

Linear genetic programming

**Springer Science+Business Media, Markus Brameier
and Wolfgang Banzhaf, 2007, 315 pp, Book Series: Genetic
Programming, Hard Cover, €62.95, ISBN 0-387-31029-0**

Vic Ciesielski

Published online: 15 August 2007
© Springer Science+Business Media, LLC 2007

This is a very good book. It provides a well organized description of linear genetic programming (LGP). Much material, previously only available in research papers, has been consolidated, reorganized and extended for this book. It provides an excellent treatment of the theoretical and empirical aspects of LGP. The book is generally well written and easy to follow.

LGP is a variant of genetic programming in which the evolved individuals are programs in a specialized assembly language for a register based machine. Instructions for the machine typically involve 3 operand registers. Register 0 is designated as the output register and the contents of this register are the output of the program at termination. The term “linear” is used because the programs are in an imperative language and instructions are executed in a linear fashion, in contrast to tree based genetic programming in which a recursive traversal of a tree is required.

The book consists of 11 chapters divided into 3 parts, with each part going into more depth and detail. Readers can easily choose the depth to which they want to study the material. Readers interested only in the basic concepts of LGP would stop at part I. Part I, Fundamental Analysis, consists of three chapters two of which describe the basic concepts of LGP and the third gives a comparison with neural networks on a number of classification problems.

Part II, Method Design, consists of 5 chapters. Four chapters describe a systematic exploration of the various design and configuration choices that are possible in LGP. The chapters have a uniform structure that works well. In the early sections of a chapter particular design choices are discussed and questions posed about which choice might be best. The questions are then answered by the appropriate experiments in the second part of the chapter. For example, in the first

V. Ciesielski (✉)
School of Computer Science and Information Technology, RMIT University,
Melbourne, VIC, Australia
e-mail: vc@cs.rmit.edu.au

part of chapter 5, a number of different recombination approaches for linear programs are described and in the second part of chapter 5 comparison experiments are presented.

The last chapter in this section gives a comparison between LGP and tree based genetic programming on some GP benchmark problems and classification problems from the UCI repository. Interestingly, in terms of training error and classification error rate, LGP is superior on every problem, sometimes by a considerable margin. The reasons for this are not explored and it is perhaps a challenge to proponents of tree based genetic programming to explain these results.

Part III, Advanced Techniques and Phenomena, consists of the final three chapters. These chapters have the same structure as those in part II and address a number of more esoteric issues in LGP. These include control of diversity and variation step size, code growth and neutral variations, and program teams or classifier ensembles.

While the concepts of LGP are generally well described and the arguments about the usefulness of LGP are well developed, I have reservations about some aspects. All of the empirical work is on classification problems. Genetic programming classifiers have some disadvantages, that is, long training times, they don't scale well to large training data sets and the classifiers are not very understandable. Other kinds of problems are better suited to genetic programming. While the book is not primarily about classification it may give the impression that LGP is only for classification problems. Also, the use of multi-objective techniques in LGP has not really been explored. These techniques have proved to be very beneficial in tree based genetic programming, in terms of evolving small, fit programs and could also provide benefits to LGP.

In looking at the book overall, I wonder if the authors have missed an opportunity to have a book for a wider audience. Including an introductory chapter on evolutionary computing might have made the book readable by the general computer scientist, in addition to the evolutionary computing researcher.

Who would benefit from buying this book? This book is primarily for the evolutionary computing researcher who wants to know more about LGP or is interested in research in the area. Post graduate students in genetic programming should read this book even if they are working in another paradigm. The book should be in university libraries. Considering the cost of many books these days this one is very well priced for its size and content.