Preface

Designing complex programs or systems such as operating systems, compilers, filing systems, data base systems, etc. is an old ever lasting research area. Genetic programming is a relatively new promising and growing research area. Among other uses, it provides efficient tools to deal with hard problems by evolving creative and competitive solutions. Systems Programming is generally strewn with such hard problems. This book is devoted to reporting innovative and significant progress about the contribution of genetic programming in systems programming. The contributions of this book clearly demonstrate that genetic programming is very effective in solving hard and yet-open problems in systems programming. Followed by an introductory chapter, in the remaining contributed chapters, the reader can easily learn about systems where genetic programming can be applied successfully. These include but are not limited to, information security systems (see Chapter 3), compilers (see Chapter 4), data mining systems (see Chapter 5), stock market prediction systems (see Chapter 6), robots (see Chapter 8) and automatic programming (see Chapters 7 and 9).

In Chapter 1, which is entitled "*Evolutionary Computation: from Genetic Algorithms to Genetic Programming*", the authors introduce and review the development of the field of evolutionary computations from standard genetic algorithms to genetic programming, passing by evolution strategies and evolutionary programming. The main differences among the different evolutionary computation techniques are also illustrated in this Chapter.

In Chapter 2, which is entitled "*Automatically Defined Functions in Gene Expression Programming*", the author introduces the cellular system of Gene Expression Programming with Automatically Defined Functions (ADF) and discusses the importance of ADFs in Automatic Programming by comparing the performance of sophisticated learning systems with ADFs with much simpler ones without ADFs on a benchmark problem of symbolic regression.

In Chapter 3, which is entitled "*Evolving Intrusion Detection Systems*", the authors present an Intrusion Detection System (IDS), which is a program that analyzes what happens or has happened during an execution and tries to find indications that the computer has been misused. An IDS does not eliminate the use of preventive mechanism but it works as the last defensive mechanism in securing the system. The authors also evaluate the performances of two Genetic Programming techniques for IDS namely Linear Genetic Programming (LGP) and Multi-Expression Programming (MEP). They compare the obtained results with some machine learning techniques like Support Vector Machines (SVM) and Decision Trees (DT). The authors claim that empirical results clearly show that GP techniques could play an important role in designing real time intrusion detection systems.

In Chapter 4, which is entitled "*Evolutionary Pattern Matching Using Genetic Programming*", the authors apply GP to the hard problem of engineering pattern matching automata for non-sequential pattern set, which is almost always the case in functional programming. They engineer good traversal orders that allow one to design an efficient adaptive pattern-matchers that visit necessary positions only. The authors claim that doing so the evolved pattern matching automata improves time and space requirements of pattern-matching as well as the termination properties of term evaluation.

In Chapter 5, which is entitled "*Genetic Programming in Data Modelling*", the author demonstrates some abilities of Genetic Programming (GP) in Data Modelling (DM). The author shows that GP can make data collected in large databases more useful and understandable. The author concentrates on mathematical modelling, classification, prediction and modelling of time series.

In Chapter 6, which is entitled "*Stock Market Modeling Using Genetic Programming Ensembles*", the authors introduce and use two Genetic Programming (GP) techniques: Multi-Expression Programming (MEP) and Linear Genetic Programming (LGP) for the prediction of two stock indices. They compare the performance of the GP techniques with an artificial neural network trained using Levenberg-Marquardt algorithm and Takagi-Sugeno neuro-fuzzy model. As a case study, the authors consider Nasdaq-100 index of Nasdaq Stock Market and the S\&P CNX NIFTY stock index as test data. Based on the empirical results obtained the authors conclude that Genetic Programming techniques are promising methods for stock prediction. Finally, they formulate an ensemble of these two techniques using a multiobjective evolutionary algorithm and claim that results reached by ensemble of GP techniques are better than the results obtained by each GP technique individually.

In Chapter 7, which is entitled "*Evolutionary Digital Circuit Design Using Genetic Programming*", the authors study two different circuit encodings used for digital circuit evolution. The first approach is based on genetic programming, wherein digital circuits consist of their data flow based specifications. In this approach, individuals are internally represented by the abstract trees/DAG of the corresponding circuit specifications. In the second approach, digital circuits are thought of as a map of rooted gates. So individuals are represented by two-dimensional arrays of cells. The authors compare the impact of both individual representations on the evolution process of digital circuits. The authors reach the conclusion that employing either of these approaches yields circuits of almost the same characteristics in terms of space and response time. However, the evolutionary process is much shorter with the second linear encoding.

In Chapter 8, which is entitled "*Evolving Complex Robotic Behaviors Using Genetic Programming*", the author reviews different methods for evolving complex robotic

behaviors. The methods surveyed use two different approaches: The first one introduces hierarchy into GP by using library of procedures or new primitive functions and the second one uses GP to evolve the building modules of robot controller hierarchy. The author comments on including practical issues of evolution as well as comparison between the two approaches.

In Chapter 9, which is entitled "*Automatic Synthesis of Microcontroller Assembly Code Through Linear Genetic Programming*", the authors distinguish the potential of linear genetic programming in the automatic synthesis of microcontroller assembly language programs. For them, these programs implement strategies for time-optimal or sub-optimal control of the system to be controlled, based on mathematical modeling through dynamic equations. They also believe that within this application class, the best model is the one used in linear genetic programming, in which each chromosome is represented by an instruction list. The authors find the synthesis of programs that implement optimal-time control strategies for microcontrollers, directly in assembly language, as an attractive alternative that overcomes the difficulties presented by the conventional design of optimal control systems. This chapter widens the perspective of broad usage of genetic programming in automatic control.

We are very much grateful to the authors of this volume and to the reviewers for their tremendous service by critically reviewing the chapters. The editors would like to thank Dr. Thomas Ditzinger Springer Verlag, Germany for the editorial assistance and excellent cooperative collaboration to produce this important scientific work. We hope that the reader will share our excitement to present this volume on '***Genetic Systems Programming***' and will find it useful.

Nadia Nedjah
State University of Rio de Janeiro, Brazil

Ajith Abraham
IITA Professorship Program, Chung-Ang University, Korea

Luiza de Macedo Mourelle
State University of Rio de Janeiro, Brazil

(Editors)