



Dynamic population variation in genetic programming

Peyman Kouchakpour*, Anthony Zaknich, Thomas Bräunl

School of Electrical, Electronic and Computer Engineering, University of Western Australia, Nedlands, Perth, WA 6009, Australia

ARTICLE INFO

Article history:

Received 12 April 2007

Received in revised form 18 July 2008

Accepted 17 December 2008

Keywords:

Genetic programming
Computational effort
Average number of evaluations
Convergence
Diversity
Population variation
Dynamic population variation

ABSTRACT

Three innovations are proposed for dynamically varying the population size during the run of the genetic programming (GP) system. These are related to what is called Dynamic Population Variation (DPV), where the size of the population is dynamically varied using a heuristic feedback mechanism during the execution of the GP with the aim of reducing the computational effort compared with Standard Genetic Programming (SGP). Firstly, previously developed population variation pivot functions are controlled by four newly proposed characteristic measures. Secondly, a new gradient based pivot function is added to this dynamic population variation method in conjunction with the four proposed measures. Thirdly, a formula for population variations that is independent of special constants is introduced and evaluated. The efficacy of these innovations is examined using a comprehensive range of standard representative problems. It is shown that the new ideas do have the capacity to provide solutions at a lower computational cost compared with standard genetic programming and previously reported algorithms such as the plague operator and the static population variation schemes previously introduced by the authors.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

The automatic computational solution of problems has been central to Artificial Intelligence (AI) and machine learning. In his 1959 paper [43], Arthur Samuel posed the question of, “How can computers learn to solve problems without being explicitly programmed? In other words, how can computers be made to do what is needed to be done, without being told exactly how to do it?” genetic programming (GP), which is an automatic domain-independent method, has addressed such questions with a good measure of success. It has been applied in various branches of engineering and sciences including biomedical science [17,19,21,35,44,47,53], classification tasks [14,27,31,49,50,54,55,57], navigation tasks [1,4,36], image processing and pattern recognition [9,10,22,28,57], neural networks [1,7,36,39] and robotics [23,32,33,40], and in many other various applications and disciplines [8,12,18,20,34,40,41,45,46,51,56], to name but a few. However, one of the main drawbacks of GP has been the often large amount of computational effort required to solve complex problems. Researchers have investigated various methods for overcoming this problem. This issue of computational effort was addressed in [26] by introducing a number of different methods for improving GP execution time. While such investigations were mainly focused on the design and realization of the GP platform from a software implementation point of view, many other researchers have considered techniques to improve efficiency from the GP paradigms perspective. Some of these included the development of multi-objective methods for tree size control [11], the development of the island-model [15] controlling the bloat phenomenon and the Evolutionary Module Acquisition (MA) [3] and many other practices such as reported in [2,24,38,52], to name but a few.

Various researchers have investigated the effects of population size on the GP algorithm and have reported a relationship between the population size and computational effort. A brief study on population size was conducted in [30], where the

* Corresponding author.

E-mail address: peyman_kouch@yahoo.com.au (P. Kouchakpour).

Table 1
Population variation vs Dynamic Population Variation.

Algorithm	Acronym	Description
Population variation	PV	It employs a deterministic adaptation using simple time-varying schedules. The population is changed using a deterministic function. The population profile is known prior to the run, hence termed as “static”
Dynamic Population Variation	DPV	It employs a feedback mechanism. By using the actual progress made, the population profile is “dynamically” changed during the run. The population profile is unknown prior to any given run

population size was altered prior to the GP run and was maintained at a constant level throughout the run. The study concluded that larger population size M increased the cumulative probability $P(M, i)$ of satisfying the success predicate of a problem for GP (for generations between generation 0 and generation i), as long as the population size was below the threshold at which the cost of a larger population (in terms of individuals to be processed) outweighed the benefits it introduced. In [16,29] the population size was varied during the run resulting in computational effort savings. In [16], the plague operator was introduced, where the population size was reduced at a linear rate. It was shown that plague allowed a given fitness level to be reached with a smaller computational effort. In [29], we introduced the population variation (PV) scheme, where the population could be increased and/or decreased with a variable profile, where the increment or reduction of population size could take on any flexible profile such as linear, exponential, hyperbolic, sinusoidal or even random. We demonstrated that PV significantly improved performance and showed that the optimum profile was dependant on the problem domain, which was ultimately driven by the fitness landscape. The two investigations mentioned above involved what can be referred to as static population variation in genetic programming. Static population variation employs a deterministic adaptation approach using simple time-varying schedules, where the population size is varied according to a deterministic function $P(g)$. The function $P(g)$ is the Population-Generation profile, which determines the total number of individuals (the population size) at any generation g . One of the shortcomings of the static population variation scheme is that the population size is varied by a blind deterministic function. It is more desirable to vary the population size in an informed way during the run. Using a heuristic feedback mechanism, the population size can be dynamically varied by taking into account the actual progress of GP in solving the problem. Tomassini et al. [48] introduced a technique to dynamically vary the size of the population during the execution of the GP system. The population size was varied “on the run” according to some particular events occurring during the evolution. Table 1 clearly defines and summarizes the difference between population variation and dynamic population variation.

In this paper, we propose various new ways to dynamically vary the population size during the run of the GP system. Our proposed approach, which we refer to as Dynamic Population Variation (DPV), extends the work of Tomassini et al. [48] and proposes new modifications to it. We show in this paper that our proposed way of dynamically varying the population is superior to Standard Genetic Programming (SGP), the plague operator [16], the dynamic population modification technique in [48] and all the static PV schemes reported in [29]. The paper is structured as follows. Section 2 discusses dynamic population variation together with the proposed modifications and highlights previously published errors. The applications which have been used in this study are briefly described in Section 3. In Section 4, experimental results are presented and analysed for the proposed DPV approach and comparisons are made with the existing population variation method reported in [48], SGP, plague operator and our previously documented population variation PV scheme, followed by conclusions in Section 5.

2. Dynamic population variation

A technique to dynamically vary the size of the population during the execution of the GP system was introduced in [48]. According to this technique the population size is dynamically increased if the change in fitness improvement is less than some functional value, called the pivot ρ as defined by (1)–(3), and reduced if this change is more than the pivot. The motivation is to add new individuals when the GP system is reaching a stagnation phase and remove individuals when the GP process is progressing well. The dynamic modification of the population size is described in [48] as a pseudo-algorithm, which we summarize by (1)–(3), showing the respective calculations of the pivot functions. In the first pivot function (1), the pivot is calculated by the main operation of division of the previous generation delta (Δ) function by the current generation delta (Δ) function, hence known as the DIV (Division) pivot function. In the second pivot function (2), the pivot is calculated by the main operation of subtraction of the current generation delta function from the previous generation delta function, hence known as the SUB (Subtraction) pivot function

$$\rho_{DIV}^{<jj+T-1>} = \frac{1}{T} \times \sum_{i=j}^{j+T-1} \begin{cases} \frac{\Delta_{i-1}}{\Delta_i}, & \Delta_i \neq 0, \\ 1, & \Delta_i = 0, \end{cases} \quad (1)$$

$$\rho_{SUB}^{(jj+T-1)} = \frac{1}{T} \times \sum_{i=j}^{j+T-1} (\Delta_{i-1} - \Delta_i), \quad (2)$$

$$\Delta_i = f_{i-1} - f_i, \quad (3)$$

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات