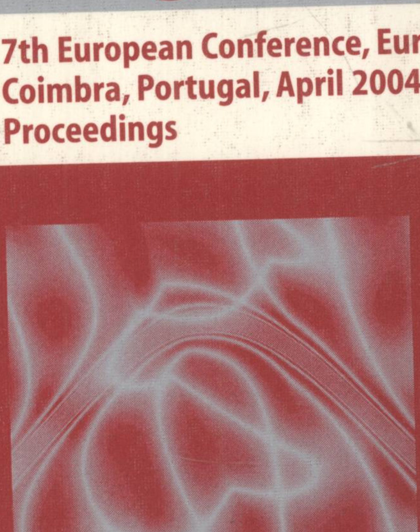
Maarten Keijzer
Una-May O'Reilly
Simon M. Lucas
Ernesto Costa
Terence Soule (Eds.)

LNCS 3003

Genetic Programming

7th European Conference, EuroGP 2004
Coimbra, Portugal, April 2004
Proceedings



Springer

Maarten Keijzer Una-May O'Reilly
Simon M. Lucas Ernesto Costa
Terence Soule (Eds.)

Genetic Programming

7th European Conference, EuroGP 2004
Coimbra, Portugal, April 5-7, 2004
Proceedings

Springer

Volume Editors

Maarten Keijzer
KiQ Ltd
De Lairessestraat 150, 1075 HL, Amsterdam, The Netherlands
E-mail: mkeijzer@xs4all.nl

Una-May O'Reilly
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology, Cambridge, MA, 02139, USA
E-mail: unamay@ai.mit.edu

Simon M. Lucas
University of Essex, Dept. of Computer Science
Colchester CO4 3SQ,UK
E-mail: sml@essex.ac.uk

Ernesto Costa
University of Coimbra, Department of Computer Science
Polo II - Pinhal Marrocos, 3030-290 Coimbra, Portugal
E-mail: ernesto@dei.uc.pt

Terence Soule
University of Idaho, Department of Computer Science, Moscow, Id 83844-1010, USA
E-mail: tsoule@cs.uidaho.edu

Library of Congress Control Number: 2004102630

Coverillustration: "Embrace" by Anargyros Sarafopoulos

CR Subject Classification (1998): D.1, F.1, F.2, I.5, I.2, J.3

ISSN 0302-9743
ISBN 3-540-21346-5 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 10992999 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3003

Preface

In this volume we present the accepted contributions for the 7th European Conference on Genetic Programming (EuroGP 2004). The conference took place on 5–7 April 2004 in Portugal at the University of Coimbra, in the Department of Mathematics in Praça Dom Dinis, located on the hill above the old town.

EuroGP is a well-established conference and the sole one exclusively devoted to Genetic Programming. Previous proceedings have all been published by Springer-Verlag in the LNCS series. EuroGP began as an international workshop in Paris, France in 1998 (14–15 April, LNCS 1391). Subsequently the workshop was held in Göteborg, Sweden in 1999 (26–27 May, LNCS 1598) and then EuroGP became an annual conference: in 2000 in Edinburgh, UK (15–16 April, LNCS 1802), in 2001 at Lake Como, Italy (18–19 April, LNCS 2038), in 2002 in Kinsale, Ireland (3–5 April, LNCS 2278), and in 2003 in Colchester, UK (14–16 April, LNCS 2610). From the outset, there have always been specialized workshops, co-located with EuroGP, focusing on applications of evolutionary algorithms (LNCS 1468, 1596, 1803, 2037, 2279, and 2611). This year the EvoCOP workshop on combinatorial optimization transformed itself into a conference in its own right, and the two conferences, together with the EvoWorkshops, EvoBIO, EvoIASP, EvoMUSART, EvoSTOC, EvoHOT, and EvoCOMNET, now form one of the largest events dedicated to Evolutionary Computation in Europe.

Genetic Programming (GP) is evolutionary computation that solves specific complex problems or tasks by evolving and adapting a population of computer programs, using Darwinian evolution and Mendelian genetics as its sources of inspiration. Some of the 38 papers included in these proceedings address foundational and theoretical issues, and there is also a wide variety of papers dealing with different application areas, such as computer science, engineering, language understanding, biology and design, demonstrating that GP is a powerful and practical problem-solving paradigm.

