

Combining Cartesian Genetic Programming with an Estimation of Distribution Algorithm

Genetic Programming Track

Janet Clegg
Intelligent Systems Group,
Department of Electronics
University of York, Heslington
York, YO10 5DD, UK
jc@ohm.york.ac.uk

ABSTRACT

This paper describes initial testing of a novel idea to combine a CGP with an EDA. In recent work a new improved crossover technique was successfully applied to a CGP. To implement the new method meant changing the traditional CGP representation. The new representation developed in that work lends itself very nicely to some probability distribution being implemented. The work in this paper has investigated this idea of incorporating estimated probability distributions into the new CGP method with crossover.

Categories and Subject Descriptors

I.2.2 [Artificial Intelligence]: Automatic Programming—*Program synthesis*; I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods and Search

General Terms

Algorithms, Design, Performance

Keywords

Cartesian Genetic Programming, optimization, crossover techniques

1. INTRODUCTION

A new crossover technique [1] for Cartesian Genetic Programming (CGP) was recently introduced by Janet Clegg. The CGP representation was modified in order to enable the crossover technique to be applied. In a CGP [2] programs are represented by strings of integers, where each integer denotes a certain choice of function or input; in the new representation programs were instead represented by strings of parameters whose values lie in a real valued interval. The choice of function was then determined by the

value of the parameters along this interval. Crossover could then be performed as in a traditional GA. In initial tests [1] it was shown that this new technique made the optimisation around three times more successful in finding solutions than previous CGP methods.

This paper describes initial tests of a novel idea of combining the new crossover technique with an Estimation of Distribution Algorithm (EDA). Since the new CGP representation is made up of a set of parameters whose values lie in a real valued interval and function choices are made relative to the value of the parameters along this interval, its very representation lends itself to some probability distribution being implemented. Rather than splitting the interval into equal parts where each part represents a certain function (such that each function is equally likely to be chosen) the interval can be split into unequally sized parts whose sizes vary according to some probability distribution. This paper describes results from implementing this new idea of integrating EDA techniques into the new crossover method. The most similar work to that described here can be found in [3], [4].

2. THE METHOD

The ideas investigated within this paper have been tested on two polynomial regression problems whose equations are given by Equations 1 and 2 below.

$$x^6 - 2x^4 + x^2 \quad (1)$$

$$x^7 - 2x^5 + x^3 \quad (2)$$

A sample of fifty data points are taken from the interval $[-1,1]$, and the cost function is defined as the sum of the absolute values of the differences between the population member's values and the true function values at each of the data points. The algorithm is classed as converged when all of these absolute values are less than 0.01. A population size of 50 has been used with 48 offspring created at each generation. Tournament selection has been chosen to select the parents and the maximum number of nodes in the CGP representation has been set at 10. The algorithm has been run 1000 times and the average convergence has been recorded at each generation. In the figures in this paper, the horizontal axis represents generation number and the vertical axis is the cost function for the best member of the population (averaged over 1000 runs) for that particular

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
Copyright 2008 ACM 978-1-60558-130-9/08/07...\$5.00.

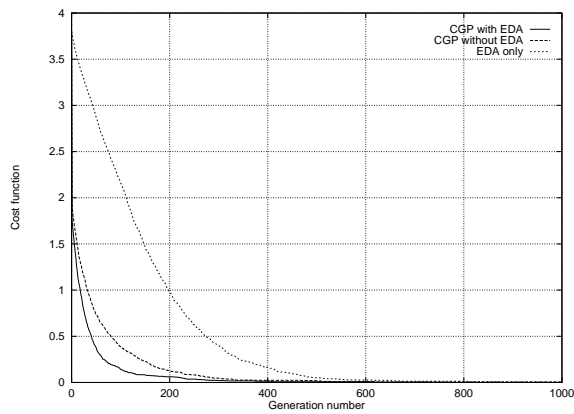


Figure 1: Comparing the CGP with EDA and the CGP without EDA for $x^6 - 2x^4 + x^2$ using the optimised probability increment

