Swarm and Evolutionary Computation 32 (2017) 110–120

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

The influence of population size in geometric semantic GP

Mauro Castelli [a,*], Luca Manzoni [c], Sara Silva [b], Leonardo Vanneschi [a], Aleš Popovič [d]

[a] NOVA IMS, Universidade Nova de Lisboa, 1070-312 Lisboa, Portugal
[b] LabMAg, FCUL, Universidade de Lisboa, 1749-016 Lisboa, Portugal
[c] Dipartimento di Informatica Sistemistica e Comunicazione (DISCo), University of Milano Bicocca, 20126 Milan, Italy
[d] University of Ljubljana, Faculty of Economics, Kardeljeva ploscad 17, 1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 25 September 2015
Received in revised form
29 January 2016
Accepted 31 May 2016
Available online 3 June 2016

Keywords:
Genetic programming
Semantics
Population size

ABSTRACT

In this work, we study the influence of the population size on the learning ability of Geometric Semantic Genetic Programming for the task of symbolic regression. A large set of experiments, considering different population size values on different regression problems, has been performed. Results show that, on real-life problems, having small populations results in a better training fitness with respect to the use of large populations after the same number of fitness evaluations. However, performance on the test instances varies among the different problems: in datasets with a high number of features, models obtained with large populations present a better performance on unseen data, while in datasets characterized by a relative small number of variables a better generalization ability is achieved by using small population size values. When synthetic problems are taken into account, large population size values represent the best option for achieving good quality solutions on both training and test instances.

1. Introduction

As reported in several studies (see for instance [8,4,5,16,37,39]) the performance of Genetic Programming (GP) [25] is strongly dependent on the value of a set of parameters. Among those parameters, one that has a deep impact on GP's functioning is the size of the population, i.e. the number of candidate solutions that are evolved. In particular, the size of the population is involved in several phenomena that characterize GP. For instance, population size and population diversity are related to premature convergence [45,7] and it was also hypothesized that the population size is related to the occurrence of bloat [34]. Furthermore, while existing studies suggest that bloat and overfitting are unrelated phenomena [43], other studies hint the existence of a relation between these phenomena [32]. Under this perspective, an incorrect choice of the population size may be one of the reasons for the overfitting of training data. For all these reasons, an accurate choice of the value of this parameter is often crucial. For instance, a small population may result in premature convergence or in poor performance of GP. On the other hand, a large population may cause a slowdown of the algorithm due to the high number of fitness evaluations that are needed.

The study of the parameters that characterize GP is an important hot topic, in particular when geometric semantic operators, defined by Moraglio and coauthors in 2013 [28], are used to explore the search space [44]. The definition of these operators, in fact, has opened a new research line in the GP community, and a lot of theoretical studies have appeared [29,31]. Besides being grounded in a strong body of theory, the use of these genetic operators has produced substantially better results, compared to standard GP, on a number of problems, both benchmarks [42] and real-world applications [12,13].

The objective of this paper is to study the role of population size on the learning process of GP when geometric semantic operators are used. In particular, we want to investigate the role of the population size in achieving good quality models, both on training and unseen data. This study has been performed considering several test problems and different population size values, including GP with only one candidate solution in the population.

The paper is organized as follows: Section 2 presents previous works related to the importance of population size in evolutionary computation, pointing out some interesting findings; Section 3 reports the definition of the geometric semantic operators presented in [28]; Section 4 presents the experimental settings and the obtained results, discussing the effect of different population size values on the learning process. In particular, an analysis of the quality of the obtained models and their ability to generalize on unseen data is proposed. Finally, Section 5 concludes the paper and provides hints for future research directions.

* Corresponding author.
E-mail addresses: mcastelli@novaims.unl.pt (M. Castelli), luca.manzoni@disco.unimib.it (L. Manzoni), sara@fc.ul.pt (S. Silva), lvanneschi@novaims.unl.pt (L. Vanneschi), apopovic@novaims.unl.pt (A. Popovič).

http://dx.doi.org/10.1016/j.swevo.2016.05.004
2210-6502/

2. Population size: previous and related work

The study of the effect of the population size in evolutionary algorithms has been investigated in several works so far. In this section a brief literature review is presented, in order to frame our work in the context of the existing studies. The first studies that have appeared concern Genetic Algorithms (GAs): in [18] the notion of genetic drift was introduced and a study on the relation between genetic drift and population size has been reported. Genetic drift was defined as an effect based on stochastic property of the algorithm. Consider having population of single digit binary strings, the first half of them is "l", and the second one is "0". If we will choose strings by chance for the creation of the new generation, we can expect getting equal quantity of different strings. In real, as generations passed, we will observe increase of heterogeneity, which can finally lead to the disappearance of the definite type strings from population. This phenomenon of loss of strings and their parts was called allele loss. In [18], the author observed that increasing the population size we can reduce not only allele loss, but also genetic drift significantly. In the same work, the author also showed that, in large populations, fitness improves more slowly in the initial phase of the evolution but the overall result is better than the one achieved with a small population. In [22], the results of experiments to determine the optimum population size and mutation rate were presented for a simple real genetic algorithm defined to solve problems in the electromagnetic domain. A general domain-independent theoretical analysis of the optimal population size is provided in [20]. In [41] a deeper look on the problem of the population size and limit of generations was taken. The research involved investigation of GA's behavior on different types of functions (i.e., functions with many optima, functions with limits in search space and NP-hard functions). Results have shown that a population size increase improves the performance of GAs and affects the results more than changing the number of generations.