A total of 61 papers were received. A rigorous, double-blind, peer-review selection mechanism was applied to 58 of them. This resulted in 19 plenary talks (31% of those submitted) and 19 research posters. Every paper was reviewed by at least two of the 46 members of the program committee who were carefully selected internationally for their knowledge and competence. As far as possible, papers were matched with the reviewer's particular interests and special expertise. The result of this careful process can be seen here in the high quality of the contributions published within this volume.

Of the 38 accepted papers, 32 have authors who came from European countries (about 85%), confirming the strong European character of the conference. The other 6 came from the USA, Korea, China, New Zealand, and Australia, emphasizing the global nature of our field.

We would like to express our sincere thanks especially to the two internationally renowned speakers who gave keynote talks at the joint conference and workshops plenary sessions: Prof. Stephanie Forrest of the University of New Mexico, and Prof. Zbigniew Michalewicz of the University of North Carolina.

The success of any conference results from the efforts of many people, to whom we would like to express our gratitude. First, we would like to thank the members of the program committee for their attentiveness, perseverance, and willingness to provide high-quality reviews. We would especially like to thank Jennifer Willies who ensured the conference's continued existence and has been greatly influential in sustaining the high quality of the conference organization. Without Jennifer, we would have been lost. Last but not least, we thank the University of Coimbra for hosting the conference.

April 2004

Maarten Keijzer
Una-May O'Reilly
Simon Lucas
Ernesto Costa
Terence Soule

Organization

EuroGP 2004 was organized by EvoGP, the EvoNet Working Group on Genetic Programming.

Organizing Committee

Conference Co-chairs	Maarten Keijzer (Free University, The Netherlands) Una-May O'Reilly (MIT, USA)
Publicity Chair	Simon Lucas (University of Essex, UK)
Local Chair	Ernesto Costa (University of Coimbra, Portugal)
Publication Chair	Terence Soule (University of Idaho, USA)

Program Committee

Vladan Babovic, Tectrasys AG
Wolfgang Banzhaf, Memorial University of Newfoundland
Bertrand Braunschweig, Institut Français du Pétrole
Martin C. Martin, MIT
Stefano Cagnoni, University of Parma
Jean-Jacques Chabrier, University of Burgundy
Pierre Colet, Laboratoire d'Informatique du Littoral
Ernesto Costa, University of Coimbra
Marco Dorigo, Université Libre de Bruxelles
Malachy Eaton, University of Limerick
Marc Ebner, Universitaet Wuerzburg
Jeroen Eggermont, Leiden University
Aniko Ekart, Hungarian Academy of Sciences
Daryl Essam, University of New South Wales
Francisco Fernandez de Vega, University of Extremadura
Cyril Fonlupt, Université du Littoral
Alex Freitas, University of Kent
Wolfgang Golubski, University of Siegen
Steven Gustafson, University of Nottingham
Jin-Kao Hao, Université d'Angers
Daniel Howard, QinetiQ
Christian Jacob, University of Calgary
Colin Johnson, University of Kent at Canterbury
Didier Keymeulen, Jet Propulsion Laboratory, CA, USA
Bill Langdon, University College London
Simon Lucas, University of Essex

Lecture Notes in Computer Science

For information about Vols. 1–2881

please contact your bookseller or Springer-Verlag

Vol. 3005: G.R. Raidl, S. Cagnoni, J. Branke, D.W. Corne, R. Drechsler, Y. Jin, C.G. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G.D. Smith, G. Squillero (Eds.), Applications of Evolutionary Computing. XVII, 562 pages. 2004.

Vol. 3004: J. Gottlieb, G.R. Raidl (Eds.), Evolutionary Computation in Combinatorial Optimization. X, 241 pages. 2004.

Vol. 3003: M. Keijzer, U.-M. O'Reilly, S.M. Lucas, E. Costa, T. Soule (Eds.), Genetic Programming. XI, 410 pages. 2004.

Vol. 2997: S. McDonald, J. Tait (Eds.), Advances in Information Retrieval. XIII, 427 pages. 2004.

Vol. 2996: V. Diekert, M. Habib (Eds.), STACS 2004. XVI, 658 pages. 2004.

Vol. 2995: C. Jensen, S. Poslad, T. Dimitrakos (Eds.), Trust Management. XIII, 377 pages. 2004.

Vol. 2994: E. Rahm (Ed.), Data Integration in the Life Sciences. X, 221 pages. 2004. (Subseries LNBI).

