each of the curve, and then we have computed the average length per generation. We notice that when we increase the size of the population, the average length is larger. So, the first idea extracted from equation (3) is confirmed by results obtained.

The second idea that can be extracted from equation (3) is simple: if we change the number of genetic operations applied per generation, or equivalently, the number of new individuals created per generation, we may change the bloat rate. We have thus performed another experiment that helps us to confirm this idea for the panmictic model.

The experiment is simple: suppose we don't create each generation a number of new individuals proportional to the size of the population n, but we create instead a number of new ones proportional to $n/2$ or proportional to $n/4$. Equation (3) tell us that bloat obtained is proportional to the number of new individuals created, and this time, this value would not be equivalent to the size of the population.

Figure 2 shows the evolution of the average length of individuals that have been obtained in experiments following the idea described above. Each curve corresponds to an experiment in which a number of new individuals are created (a percentage of the size of the population). We can easily observe that bloat is smaller when the num-

ber of new individuals created reduces (a smaller number of crossover and mutation operations are applied). Its interesting to see that bloat is not present when no individuals are created, as we could expect (curve labelled as "without evolution"). Several population sizes have been employed with similar results.

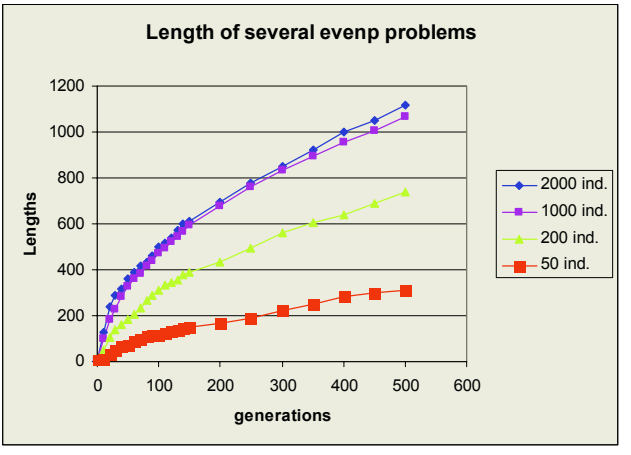

Fig. 1. Length Evolution in the even parity 5 problem, with different populations' sizes. The bigger population, the larger bloat.

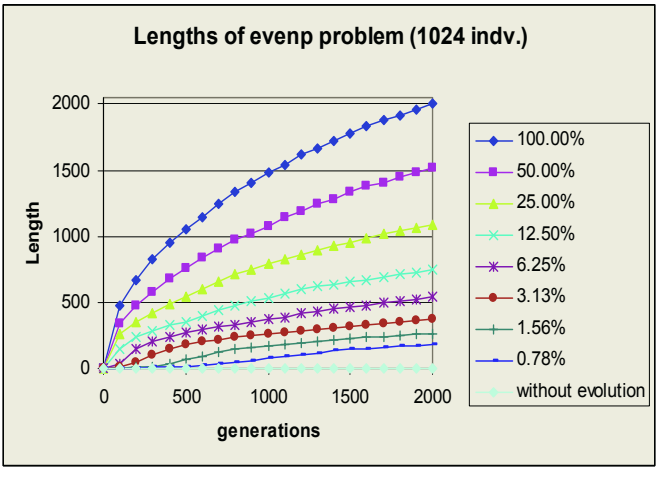

Fig. 2. Length Evolution in the evenp-5 problem Size of population= 1024. Percentage values means the number of new individuals over the size of the population created each generation.

Table 3, numerically shows the average size of individuals in different number of generations. These results numerically show the same results provided in figures 1 and 2. Nevertheless, they help to verify quantitatively how different rates of creation of individuals change the rate of bloat. We can notice that no bloat occurs when no new individuals are created.

Table 3. Length evolution in the evenp problem.

