Cândida Ferreira

Gene Expression Programming

Studies in Computational Intelligence, Volume 21

Editor-in-chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series
can be found on our homepage:
springer.com

Vol. 5. Da Ruan, Guoqing Chen, Etienne E. Kerre, Geert Wets (Eds.)
Intelligent Data Mining, 2005
ISBN 3-540-26256-3

Vol. 6. Tsau Young Lin, Setsuo Ohsuga, Churn-Jung Liau, Xiaohua Hu, Shusaku Tsumoto (Eds.)
Foundations of Data Mining and Knowledge Discovery, 2005
ISBN 3-540-26257-1

Vol. 7. Bruno Apolloni, Ashish Ghosh, Ferda, Alpaslan, Lakhmi C. Jain, Srikanta Patnaik (Eds.)
Machine Learning and Robot Perception, 2005
ISBN 3-540-26549-X

Vol. 8. Srikanta Patnaik, Lakhmi C. Jain, Spyros G. Tzafestas, Germano Resconi, Amit Konar (Eds.)
Innovations in Robot Mobility and Control, 2005
ISBN 3-540-26892-8

Vol. 9. Tsau Young Lin, Setsuo Ohsuga, Churn-Jung Liau, Xiaohua Hu (Eds.)
Foundations and Novel Approaches in Data Mining, 2005
ISBN 3-540-28315-3

Vol. 10. Andrzej P. Wierzbicki, Yoshiteru Nakamori
Creative Space, 2005
ISBN 3-540-28458-3

Vol. 11. Antoni Ligęza
Logical Foundations for Rule-Based Systems, 2006
ISBN 3-540-29117-2

Vol. 13. Nadia Nedjah, Ajith Abraham, Luiza de Macedo Mourelle (Eds.)
Genetic Systems Programming, 2006
ISBN 3-540-29849-5

Vol. 14. Spiros Sirmakessis (Ed.)
Adaptive and Personalized Semantic Web, 2006
ISBN 3-540-30605-6

Vol. 15. Lei Zhi Chen, Sing Kiong Nguang, Xiao Dong Chen
Modelling and Optimization of Biotechnological Processes, 2006
ISBN 3-540-30634-X

Vol. 16. Yaochu Jin (Ed.)
Multi-Objective Machine Learning, 2006
ISBN 3-540-30676-5

Vol. 17. Te-Ming Huang, Vojislav Kecman, Ivica Kopriva
Kernel Based Algorithms for Mining Huge Data Sets, 2006
ISBN 3-540-31681-7

Vol. 18. Chang Wook Ahn
Advances in Evolutionary Algorithms, 2006
ISBN 3-540-31758-9

Vol. 19. Ajita Ichalkaranje, Nikhil Ichalkaranje, Lakhmi C. Jain (Eds.)
Intelligent Paradigms for Assistive and Preventive Healthcare, 2006
ISBN 3-540-31762-7

Vol. 20. Wojciech Pecznek, Agata Pó³rola
Advances in Verification of Time Petri Nets and Timed Automata, 2006
ISBN 3-540-32869-6

Vol. 21 Cândida Ferreira
Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence, 2006
ISBN 3-540-32796-7

Cândida Ferreira

Gene Expression Programming

Mathematical Modeling by an Artificial Intelligence

Second, revised and extended edition

Springer

Dr. Cândida Ferreira
Chief Scientist
Gepsoft Ltd.
73 Elmtree Drive
Bristol BS13 8NA
United Kingdom
E-mail : candidaf@gepsoft.com

Library of Congress Control Number: 2006921791

ISSN print edition: 1860-949X
ISSN electronic edition: 1860-9503
ISBN-10 3-540-32796-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-327967 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springer.com

Printed in The Netherlands

Cover design: deblik, Berlin
Typesetting: by the author and TechBooks
Printed on acid-free paper SPIN: 11506591 89/Strasser 5 4 3 2 1 0