Vol. 2993: R. Alur, G.J. Pappas (Eds.), Hybrid Systems: Computation and Control. XII, 674 pages. 2004.

Vol. 2992: E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Böhm, E. Ferrari (Eds.), Advances in Database Technology - EDBT 2004. XVIII, 877 pages. 2004.

Vol. 2991: R. Alt, A. Frommer, R.B. Kearfott, W. Luther (Eds.), Numerical Software with Result Verification. X, 315 pages. 2004.

Vol. 2989: S. Graf, L. Mounier (Eds.), Model Checking Software. X, 309 pages. 2004.

Vol. 2988: K. Jensen, A. Podelski (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. XIV, 608 pages. 2004.

Vol. 2987: I. Walukiewicz (Ed.), Foundations of Software Science and Computation Structures. XIII, 529 pages. 2004.

Vol. 2986: D. Schmidt (Ed.), Programming Languages and Systems. XII, 417 pages. 2004.

Vol. 2985: E. Duesterwald (Ed.), Compiler Construction. X, 313 pages. 2004.

Vol. 2984: M. Wermelinger, T. Margaria-Steffen (Eds.), Fundamental Approaches to Software Engineering. XII, 389 pages. 2004.

Vol. 2983: S. Istrail, M.S. Waterman, A. Clark (Eds.), Computational Methods for SNPs and Haplotype Inference. IX, 153 pages. 2004. (Subseries LNBI).

Vol. 2982: N. Wakamiya, M. Solarski, J. Sterbenz (Eds.), Active Networks. XI, 308 pages. 2004.

Vol. 2981: C. Müller-Schloer, T. Ungerer, B. Bauer (Eds.), Organic and Pervasive Computing – ARCS 2004. XI, 339 pages. 2004.

Vol. 2980: A. Blackwell, K. Marriott, A. Shimojima (Eds.), Diagrammatic Representation and Inference. XV, 448 pages. 2004. (Subseries LNAI).

Vol. 2978: R. Groz, R.M. Hierons (Eds.), Testing of Communicating Systems. XII, 225 pages. 2004.

Vol. 2977: G. Di Marzo Serugendo, A. Karageorgos, O.F. Rana, F. Zambonelli (Eds.), Engineering Self-Organising Systems. X, 299 pages. 2004. (Subseries LNAI).

Vol. 2976: M. Farach-Colton (Ed.), LATIN 2004: Theoretical Informatics. XV, 626 pages. 2004.

Vol. 2973: Y. Lee, J. Li, K.-Y. Whang, D. Lee (Eds.), Database Systems for Advanced Applications. XXIV, 925 pages. 2004.

Vol. 2970: F. Fernández Rivera, M. Bubak, A. Gómez Tato, R. Doallo (Eds.), Grid Computing. XI, 328 pages. 2004.

Vol. 2964: T. Okamoto (Ed.), Topics in Cryptology – CT-RSA 2004. XI, 387 pages. 2004.

Vol. 2963: R. Sharp, Higher Level Hardware Synthesis. XVI, 195 pages. 2004.

Vol. 2962: S. Bistarelli, Semirings for Soft Constraint Solving and Programming. XII, 279 pages. 2004.

Vol. 2961: P. Eklund (Ed.), Concept Lattices. IX, 411 pages. 2004. (Subseries LNAI).

Vol. 2960: P.D. Mosses (Ed.), CASL Reference Manual. XVII, 528 pages. 2004.

Vol. 2958: L. Rauchwerger (Ed.), Languages and Compilers for Parallel Computing. XI, 556 pages. 2004.

Vol. 2957: P. Langendoerfer, M. Liu, I. Matta, V. Tsaoussidis (Eds.), Wired/Wireless Internet Communications. XI, 307 pages. 2004.

Vol. 2954: F. Crestani, M. Dunlop, S. Mizzaro (Eds.), Mobile and Ubiquitous Information Access. X, 299 pages. 2004.

Vol. 2953: K. Konrad, Model Generation for Natural Language Interpretation and Analysis. XIII, 166 pages. 2004. (Subseries LNAI).

Vol. 2952: N. Guelfi, E. Astesiano, G. Reggio (Eds.), Scientific Engineering of Distributed Java Applications. X, 157 pages. 2004.

Vol. 2951: M. Naor (Ed.), Theory of Cryptography. XI, 523 pages. 2004.