% new indiv.	Generations			
	2000	1000	500	100
100.00%	2002.4	1483.8	1057.5	474.6
50.00%	1514.8	1074.9	760.1	335.2
25.00%	1091.2	789.6	548.1	258.5
12.50%	746.5	533.8	353.1	148.4
6.25%	541.0	374.2	270.6	38.2
3.13%	368.3	262.4	179.3	7.4
1.56%	263.6	169.4	64.0	5.1
0.78%	180.4	75.6	11.8	5.0
0.39%	65.1	10.4	5.3	5.0
0.00%	5.0	5.0	5.0	5.0

4.2 Island Model

The next step was to analyse the island model. We want to maintain the classical setting, by generating a number of new individuals per generation proportional to the size of the population. We want to compare bloat evolution with that observed within the panmictic model.

Several experiments have been performed for the evenp-5 problem, employing different population sizes for both the panmictic model, and also employing the Island Model and 2, 4, 10 and 20 subpopulations (equally distributing all of the individuals among the subpopulations employed in each experiment). Figure 3 shows the average length value obtained over 50 runs in a couple of experiment that employs 100 individuals, while figure 4 has been obtained employing 2500 individuals. We can see that bloat reduces as a larger number of subpopulations are employed to distribute the global population.

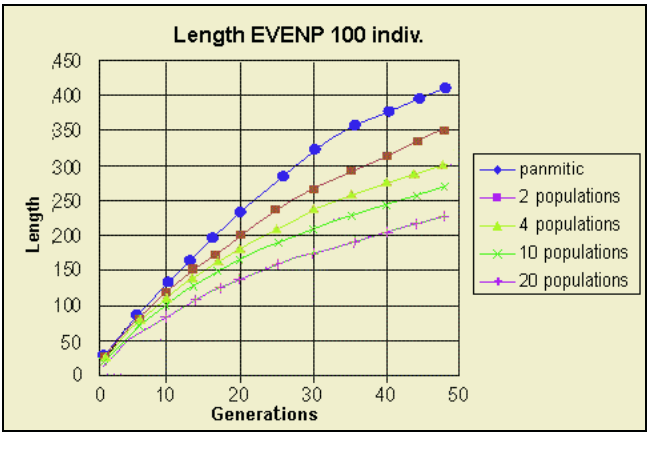

Fig. 3. Length evolution in the evenp-5 problem when using the Island Model.

We have not tried to study in this paper which is the best number of island to be employed in a given experiment. This problem has been addressed before, and no

perfect recipes exist [7]. Nevertheless, the above presented results make evident a new advantage of the Island Model that to our best knowledge has not been presented for GP before.

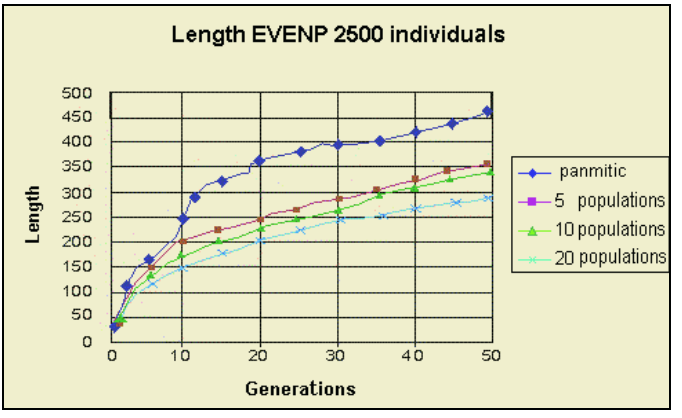

Fig. 4. Length evolution in the ant problem when using the Island Model.

The results obtained with the Island Model in the problem, confirms the prediction of the theoretical model employed in section 2: the bloat rate reduces when individuals are distributed among a set of subpopulations. This is a new advantage for the Island Model: Not only it find better fitness quality, but also reduces the bloat phenomenon.

The above mentioned advantage of the Island Models is now added to another important feature of island-based GP: it can be easily computed on multiprocessor systems: we just have to distribute populations among processors, and both advantages helps together to save computing effort when GP is applied to solve optimization problems.

