Surrogate Genetic Programming: A semantic aware evolutionary search

CrossMark

Ahmed Kattan [a,*], Yew-Soon Ong [b]

[a] AI Real-world Application Lab, Um Al-Qura University, Saudi Arabia
[b] Computer Science Department in Nanyang Technological University, Singapore

ABSTRACT

Many semantic search based on Genetic Programming (GP) use a trial-and-error scheme to attain semantically diverse offspring in the evolutionary search. This results in significant impediments on the success of semantic-based GP in solving real world problems, due to the additional computational overheads incurred. This paper proposes a surrogate Genetic Programming (or sGP in short) to retain the appeal of semantic-based evolutionary search for handling challenging problems with enhanced efficiency. The proposed sGP divides the population into two parts (μ and λ) then it evolves μ percentage of the population using standard GP search operators, while the remaining λ percentage of the population are evolved with the aid of meta-models (or approximation models) that serve as surrogate to the original objective function evaluation (which is computationally intensive). In contrast to previous works, two forms of meta-models are introduced in this study to make the idea of using surrogate in GP search feasible and successful. The first denotes a "Semantic-model" for prototyping the semantic representation space of the GP trees (genotype/syntactic-space). The second is a "Fitness-model", which maps solutions in the semantic space to the objective or fitness space. By exploiting the two meta-models collectively in serving as a surrogate that replaces the original problem landscape of the GP search process, more cost-effective generation of offspring that guides the search in exploring regions where high quality solutions resides can then be attained. Experimental studies covering three separate GP domains, namely, (1) Symbolic regression, (2) Even n-parity bit, and (3) a real-world Time-series forecasting problem domain involving three datasets, demonstrate that sGP is capable of attaining reliable, high quality, and efficient performance under a limited computational budget. Results also showed that sGP outperformed the standard GP, GP based on random training-set technique, and GP based on conventional data-centric objectives as surrogate.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Evolutionary algorithms (EAs) are approaches that take their inspirations from the principles of natural selection and survival of the fittest in the biological kingdom. Among the many variants of EAs, Genetic Programming (GP) is among one of those that have withstood the realms of time with success stories reported in a plethora of real-world applications. In

* Corresponding author.
E-mail addresses: ajkattan@uqu.edu.sa (A. Kattan), asysong@ntu.edu.sg (Y.-S. Ong).

particular, GP has been deemed as capable of providing transparency into how decisions or solutions are made. While a technique to evolve trees was suggested by Cramer in 1985 [6], the field of GP was founded by John Koza in 1992 [16]. GP is a powerful learning engine inspired by biology and natural evolution, for automatically generating working computer programs. Based on Darwins theory of evolution, computer programs are measured against the task they are intended to do and then receive scores accordingly. The programs with the highest scores are considered the fittest. The fittest programs are selected to join the evolutionary process via three standard genetic operators: crossover, mutation and reproduction. These operators aim to amend the programs structure and create new offspring, which will hopefully be better. Computer programs are treated as a collection of functions and inputs for these functions; which can be seen as their genetic material. In the standard representations of GP, programs are represented as trees where the functions appear in the internal nodes of the tree and the inputs for the functions appear on the leaves. This representation flexibility adds an extra advantage to the GP since it can solve complex problems and even comes up with solutions beyond human thoughts in some cases.

In GP, the search space can be viewed from multiple facets; (A) Structural space or Syntactic space or Genotype in which most GP systems operate on, in attempting to evolve/alter the trees, via the genetic operators, hoping to find better ones that translate to desirable behaviours, (B) Phenotype or Semantic space where GP systems try to alter the behaviours of the programs directly, and finally, and (C) Fitness space where individuals are evaluated into numerics that quantify the quality in relation to solving the given problem. In general, it is common for the semantic space of GP to be represented in the form of real number vectors and are defined by the outputs of GP trees when their instructions (or functions) are executed.

In traditional GP, although remarkable success on different real-world problems have been achieved (e.g., [36,14]), existing systems operate fundamentally within the syntactic space, ignoring the semantic information of the candidate GP trees or programs. Advancing studies on semantic GP have highlighted the potentials of using the semantic information within the search. In particular, incorporations on semantic information of candidate solutions into the GP evolutionary process have recently been reported to generate significant enhancements in the search performances [33,4,26]. Generally, the term semantic refers to the chromosomes or individuals' behaviour (for example, represented in the form of a real-valued vector), while syntactic refers to the structure of the evolved program. In most real-world applications, if not all, there is no obvious relationship between the syntactic space and semantic space. Thus, small changes in the shape of a tree can result in a significant departure in the resultant behaviour and vice versa, any small deviations requested in the program behaviour (or semantic) would need major variations in the tree structure. Due to the complex structure of GP in the syntactic space, i.e., tree-like structure representation, it is hard to identify a suitable syntactic distance measure that correlates well with the fitness landscape. In other words, it is hard to define a distance (or similarity measure) that quantifies the structural similarities between two trees and at the same time quantifies the similarities between their fitness values. Even if such distance measure is available, it is mostly problem-dependent. Generally, it is often deemed easier to determine the distance that correlates fitness landscape between individuals' semantics (where semantics represented as vectors of real numbers) than between individuals' syntactic (i.e., tree-like structure representation). This is because it is easier to use a generic distance metrics that applies across problem domains between vectors than trees. For these reasons, it is often argued that the semantic space is easier to search upon, than the syntactic space.

Despite the increasing research interests and potentials of semantic GP, one of the main criticism has been on the excessive slow nature of the approach, attributed not only to the natural mechanisms of evolution which involves an iterative process of candidate solutions that evolves across many generations, but more importantly, the trial-and-error scheme used to facilitate semantically diverse offspring in the search have led to significant increase in the computational resources needed before convergence to credible and reliable results can be attained. In this work, our interest is to retain the appeal of GP algorithms, especially semantic GP, that can handle challenging problems with high quality designs at enhanced computational efficiency.

We present a study on surrogate Genetic Programming or sGP in short. The core characteristics and motivations for proposing the sGP can be summarised as follows: (1) Present Semantic GPs heavily follow a trial-and-error scheme [25,34,11], which led to highly computationally intensive search. (2) It is desirable to conduct a search using Semantic GPs for high quality solutions under a limited computational budget. (3) It is non-trivial to map from syntactic to the semantic space [25]. Particularly, in contrast to previous studies on semantic based operators, where efforts have been placed on exploiting individuals' behaviour to maintain semantic diversity in the search population, our study on sGP in this paper differs in the use of meta-models for exploring the semantic space, so as to enhance search efficiency through eliminating the need to evaluate each computationally intensive candidate solutions completely.

The proposed model of sGP divides the population into two parts μ and λ then it evolves μ portion of the population using standard search operators and the original objective function, while the remaining λ population via the surrogate. On the λ portion of the population, sGP uses two forms of meta-models, namely the "semantic-model" and "fitness-model" as the surrogate that aids in the GP search. The semantic-model prototypes the semantic space of the GP trees (genotype/syntactic-space), while the fitness-model maps the semantic space to the fitness space (more details are given in Section 3). In order to construct the semantic space, it is generally necessary to evaluate the GP programs completely. But this is often deemed as computationally intensive and impractical for many real-world complex applications. By using the semantic-model, on the other hand, solution individuals only need to be partially evaluated and the complete behaviours of the GP trees (i.e., fitness value) is then predicted via the fitness-model. The semantic and fitness-models thus serve as a partial replacement or surrogate of the computationally expensive objective function in semantic GP, leading to enhanced search efficiency. As will be shown in the results section, sGP is capable of achieving good solutions with smaller numbers of function evaluations.