Vol. 2949: R. De Nicola, G. Ferrari, G. Meredith (Eds.), Coordination Models and Languages. X, 323 pages. 2004.

Vol. 2948: G.L. Mullen, A. Poli, H. Stichtenoth (Eds.), Finite Fields and Applications. VIII, 263 pages. 2004.

Vol. 2947: F. Bao, R. Deng, J. Zhou (Eds.), Public Key Cryptography – PKC 2004. XI, 455 pages. 2004.

Vol. 2946: R. Focardi, R. Gorrieri (Eds.), Foundations of Security Analysis and Design II. VII, 267 pages. 2004.

Vol. 2943: J. Chen, J. Reif (Eds.), DNA Computing. X, 225 pages. 2004.

Vol. 2941: M. Wirsing, A. Knapp, S. Balsamo (Eds.), Radical Innovations of Software and Systems Engineering in the Future. X, 359 pages. 2004.

Vol. 2940: C. Lucena, A. Garcia, A. Romanovsky, J. Castro, P.S. Alencar (Eds.), Software Engineering for Multi-Agent Systems II. XII, 279 pages. 2004.

Vol. 2939: T. Kalker, I.J. Cox, Y.M. Ro (Eds.), Digital Watermarking. XII, 602 pages. 2004.

Vol. 2937: B. Steffen, G. Levi (Eds.), Verification, Model Checking, and Abstract Interpretation. XI, 325 pages. 2004.

Vol. 2934: G. Lindemann, D. Moldt, M. Paolucci (Eds.), Regulated Agent-Based Social Systems. X, 301 pages. 2004. (Subseries LNAI).

Vol. 2930: F. Winkler (Ed.), Automated Deduction in Geometry. VII, 231 pages. 2004. (Subseries LNAI).

Vol. 2926: L. van Elst, V. Dignum, A. Abecker (Eds.), Agent-Mediated Knowledge Management. XI, 428 pages. 2004. (Subseries LNAI).

Vol. 2923: V. Lifschitz, I. Niemelä (Eds.), Logic Programming and Nonmonotonic Reasoning. IX, 365 pages. 2004. (Subseries LNAI).

Vol. 2919: E. Giunchiglia, A. Tacchella (Eds.), Theory and Applications of Satisfiability Testing. XI, 530 pages. 2004.

Vol. 2917: E. Quintarelli, Model-Checking Based Data Retrieval. XVI, 134 pages. 2004.

Vol. 2916: C. Palamidessi (Ed.), Logic Programming. XII, 520 pages. 2003.

Vol. 2915: A. Camurri, G. Volpe (Eds.), Gesture-Based Communication in Human-Computer Interaction. XIII, 558 pages. 2004. (Subseries LNAI).

Vol. 2914: P.K. Pandya, J. Radhakrishnan (Eds.), FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science. XIII, 446 pages. 2003.

Vol. 2913: T.M. Pinkston, V.K. Prasanna (Eds.), High Performance Computing - HiPC 2003. XX, 512 pages. 2003. (Subseries LNAI).

Vol. 2911: T.M.T. Sembok, H.B. Zaman, H. Chen, S.R. Urs, S.H. Myaeng (Eds.), Digital Libraries: Technology and Management of Indigenous Knowledge for Global Access. XX, 703 pages. 2003.

Vol. 2910: M.E. Orlowska, S. Weerawarana, M.M.P. Papazoglou, J. Yang (Eds.), Service-Oriented Computing - ICSOC 2003. XIV, 576 pages. 2003.

Vol. 2909: R. Solis-Oba, K. Jansen (Eds.), Approximation and Online Algorithms. VIII, 269 pages. 2004.

Vol. 2908: K. Chae, M. Yung (Eds.), Information Security Applications. XII, 506 pages. 2004.

Vol. 2907: I. Lirkov, S. Margenov, J. Wasniewski, P. Yalamov (Eds.), Large-Scale Scientific Computing. XI, 490 pages. 2004.

Vol. 2906: T. Ibaraki, N. Katoh, H. Ono (Eds.), Algorithms and Computation. XVII, 748 pages. 2003.

Vol. 2905: A. Sanfeliu, J. Ruiz-Shulcloper (Eds.), Progress in Pattern Recognition, Speech and Image Analysis. XVII, 693 pages. 2003.