5 Conclusions

A new approach for reducing the bloat problem is presented in this paper. We have first employed a theoretical model that predicted the different rate of code growth when the island model is employed in Genetic Programming. The model suggests that distributing individuals among a number of subpopulation will reduce the bloat phenomenon, and this reduction is larger when more subpopulations are employed.

The even partiy 5 problem has been employed as a benchmark for experimenting the evolution of code growth. We also shown that results obtained are coherent with those predicted by the model.

We have studied the evolution of bloat in panmictic models, when the number of genetic operations applied each generation (and also the number of new individuals created) is not the same as the size of the population. By means of this experiments we have presented evidences that make us to be confident about the predictions of the theory on the Island Model.

Finally, the study has focussed on the Island Model. We have first seen that bloat depends on the size of the subpopulations, and then we have performed several tests employing the Island Model, that have experimentally shown in a well-known GP problem that distributing a population into several ones of smaller size helps to prevent the bloat phenomenon.

In the future, we will present a larger report including a wider set of both test and real-life problems corroborating conclusions presented in this paper.

References

1. Philip M. Papadopoulos, Mason J. Katz, and Greg Bruno, *NPACI Rocks: Tools and Techniques for Easily Deploying Manageable Linux Clusters* ,. Concurrency and Computation: Practice and Experience. Volume 15, Issue 7-8, Date: June - July 2003, Pages: 707-725.
2. J.R. Koza: "Genetic Programming. On the programming of computers by means of natural selection". Cambridge MA: The MIT Press. 1992.
3. W. Langdom and R. Poli. "*Fitness causes bloat"*. In P.K. Chawdhry et. Al., editors. Soft Computing in Engineering Design and Manufacturing, pp 13-22. Springer London, 1997.
4. W. Banzhaf, W. B. Langdon, *"Some Considerations on the Reason for Bloat"*, In Genetic Programming and Evolvable Machines, 3, 81-1, 2002.
5. W.B. Langdom, Riccardo Poli. *Foundations of Genetic Programming*. Ed. Springer, 2001. "Convergence and *bloat*". Pp 193-217
6. W. B. Langdon, *"Quadratic Bloat in Genetic Programming"*, In proceedings of the 2000 Genetic and Evolutionary Computation Conference. 2000.
7. F. Fernández, "Parallel and Distributed Genetic Programming models, with application to logic synthesis on FPGAs", PhD Thesis. Universidad de Extremadura, February 2001.
8. F. Fernández, M. Tomassini, L. Vanneschi, "*An Empirical Study of Multipopulation Genetic Programming*", . Genetic Programming and Evolvable Machines, Vol. 4. 2003. pp. 21-51. Kluwer Academic Publishers.
9. G. Galeano, F. Fernández, M. Tomassini, L. Vanneschi, "*Studying the Influence of Synchronous and Asynchronous Parallel GP on Programs Length Evolution"*. In Proceedings of Conference on Evolutionary Computation 2002.
10. Erick Cantú-Paz. "Efficient and Accurate Parallel Genetic Algorithms ". Kluwer Academic PublishersISBN 0-7923-7221-2. Volume 1 of the Book Series on Genetic Algorithms and Evolutionary Computation
11. W.F. Punch: "How effective are multiple populations in Genetic Programming". Genetic Programming 1998: Proceedings of the Third Annual Conference, J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, D. Goldberg, H. Iba and R. L. Riolo (Eds),Morgan Kaufmann, San Francisco, CA, 308-313, 1998.
12. M. Tomassini, F. Fernández, L. Vanneschi, L. Bucher, "*An MPI-Based Tool for Distributed Genetic Programming"*. In Proceedings of IEEE International Conference on Cluster Computing CLUSTER2000, IEEE Computer Society. Pp.209-216. 2000.
13. T. Soule, *Exons and code growth in genetic programming,* In J. A. Foster et al (eds.) LNCS 2278. pp. 142-151. Aril 2002.
14. S. Luke « Modification Point Depth and Genome Growth in Genetic Programming ». Evolutionary Computation. Spring 2003. Vol 11, Num 1. pp 67.