To José Simas
For All the Dreams

and

To my Grandfather, Domingos de Carvalho
For His Vision

Preface to the Second Edition

The idea for this second edition came from Janusz Kacprzyk on April 29, 2005, who kindly invited me to his new Springer series, Studies in Computational Intelligence. The initial plan was to correct the usual typos and mistakes but leave the book unchanged, as Janusz thought (and I agreed with him) that it was the proper moment for a second edition. But then there was the problem of the new format and I had to reformat and proofread everything again. And I just thought that while I was at it, I might as well change some things in the book to make it more enjoyable and interesting. Foremost in my thoughts was the restructuring of chapter 4, The Basic GEA in Problem Solving. In that chapter, buried together with a wide variety of problems, were several important new algorithms that I wanted to bring to the forefront. These algorithms include: the GEP-RNC algorithm (the cornerstone of several new other algorithms); automatically defined functions; polynomial induction; and parameter optimization. So I removed all these materials from chapter 4 and gave them the deserved attention by writing four new chapters (chapter 5 Numerical Constants and the GEP-RNC Algorithm, chapter 6 Automatically Defined Functions in Problem Solving, chapter 7 Polynomial Induction and Time Series Prediction, and chapter 8 Parameter Optimization). Then this new structure just begged for me to include one of the last additions to the GEP technique – decision trees – that I was regretfully unable to include in the first edition (I implemented decision trees in August of 2002, two months before sending the manuscript to the printer). So, chapter 9, Decision Tree Induction, is totally new to this second edition and an interesting addition both to the book and to GEP. The last three chapters, chapter 10 Design of Neural Networks, chapter 11 Combinatorial Optimization, and chapter 12 Evolutionary Studies, remain basically unchanged.

With all this restructuring in chapter 4, I was able to develop several new topics, including solving problems with multiple outputs in one go and designing parsimonious solutions both with parsimony pressure and user defined functions, also interesting new extensions to the GEP technique. Furthermore, the section on Logic Synthesis was totally restructured and an interesting analysis of the most common universal logical systems is presented.

Chapter 3, The Basic Gene Expression Algorithm, with the exception of section 2, Fitness Functions and the Selection Environment, remains practically unchanged. In section 2, however, I introduce several new fitness functions that are then used to explore more efficiently the solution landscapes of the problems solved in the book.

The inversion operator is one of the latest additions to the GEP technique, and you will notice an extra entry for it in chapters 3 and 10. Furthermore, you'll also notice that both IS and RIS transposition were slightly modified so that the transposon sizes were automatically chosen rather than a priori set. But unbeknownst to me, this slightly different implementation had consequences in the performance of these operators, and the new implementation is slightly more unpredictable in terms of performance. This is the reason why in the Evolutionary Studies of chapter 12, the old implementation of these operators is still used. And to be fair, the comparison of the inversion operator with these transposition operators should also use a fixed set of sizes for the inverted sequences. And this is the reason why inversion is not analyzed in chapter 12.

Cândida Ferreira
December 20, 2005

Preface to the First Edition

I developed the basic ideas of gene expression programming (GEP) in September and October of 1999 almost unaware of their uniqueness. I was reading Mitchell's book *An Introduction to Genetic Algorithms* (Mitchell 1996) and meticulously solving all the computer exercises provided at the end of each chapter. Therefore, I implemented my first genetic algorithm and I also implemented what I thought was a genetic programming (GP) system. Like a GP system, this new system could also evolve computer programs of different sizes and shapes but, surprisingly, it surpassed the old GP system by a factor of 100-60,000. So, what happened here? What was responsible for this astounding difference in performance? For an evolutionary biologist, the answer is quite straightforward: this new system – gene expression programming – simply crossed the phenotype threshold. This means that the complex computer programs (the phenotype) evolved by GEP are totally encoded in simple strings of fixed length (the chromosomes or genotype). The separation of the genotype from the phenotype is comparable to opening a Pandora box full of good things or possibilities. Of these good things, perhaps the most important is that there are virtually no restrictions concerning the number or type of genetic operators used. Another important thing is that the creation of higher levels of complexity becomes practically a trivial task. Indeed, it was trivial to create a multigenic system from a unigenic one and a multicellular system from a unicellular one. And each new system creates its own box of new possibilities, which enlarges considerably the scope of this new technique.

In this first book on gene expression programming I describe thoroughly the basic gene expression algorithm and numerous modifications to this new algorithm, providing all the implementation details so that anyone with elementary programming skills (or willing to learn them) will be able to implement it themselves. The first chapter briefly introduces the main players of biological gene expression in order to show how they relate to the main players of artificial evolutionary systems in general and GEP in particular. The second chapter introduces the players of gene expression programming, showing their structural and functional organization in detail. The language especially created to express the genetic information of GEP chromosomes is also described in this chapter. Chapter 3 gives a detailed description of the basic gene expression algorithm and the basic genetic operators. In addition, a very simple problem is exhaustively dissected, showing all the individual programs created during the discovery process in order to demystify the workings of adaptation and evolution. Chapter 4 describes some of the applications of the basic gene expression algorithm, including a large body of unpublished materials, namely, parameter optimization, evolution of Kolmogorov-Gabor polynomials, time series prediction, classifier systems, evolution of linking functions, multicellularity, automatically defined functions, user defined functions and so forth. The materials of Chapter 5 are also new and show how to simulate complete neural networks with gene expression programming. Two benchmark problems are solved with these GEP-nets, providing an effective measure of their performance. Chapter 6 shows how to do combinatorial optimization with gene expression programming. Multigene families and several combinatorial-specific operators are introduced and their performance evaluated on two scheduling problems. The last chapter discusses some important and controversial evolutionary topics that might be refreshing to both evolutionary computists and evolutionary biologists.

