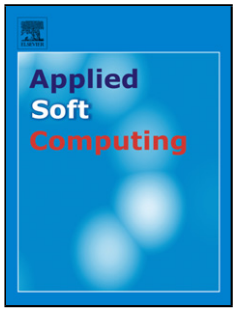

Title: Co-evolutionary Multi-Population Genetic Programming for Classification in Software Defect Prediction: an Empirical Case Study

Author: Goran Mauša Tihana Galinac Grbac

PII: S1568-4946(17)30065-0
DOI: http://dx.doi.org/doi:10.1016/j.asoc.2017.01.050
Reference: ASOC 4044

To appear in: *Applied Soft Computing*

Received date: 8-12-2016
Revised date: 18-1-2017
Accepted date: 26-1-2017

Please cite this article as: Goran Mauša, Tihana Galinac Grbac, Co-evolutionary Multi-Population Genetic Programming for Classification in Software Defect Prediction: an Empirical Case Study, *<![CDATA[Applied Soft Computing Journal]]>* (2017), http://dx.doi.org/10.1016/j.asoc.2017.01.050

Co-evolutionary Multi-Population Genetic Programming for Classification in Software Defect Prediction: an Empirical Case Study

Goran Mauša[a,*], Tihana Galinac Grbac[a]

[a] *Vukovarska 58, 51000 Rijeka, Faculty of Engineering, University of Rijeka, Croatia*

Abstract

Evolving diverse ensembles using genetic programming has recently been proposed for classification problems with unbalanced data. Population diversity is crucial for evolving effective algorithms. Multilevel selection strategies that involve additional colonization and migration operations have shown better performance in some applications. Therefore, in this paper, we are interested in analyzing the performance of evolving diverse ensembles using genetic programming for software defect prediction with unbalanced data by using different selection strategies. We use colonization and migration operators along with three ensemble selection strategies for the multi-objective evolutionary algorithm. We compare the performance of the operators for software defect prediction datasets with varying levels of data imbalance. Moreover, to generalize the results, gain a broader view and understand the underlying effects, we replicated the same experiments on UCI datasets, which are often used in the evolutionary computing community. The use of multilevel selection strategies provides reliable results with relatively fast convergence speeds and outperforms the other evolutionary algorithms that are often used in this research area and investigated in this paper. This paper also presented a promising ensemble strategy based on a simple convex hull approach and at the same time it raised the question whether ensemble strategy based on the whole population should also be investigated.

Keywords: Genetic Programming, Classification, Coevolution, Software Defect Prediction

1. Introduction

Software defect prediction (SDP) is an important decision support activity in software quality assurance for large and complex software systems. Its goal is to improve the allocation of testing resources (and consequently, costs) by identifying defect-prone software components in advance. SDP datasets are collections of measurements performed for each software component (file, class, method), and each component is represented by the number of static code attributes (like cyclomatic complexity, lines of code, number of methods, etc.) and the number of defects identified and corrected on that component. Such datasets are used to build models to predict defective components for a subsequent software release and to adjust the development and verification strategy accordingly.

Defects in complex software systems usually behave according to the Pareto principle, meaning that the majority of defects (approximately 80%) are concentrated in a small proportion of system components (approximately 20%). It seems that this simple empirical principle is universally valid for all software systems [1]. On the other hand, software defects generally do not follow any particular probability distribution that could provide a mathematical model [2]. A systematic literature survey performed by [3] did not find a widely applicable SDP model despite a number of studies that aimed to find the best performing model, generally speaking. The main reason lies in the very nature of software datasets and their imbalance, complexity and properties that seem to be dependent on the environmental conditions and application domain. Class distributions are highly skewed, which is con-

*Corresponding author
Email addresses: gmausa@riteh.hr (Goran Mauša), tgalinac@riteh.hr (Tihana Galinac Grbac)