Vol. 2904: T. Johansson, S. Maitra (Eds.), Progress in Cryptology - INDOCRYPT 2003. XI, 431 pages. 2003.

Vol. 2903: T.D. Gedeon, L.C.C. Fung (Eds.), AI 2003: Advances in Artificial Intelligence. XVI, 1075 pages. 2003. (Subseries LNAI).

Vol. 2902: F.M. Pires, S.P. Abreu (Eds.), Progress in Artificial Intelligence. XV, 504 pages. 2003. (Subseries LNAI).

Vol. 2901: F. Bry, N. Henze, J. Ma luszyński (Eds.), Principles and Practice of Semantic Web Reasoning. X, 209 pages. 2003.

Vol. 2900: M. Bidoit, P.D. Mosses (Eds.), Casl User Manual. XIII, 240 pages. 2004.

Vol. 2899: G. Ventre, R. Canonico (Eds.), Interactive Multimedia on Next Generation Networks. XIV, 420 pages. 2003.

Vol. 2898: K.G. Paterson (Ed.), Cryptography and Coding. IX, 385 pages. 2003.

Vol. 2897: O. Balet, G. Subsol, P. Torguet (Eds.), Virtual Storytelling. XI, 240 pages. 2003.

Vol. 2896: V.A. Saraswat (Ed.), Advances in Computing Science – ASIAN 2003. VIII, 305 pages. 2003.

Vol. 2895: A. Ohori (Ed.), Programming Languages and Systems. XIII, 427 pages. 2003.

Vol. 2894: C.S. Laih (Ed.), Advances in Cryptology - ASIACRYPT 2003. XIII, 543 pages. 2003.

Vol. 2893: J.-B. Stefani, I. Demeure, D. Hagimont (Eds.), Distributed Applications and Interoperable Systems. XIII, 311 pages. 2003.

Vol. 2892: F. Dau, The Logic System of Concept Graphs with Negation. XI, 213 pages. 2003. (Subseries LNAI).

Vol. 2891: J. Lee, M. Barley (Eds.), Intelligent Agents and Multi-Agent Systems. X, 215 pages. 2003. (Subseries LNAI).

Vol. 2890: M. Broy, A.V. Zamulin (Eds.), Perspectives of System Informatics. XV, 572 pages. 2003.

Vol. 2889: R. Meersman, Z. Tari (Eds.), On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops. XIX, 1071 pages. 2003.

Vol. 2888: R. Meersman, Z. Tari, D.C. Schmidt (Eds.), On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. XXI, 1546 pages. 2003.

Vol. 2887: T. Johansson (Ed.), Fast Software Encryption. IX, 397 pages. 2003.

Vol. 2886: I. Nyström, G. Sanniti di Baja, S. Svensson (Eds.), Discrete Geometry for Computer Imagery. XII, 556 pages. 2003.

Vol. 2885: J.S. Dong, J. Woodcock (Eds.), Formal Methods and Software Engineering. XI, 683 pages. 2003.

Vol. 2884: E. Najm, U. Nestmann, P. Stevens (Eds.), Formal Methods for Open Object-Based Distributed Systems. X, 293 pages. 2003.

Vol. 2883: J. Schaeffer, M. Müller, Y. Björnsson (Eds.), Computers and Games. XI, 431 pages. 2003.

Vol. 2882: D. Veit, Matchmaking in Electronic Markets. XV, 180 pages. 2003. (Subseries LNAI).

Table of Contents

Papers

Evaluation of Chess Position by Modular Neural Network Generated by Genetic Algorithm 1
Mathieu Autonès, Ariel Beck, Philippe Camacho, Nicolas Lassabe, Hervé Luga, François Scharffe

Coevolution of Algorithms and Deterministic Solution of Equations in Free Groups 11
Richard F. Booth, Alexandre V. Borovik

Designing Optimal Combinational Digital Circuits Using a Multiple Logic Unit Processor 23
Sin Man Cheang, Kin Hong Lee, Kwong Sak Leung

A Data Structure for Improved GP Analysis via Efficient Computation and Visualisation of Population Measures 35
Anikó Ekárt, Steven Gustafson

Boosting Technique for Combining Cellular GP Classifiers 47
Gianluigi Folino, Clara Pizzuti, Giandomenico Spezzano

Co-evolving Faults to Improve the Fault Tolerance of Sorting Networks 57
Michael L. Harrison, James A. Foster