Acknowledgments

The invention of a new paradigm can often create strong resistance, especially if it seems to endanger long established technologies and enterprises. The publication of my work on scientific journals and conferences, which should be forums for discussing and sharing new ideas, became a nightmare and both my work and myself were outright dismissed and treated with scorn. Despite the initial opposition and due to a set of happy circumstances and resources, I was finally able to make my work known and available to all. I am deeply indebted to José Simas, an accomplished graphic and web designer and software developer, for believing in me and in GEP from the beginning and for helping its expansion and promotion on the World Wide

Web. Together we founded Gepsoft and developed software based on gene expression programming which is already helping numerous scientists and engineers worldwide. And thanks to Gepsoft it was possible for me to concentrate fully on the writing of this book and on the development of several new algorithms. Indeed, my work at Gepsoft benefited tremendously from my writing and vice versa.

I am also very grateful to Pedro Carneiro, a talented musician with an avid mind, for reading and editing the first three chapters of the manuscript. José Simas also read several drafts of the manuscript, accompanying the process from the beginning and contributing with valuable discussions and suggestions. He is, in fact, my first reader and I always write with him in my mind.

Finally, I would like to thank José Gabriel, a talented printer and skilled craftsman, for his involvement in the making of this book from the start and for handling its printing with special care.

Cândida Ferreira
October 15, 2002

List of Symbols

ADF	Automatically defined function
ADF-RNC	ADFs with random numerical constants
AND	AND or Boolean function of two arguments #8
APS	Gepsoft Automatic Problem Solver
CA	Cellular automata
Dc	Gene domain for encoding random numerical constants
DT	Decision tree
Dt	Gene domain for encoding the thresholds of neural networks
Dw	Gene domain for encoding the weights of neural networks
EDT	Evolvable decision trees
EDT-RNC	Evolvable decision trees with numeric attributes
ET	Expression tree
FN	False negatives
FP	False positives
GA	Genetic algorithm
GEA	Gene expression algorithm
GEP	Gene expression programming
GEP-ADF	GEP with automatically defined functions
GEP-EDT	GEP for inducing evolvable decision trees
GEP-KGP	GEP for inducing Kolmogorov-Gabor polynomials
GEP-MO	GEP for solving problems with multiple outputs
GEP-NC	GEP with numerical constants
GEP-nets	GEP for inducing neural networks
GEP-NN	GEP for inducing neural networks
GEP-PO	GEP for parameter optimization
GEP-RNC	GEP with random numerical constants
GKL	Gacs-Kurdyumov-Levin rule
GOE	Greater Or Equal or Boolean function of two arguments #13

GP	Genetic programming
GT	Greater Than or Boolean function of two arguments #4
HZero	Parameter optimization algorithm with a head size of zero
IC	Initial configuration
IF	Rule of three arguments #202, If $a = 1$, then b, else c
IS	Insertion sequence elements
LOE	Less Or Equal or Boolean function of two arguments #11
LT	Less Than or Boolean function of two arguments #2
MAJ	Majority function of three arguments or rule #232
MGF	Multigene family
MIN	Minority function of three arguments or rule #23
MUX	3-Multiplexer or rule #172, If $a = 0$, then b, else c
NAND	NAND or Boolean function of two arguments #7
NC	Numerical constants
NLM	NAND-like module
NOR	NOR or Boolean function of two arguments #1
NPV	Negative predictive value
NXOR	NXOR or Boolean function of two arguments #9
OR	OR or Boolean function of two arguments #14
ORF	Open reading frame
PPV	Positive predictive value
RIS	Root insertion sequence elements
RNC	Random numerical constants
TAP	Task assignment problem
TN	True negatives
TP	True positives
TSP	Traveling salesperson problem
UDF	User defined function
ULM	Universal logical module
XOR	Exclusive-OR or Boolean function of two arguments #6

Contents