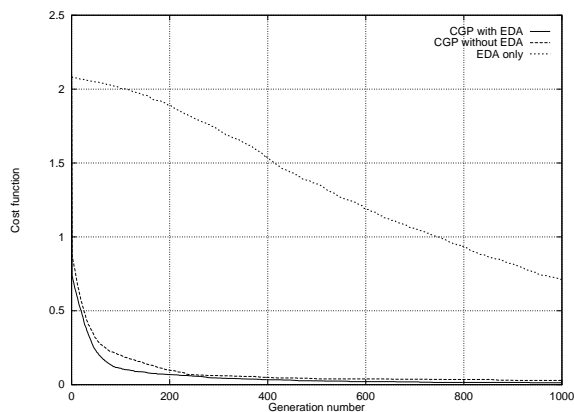


Figure 2: Comparing the CGP with EDA and the CGP without EDA for $x^7 - 2x^5 + x^3$ using the optimised probability increment

generation number. The quality of each different method is determined by its speed of convergence.

The simplest form of EDA is called a Univariate Marginal Distribution Algorithm (UMDA) and is one in which the genes are assumed to have probability distributions which are independent of each other. In this paper we use this simple EDA. The CGP with EDA begins in the first generation assuming that each possible value for each gene is equally likely, i.e. each gene has a uniform probability distribution. As the generations progress these probabilities are incremented in line with the number of times each value of the gene occurs in good members of the populations.

One important aspect to consider in this, is by how much to increment the probability distribution at each generation. One would assume that for the initial generations the predicted probabilities will be less reliable than towards the latter generations. Several rates of incrementing the probabilities have been investigated, and these rates are 0.01, 0.001, 0.0001, 0.00001, 0.000001, 0.0000001 and 0.00000001. As a simple example of how the probability distributions are incremented suppose a gene has two possible values (0 or 1, say). The particular algorithm with an incremental rate of 0.001 will begin with a uniform distribution for these two

values; the probability of a 0 and the probability of a 1 are both set initially to 500/1000. If 20 parents are selected and 15 of these parents have a value 0 and 5 of these have a value 1, then the probabilities are incremented to 515/1020 and 505/1020. If at the next generation 20 more parents are selected where 18 have a value 0 and only 2 have value 1, then the probabilities are incremented to 533/1040 and 507/1040. The process carries on through the generations such that, as it converges, the probabilities converge towards a 1 for the optimal value of the gene and zeros for the other values.

Figures 1 and 2 display results from running the combination of CGP and EDA for the regression problems given by Equations 1 and 2 respectively. An increment rate of 0.0000001 was found to perform best for the probability distribution updates. In both figures, the performance is compared to the CGP with crossover but without an EDA element to it. Both figures also display the best convergence of the simple EDA without CGP for comparison. An increment rate of 0.001 was found to perform best for the EDA without CGP. The integration of an EDA into the new CGP with crossover has successfully improved convergence when the EDA increments the probabilities very slowly as the generations progress. The addition of this EDA element to the new crossover technique is also very easily implemented and therefore is a valuable result.

3. CONCLUSIONS

This paper has extended the work in [1] which introduced a new crossover technique for CGP. The new crossover technique lends itself easily to the integration of an EDA technique working alongside it. This is because of the nature of the new representation for the crossover method. Each possible value for the variables are assigned to parts of a real valued interval. In [1] the whole interval was divided into equal sections, each section representing a choice of value for the variable. In this new work this interval has been split into sections whose sizes depend on estimated probability distributions. It has been shown that the addition of this EDA element within the CGP technique improves its performance.

4. REFERENCES

- [1] J. Clegg, J.A. Walker, J.F. Miller. A new crossover technique for Cartesian Genetic Programming. In *Proceedings of the 2007 Genetic and Evolutionary Computation Conference*, pages 1580–1587, London, 2007.
- [2] J. F. Miller and P. Thomson. Cartesian genetic programming. In *Proceedings of the 3rd European Conference on Genetic Programming (EuroGP 2000)*, volume 1802 of *Lecture Notes in Computer Science*, pages 121–132, Edinburgh, 2000. Springer-Verlag.
- [3] A. Ratle and M. Sebag. Avoiding the bloat with stochastic grammar-based genetic programming. In *Artificial Evolution - Lecture notes in computer science vol 2310*, pages 255–266, Springer 2002.
- [4] R. Salustowicz and J. Schmidhuber. Probabilistic incremental program evolution. In *Evolutionary Computation*, (5) pages 123–141, 1997.