Information Sciences 235 (2013) 195–213

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

On the roles of semantic locality of crossover in genetic programming

Nguyen Quang Uy [a], Nguyen Xuan Hoai [b,*], Michael O'Neill [c], R.I. McKay [d], Dao Ngoc Phong [e]

[a] Faculty of Information Technology, Military Technical Academy, Viet Nam
[b] Faculty of Information Technology, Hanoi University, Viet Nam
[c] Natural Computing Research and Applications Group, University College Dublin, Ireland
[d] School of Computer Science and Engineering, Seoul National University, South Korea
[e] Department of Information and Communication, Hanoi City Government, Viet Nam

ARTICLE INFO

Article history:
Received 5 June 2012
Received in revised form 4 January 2013
Accepted 8 February 2013
Available online 16 February 2013

Keywords:
Genetic Programming
Semantic
Locality
Crossover

ABSTRACT

Locality has long been seen as a crucial property for the efficiency of Evolutionary Algorithms in general, and Genetic Programming (GP) in particular. A number of studies investigating the effects of locality in GP can be found in the literature. The majority of the previous research on locality focuses on syntactic aspects, and operator semantic locality has not been thoroughly tested. In this paper, we investigate the role of semantic locality of crossover in GP. We follow McPhee in measuring the semantics of a subtree using the fitness cases. We use this to define a semantic distance metric. This semantic distance supports the design of some new crossover operators, concentrating on improving semantic locality. We study the impact of these semantically based crossovers on the behaviour of GP. The results show substantial advantages accruing from the use of semantic locality.

1. Introduction

Genetic Programming (GP) [38,28] is an evolutionary paradigm for finding solutions, often in the form of functional expressions, for a problem. To date, research has largely focused on syntactic aspects of GP representation, bringing valuable insights and contributions to the success of GP. However, from a programmer's perspective, maintaining syntactic correctness is just a part of program construction: programs must be correct not only syntactically, but also semantically. Thus incorporating semantic awareness in the GP evolutionary process could potentially improve performance and extend the applicability to problems that are difficult to deal with using purely syntactic approaches. Recently, several researchers have taken an interest in incorporating semantic information into GP, leading to a sharp increase in related publications (e.g. [19,21,24,2,34,49,5]).

Previous work in the broader field of evolutionary computation has shown that locality (i.e., small changes in genotype resulting in small changes in phenotype) deeply affect search performance [16,42,41]. With current GP representations, designing GP operators supporting this desirable property can be difficult. An important reason is the lack of a clear separation between syntax and semantics of individuals. Most GP genetic operators have been designed based on syntax alone; but small changes in syntax can lead to large changes in semantics. Thus ensuring semantic locality requires genetic operators that are semantically aware.

* Corresponding author.
E-mail addresses: quanguyhn@gmail.com (N.Q. Uy), nxhoai@gmail.com (N.X. Hoai), m.oneill@ucd.ie (M. O'Neill), rimsnucse@gmail.com (R.I. McKay), phongaptech@yahoo.com (D.N. Phong).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.02.008

In GP, most research on locality has focused on syntax [16,42,41]. Recently, semantic diversity has attracted some attention [2,46,13], but limited work has been conducted on semantic locality [49]. In this paper, we extend our earlier work in [49] on incorporating semantics into GP crossover operators. The earlier paper introduced the use of semantic locality to generate controllable search. Here, we propose two improvements to the crossover operator of [49]. We also further investigate the effects of these operators on locality. In [49], we argued that semantic locality is a desirable property of crossover. However it has previously been argued that syntactic locality is valuable [16]. Perhaps semantic locality's value merely reflects its correlation with syntactic locality. Here, we extend this work to show that semantic locality is substantially more beneficial than syntactic locality. We apply these methods to two real-world problems in a time series prediction task, and in producing human-competitive approximations for the Gaussian *Q*-function.

The remainder of the paper is structured as follows. The next section reviews literature on semantic information in GP, on crossover operators, and on locality. Section 3 contains a detailed descriptions of our methods. The potential effects on GP performance of improving semantic locality in crossover are investigated in Section 4, followed by an analysis of semantic exchange in standard crossover and semantic based crossovers (Section 5). The comparison between semantic locality and syntactic locality is presented in Section 6. Section 7 presents two application of these ideas to two problems: time series prediction and approximating the Gaussian *Q*-function. The conclusions are drawn in Section 8, leading to suggestions for future research in Section 9.

2. Background

This section gives a brief overview of previous work on semantics in GP, on variants of GP crossover operators, and on locality in GP.

2.1. Semantics in Genetic Programming

In GP, semantic information has generally been used to provide additional guidance to the evolutionary search. The way semantics are exploited depends on the problem domain (Boolean or real-valued), individual representation (Grammar-, Tree- or Graph-based), and the search algorithm components (fitness measure, genetic operators, ...). There have been three main approaches for representing, extracting, and using semantics to guide the evolutionary process of GP:

1. Grammar-based [52,5,6].
2. Formal methods [19,21,25].
3. GP s-tree representation [2,34,46,48,29].

The first, using attribute grammars, is the most popular. Such grammars extend context-free grammars, providing context sensitivity through a finite set of attributes [26]. GP individuals expressed in the form of attribute grammar derivation trees can incorporate semantic information, which can be used to eliminate bad (i.e., less fit) individuals from the population [6] or to prevent the generation of semantically invalid individuals [52,5]. The attributes used to present semantics are generally problem-dependent, and it is not always obvious how to determine the attributes for a given problem.

Johnson, in his recent work, has advocated formal methods as a means to incorporate semantic information into the GP process [19,21]. In this work, semantic information, extracted by formal methods, is used to quantify the fitness of individuals on some problems for which traditional sample-point-based fitness measure are unavailable or misleading.

In other work, Keijzer [25] used interval analysis to check whether an individual is defined over the whole range of input values – if an individual is undefined anywhere, that individual can be assigned minimal fitness or simply eliminated from the population. This allowed Keijzer to avoid discontinuities arising from protected operators, improving the evolvability of the system. The advantage of formal methods lies in their rigorous mathematical foundations, potentially helping GP to evolve computer programs. However they are high in complexity and difficult to implement, possibly explaining the limited number of related publications since the advocacy of Johnson [20]. Their main application to date has been in evolving control strategies.

Methods for extracting semantics from expression trees depend on the problem domain. The finite inputs of Boolean domains mean that semantics can be accurately estimated in a number of ways. Beadle and Johnson [2] checked the semantic equivalence of the offspring produced by crossover with their parents by converting them to Reduced Ordered Binary Decision Diagrams (ROBDDs). If the conversion of the two trees leads to the same ROBDD, they are semantically equivalent. This semantic equivalence checking is used to decide if new individuals are copied to the next generation. If offspring are equivalent to their parents, they are discarded and the crossover is restarted. The authors argued that this enforces semantic diversity in the evolving population, and consequently leads to improvement in GP performance [2].

By contrast, in [34], semantic information is extracted from a Boolean expression tree by enumerating all possible inputs. Two aspects of semantics were evaluated in each tree: the semantics of subtrees and the semantics of context (the remainder of an individual after removing a subtree). The variation of these semantic components throughout the GP evolutionary process was experimentally measured. Special attention was paid to fixed-semantic subtrees: subtrees where the semantics of the tree does not change when this subtree is replaced by another subtree. The authors showed that there may be many such fixed semantic subtrees when the tree size increases during the evolutionary process. Thus it becomes increasingly difficult