Expert Systems With Applications 109 (2018) 162–187

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Evolving dynamic fitness measures for genetic programming

Anisa Ragalo [a,*], Nelishia Pillay [b]

[a] School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, South Africa
[b] Department of Computer Science, University of Pretoria, South Africa

ARTICLE INFO

Article history:
Received 16 September 2017
Revised 15 February 2018
Accepted 28 March 2018
Available online 25 April 2018

Keywords:
Genetic programming
Genetic algorithm
Fitness

ABSTRACT

This research builds on the hypothesis that the use of different fitness measures on the different generations of genetic programming (GP) is more effective than the convention of applying the same fitness measure individually throughout GP. Whereas the previous study used a genetic algorithm (GA) to induce the sequence in which fitness measures should be applied over the GP generations, this research uses a meta- (or high-level) GP to evolve a combination of the fitness measures for the low-level GP. The study finds that the meta-GP is the preferred approach to generating dynamic fitness measures. GP systems applying the generated dynamic fitness measures consistently outperform the previous approach, as well as standard GP on benchmark and real world problems. Furthermore, the generated dynamic fitness measures are shown to be reusable, whereby they can be used to solve unseen problems to optimality.

1. Introduction

Genetic programming (GP) literature proposes a myriad of fitness measures (Koza, 1992b; Krawiec & O'Reilly, 2014; Krawiec & Swan, 2013; Lehman & Stanley, 2010; McKay, 2000; 2001), which aim to overcome different shortcomings, e.g. escaping local optima, reducing bloat, thereby improving on the performance of the algorithm. The convention in GP is to apply a given fitness measure individually throughout the course of the algorithm. However, the previous study (Ragalo & Pillay, 2017) showed that Dynamic Fitness Measure GP (DFMGP), which applies different fitness measures on the different GP generations, is more effective than following this convention. This premise is justified by the shift between exploration and exploitation in the course of the GP algorithm: exploration, a de facto global search, is used to promote coverage of the search space (Crepinšek, Liu, & Mernik, 2013; Eiben & Schippers, 1998); in turn, exploitation, a de facto local search, is used to refine promising solutions when good points in the search space have been discovered (Crepinšek et al., 2013; Eiben & Schippers, 1998). GP search is a constant balance between exploration and exploitation, with the former dominating the preliminary generations, and the latter, later generations (Koza, 1992b). DFMGP is shown to be effective when the fitness measure used at each generation supports the dominant search (i.e. exploration vs. exploitation) in the on-going phase of GP (Ragalo & Pillay, 2017).

In the previous study (Ragalo & Pillay, 2017), a genetic algorithm (GA) was used to induce the sequence in which fitness measures should be applied over the course of DFMGP. The current research builds on the concept of DFMGP. Rather than a GA, a meta-GP is used to generate combinations of the fitness measures. Here, the meta-GP is used to create a new function for the fitness measure, whereby the new function is a logical and arithmetic combination of existing fitness measures. Importantly, the meta-GP's variable length represenation[1] allows the evolving dynamic fitness measure to develop in size and structure, thus opening up further avenues to improve on the performance of DFMGP.

We refer to the combinations of fitness measures evolved by the meta-GP as composite fitness measures (CFMs). DFMGP applying the evolved CFMs is applied to 6 benchmark GP problem classes[2], namely, sextic regression (Koza, 1994), Keijzer-6 regression (Keijzer, 2003; White et al., 2013), even-*n* parity (Koza, 1994), *n*-bit multiplier (Walker & Miller, 2010), tartarus (Cuccu & Gomez, 2011; Dick, 2013; Trenaman, 1999) and deceptive tartarus (Cuccu & Gomez, 2011), and its performance is compared to DFMGP applying fitness measure sequences evolved by the GA approach

[1] The variable-length representation is a feature of all GP algorithms; the representation gives GP an advantage over GA, because it permits the solution structure to evolve in problems where it is difficult to a priori specify the size and structure of the optimal solution (Koza, 1992b).

[2] The literature (Haraldsson & Woodward, 2014) defines a problem class as a probability distribution over the instances of a given problem; this is the interpretation of the term used in the manuscript. The term problem domain is also used in the manuscript: a problem domain is a grouping of similar problems e.g. symbolic regression problems, Boolean function synthesis problems, etc. (Koza, 1992b).

* Corresponding author.
E-mail addresses: 204505443@stu.ukzn.za (A. Ragalo), npillay@cs.up.ac.za (N. Pillay).

https://doi.org/10.1016/j.eswa.2018.03.060
0957-4174/

(Ragalo & Pillay, 2017), as well as standard GP. DFMGP is also applied to 4 complex, real-world problems, namely, the abalone (Newman, Hettich, Blake, & Merz, 2015) and Dow Chemical parser (White et al., 2013) datasets from the symbolic regression domain, and the flame detection circuit (Kuphaldt, 2006) and binary-coded-decimal-to-seven-segment decoder (Mano, Kime, & Martin, 2008) problems from the Boolean function synthesis domain. Here, the CFMs evolved by training the meta-GP on the problem classes are tested on real-world problems from the same problem domain. The idea is to train the CFMs on simpler and less complex problems that will not take as much time to train on, and subsequently use the CFMs on the real-world problems. The study finds that it is better to use GP at the meta-level, rather than a GA: DFMGP applying the GP-evolved CFMs consistently outperforms DFMGP applying the GA-evolved sequences. The study also finds that the CFMs can be used to solve unseen problems to optimality. Therefore the current study makes the following contributions:

- The study compares the use of GP versus the use of a GA at the meta-level to evolve dynamic fitness measures for GP and finds that the former approach yields more effective dynamic fitness measures.
- The study shows that the GP-evolved CFMs are reusable, i.e. a CFM can be evolved for a problem class and yield good results on unseen problem instances of the class. The CFMs also produce better results than both the previous approach and standard GP on unseen complex, real-world problems from the same problem domain.