Other studies have investigated the relation between population size and selection or crossover. For instance, the authors of [15] investigated the relation between the population size and the selection strategy. They showed that the number of individuals needed to find a solution is much larger for proportionate selection strategy then for the truncation selection. The GA algorithm they used did not take into consideration mutation rate and it used only one-point crossover. In the same way, the authors of [17] investigated joint influence of crossover type and population size on the performance of GAs. They used an analysis of strings disruption by crossover and its interaction with population sizing. The authors have shown that in small populations more disruptive crossover (uniform and *n*-point crossover operators) is more effective, and the use of less disruptive crossover (1- and 2-point crossover) in large populations leads to better performance. The possibility of using a GA to control the parameters of another GA has also been investigated. In [21] a meta-GA is applied with interesting results to control the parameters of another GA, including the population size. Several studies have tried to adapt the size of the population during the search process. In [1], the authors pointed out how the size of the population can be critical in many GA applications. They proposed an adaptive method for maintaining variable population size, which grows and shrinks together according to some characteristics of the search. Experimental results indicated some advantages of the proposed method. In the same vein, the work reported in [38] presents an algorithm which adjusts the population size with respect to the probability of selection error. More recently, the authors of [36] performed an experimental study about the importance of population size for the algorithm's performance. Results have shown that tuning the population size is not an easy task and the impact of that choice on the algorithm's performance is significant.

Regarding GP, some of the most relevant results are due to the work of Poli and collaborators. Based on their previous theoretical results obtained on GAs [33], they have been able to establish a schema theory that is able to bind the population size of GP to the code growth and the convergence rate. Using this theory, as better specified in [35], they discovered that, in standard GP, there is a direct relationship between the growth in the size of the individuals in the population and number of individuals in the population itself. In synthetic terms, in large populations individuals tend to grow more rapidly than in small populations. As a direct consequence, if we plot the evolution of fitness against the number of fitness evaluations (as we will do also in the experimental part of this paper), populations of large size are generally penalized. More in particular, adding individuals to a population is beneficial for the effectiveness of GP until a certain threshold, which depends on the particular problem that GP is trying to solve. When this threshold is reached, further increasing the population size produces a large computational effort, that is not anymore compensated by the obtained gain in terms of fitness. Interestingly, this theoretical study (strongly corroborated by experimental evidence in [35]) was only done for fitness on the training set. To the best of our knowledge, no studies have appeared so far aiming at studying the relationship between the population size and the generalization ability of GP. We speculate that the reason for this lack of study is due to an idea, widely diffused among researchers in machine learning, including GP, according to which larger individuals are likely to overfit. As such, given that large populations tend to generate large individuals, large populations have implicitly been considered as being affected by overfitting. Our work is very different from the work of Poli and collaborators. In fact, we do not consider standard GP, but geometric semantic GP (GSGP). The theoretical results of Poli and collaborators cannot for sure be applied to GSGP for the simple reason that, as it will be clear later in this paper, in GSGP the growth rate of the individuals in the population is fixed and independent from the population size. This is the first, and possibly most important, motivation for our work: for the first time, we want to understand the impact of population size on the performance of GSGP. Furthermore, we think that such a study cannot overlook the generalization ability of the studied system. For this reason, great attention is dedicated in this work to results obtained by GSGP on testing data, unseen at training time.

A contribution of a slightly different nature has been proposed in [45], where the authors studied a GP variable population size for dynamic optimization problems. Another contribution is described in [19], where the authors proposed a method for reducing the size of populations at a linear rate. This was achieved by removing a fixed number of individuals at each generation. This technique was called plague and it has been shown to have some positive effects on GP performance. A refinement of this work has been proposed in [40], where an extension of the plague technique, aimed at varying the population size in an intelligent way during the execution of each GP run, was presented. In that model, add and suppression of individuals are operated dynamically on the basis of the behavior of the GP system: population size is decreased while the algorithm is progressing (i.e. fitness is improving) and it is increased when the algorithm reaches the stagnation phase.

3. Geometric semantic operators

Even though the term semantics can have several different interpretations, it is a common trend in the GP community (and this is what we do also here) to identify the semantics of a solution