The manuscript is organized as follows. Section 2 discusses the GP algorithm. Section 3 details the fitness measures used in the study. Sections 4 and 5 describe DFMGP and the implemented meta-GP in detail. Section 6 outlines the methodology for the manuscript's experiment. Section 7 presents the results of the experiment, and discusses the significance of the results. Lastly, Section 8 concludes the manuscript and discusses possible directions for future work.

2. Genetic programming

Genetic programming (GP) is an evolutionary algorithm (EA) introduced by Koza (1992b) that explores a program space rather than a solution space as in the case of genetic algorithms. Hence, each element of the population is a program represented by a parse tree. The generational GP algorithm begins by creating an initial population that is iteratively refined via the processes of evaluation, selection and regeneration.

Evaluating the population involves calculating a fitness measure for each parse tree using a fitness function. Fitness functions calculate the objective fitness of a program which is a measure of how good the solution produced by the program is, and is hence problem specific. Tournament selection is the selection method usually employed by the GP algorithm to select parents (Koza, 1992b). This method uses the fitness measure of the elements of the population to make a choice. Genetic operators are applied to the chosen parents to produce offspring of the next generation. The genetic operators that are commonly used are reproduction, mutation and crossover (Koza, 1992b).

At the inception of GP, objective fitness was used as a fitness measure. As the field developed various fitness measures emerged in an attempt to overcome the shortcomings of GP such as premature convergence and the growth of redundant code. The following section provides a description of different fitness measures derived for GP. These have been chosen to include those fitness measures that have proven to be effective and have made an impact in the field, and more recent measures such as behavioural programming.

3. Fitness measures

This section describes the fitness measures used in the study. The fitness measures used are: (1) Objective fitness (OF) (Koza, 1992b), (2) Behavioural programming (BP) (Krawiec & O'Reilly, 2014; Krawiec & Swan, 2013), (3) Fitness sharing (FS) (Bersano-Begey, 1997; McKay, 2000), (4) Dynamic subset selection (DSS) (Gathercole & Ross, 1994), (5) Host-parasite coevolution (HP) (Hillis, 1990; Pagie & Hogeweg, 1997; Siegel, 1994; Williams & Mitchell, 2005), and (6) Novelty search (NS) (Lehman & Stanley, 2010; Martinez, Naredo, Trujillo, & Galván-López, 2013; Naredo & Trujillo, 2013; Naredo, Trujillo, & Martínez, 2013). The goal in providing a diverse subset of fitness measures is for the meta-GP to be able to produce optimal CFMs for varied problem classes.

A detailed description of the fitness measures ensues. For the sake of uniformity, the fitness measures are formulated as maximization functions. In formulating the fitness measures, the terminology 'taken from' is used for equations copied directly from the referenced text; however, the notation in these equations may differ from the referenced text, in conformity with the uniform notation applied in the manuscript. The terminology 'adapted from' is used for equations that borrow from the referenced text; in some instances, the equations in the referenced text are specific to the problems evaluated in the text; these equations are restated to express a general formulation for the given fitness measure; in other instances, the equations are restated to present the fitness measures as maximization functions. Lastly, where neither terminology is used, the equation is a formulation derived by the authors, based on the description of the fitness measure provided in the literature.

3.1. Objective fitness

Objective fitness (OF) is the canonical fitness measure applied at the origination of GP in Koza (1992b); most GP practitioners have the habit of relying on OF measures. In objective fitness GP (OF-GP), solutions that are closer to achieving the objective are considered to be better than solutions that are further away.

Eq. (1) is used to calculate a solution's OF score. Eq. (1) is adapted from Koza (1992b).

$$F_1(i,m) = \sum_{n=1}^{|m|} \delta(S(i,x_n), y_n) \tag{1}$$

whereby:

(1) $F_1(i, m)$ is i's OF score.
(2) i is the candidate solution.
(3) m is the fitness case training set defined for the given problem. $|m|$ is the size of m.
(4) x_n is the nth fitness case in m.
(5) $S(i, x_n)$ is the output value returned by i for x_n.
(6) y_n is the target value for x_n.
(7) In quantitative problems: $\delta(S(i,x_n), y_n) = -|S(i,x_n) - y_n|$ i.e. the negative absolute difference between $S(i, x_n)$ and y_n (the negative sign induces maximization).
In qualitative problems: $\delta(S(i,x_n), y_n) = \begin{cases} 1, & \text{if } S(i,x_n) = y_n. \\ 0, & \text{if } S(i,x_n) \neq y_n. \end{cases}$

Fig. 1 shows an example of the OF calculation for a candidate solution, i, in a simple Boolean (qualitative) problem with two inputs, A1 and A2. The same problem and candidate solution will be used for all example fitness calculations shown in this section.

3.2. Behavioural programming

Behavioural programming (BP) is motivated by the observation that OF abstracts the useful information contained in the internal