Toward an Alternative Comparison between Different Genetic Programming Systems 67
Nguyen Xuan Hoai, R.I. (Bob) McKay, D. Essam, H.A. Abbass

Lymphoma Cancer Classification Using Genetic Programming with SNR Features 78
Jin-Hyuk Hong, Sung-Bae Cho

A Practical Approach to Evolving Concurrent Programs 89
David Jackson

Evolutionary Induction of Grammar Systems for Multi-agent Cooperation 101
Clayton M. Johnson, James Farrell

Genetic Programming Applied to Mixed Integer Programming 113
Konstantinos Kostikas, Charalambos Fragakis

Efficient Crossover in the GAuGE System 125
Miguel Nicolau, Conor Ryan

Grammatical Evolution by Grammatical Evolution:
The Evolution of Grammar and Genetic Code 138
Michael O'Neill, Conor Ryan

Constrained Molecular Dynamics as a Search and Optimization Tool 150
Riccardo Poli, Christopher R. Stephens

On the Performance of Genetic Operators and the Random
Key Representation 162
Eoin Ryan, R. Muhammad Atif Azad, Conor Ryan

Analysis of GP Improvement Techniques over the Real-World
Inverse Problem of Ocean Color 174
Grégory Valigiani, Cyril Fonlupt, Pierre Collet

Evolution and Acquisition of Modules in Cartesian
Genetic Programming 187
James Alfred Walker, Julian Francis Miller

How to Choose Appropriate Function Sets for Gentic Programming 198
Gang Wang, Terence Soule

Improving Grammar-Based Evolutionary Algorithms
via Attributed Derivation Trees 208
Szilvia Zvada, Róbert Ványi

Posters

Evolved Matrix Operations for Post-processing Protein Secondary
Structure Predictions 220
Varun Aggarwal, Robert M. MacCallum

Genetic Programming for Natural Language Parsing 230
Lourdes Araujo

Comparing Hybrid Systems to Design and Optimize Artificial
Neural Networks 240
P.A. Castillo, M.G. Arenas, J.J. Merelo, G. Romero, F. Rateb, A. Prieto

An Evolutionary Algorithm for the Input-Output Block
Assignment Problem 250
Kit Yan Chan, Terence C. Fogarty

Genetic Programming for Subjective Fitness Function Identification 259
Dan Costelloe, Conor Ryan

Saving Effort in Parallel GP by Means of Plagues 269
F. Fernández, A. Martín

Sampling of Unique Structures and Behaviours
in Genetic Programming 279
Steven Gustafson, Edmund K. Burke, Graham Kendall

The Evolution of Concurrent Control Software Using
Genetic Programming 289
John Hart, Martin Shepperd

Extending Grammatical Evolution to Evolve Digital Surfaces
with Genr8 299
Martin Hemberg, Una-May O'Reilly

Evolving Text Classifiers with Genetic Programming 309
Laurence Hirsch, Masoud Saeedi, Robin Hirsch

Automatic Synthesis of Instruction Decode Logic
by Genetic Programming 318
David Jackson

Alternatives in Subtree Caching for Genetic Programming 328
Maarten Keijzer

Structural Risk Minimization on Decision Trees Using
an Evolutionary Multiobjective Optimization 338
DaeEun Kim

Global Distributed Evolution of L-Systems Fractals 349
W.B. Langdon

Reusing Code in Genetic Programming 359
Edgar Galván López, Riccardo Poli, Carlos A. Coello Coello

Exploiting Reflection in Object Oriented Genetic Programming 369
Simon M. Lucas

Evolutionary Feature Construction Using Information Gain
and Gini Index 379
Mohammed A. Muharram, George D. Smith

On the Evolution of Evolutionary Algorithms 389
Jorge Tavares, Penousal Machado, Amílcar Cardoso,
Francisco B. Pereira, Ernesto Costa

Genetic Programming with Gradient Descent Search
for Multiclass Object Classification 399
Mengjie Zhang, Will Smart

Author Index 409

Evaluation of Chess Position by Modular Neural Network Generated by Genetic Algorithm

Mathieu Autonès, Ariel Beck, Philippe Camacho, Nicolas Lassabe, Hervé Luga, and François Scharffe

Institut de Recherche en Informatique de Toulouse, Université Paul Sabatier, 118 route de Narbonne 31062 Toulouse cedex, France

Abstract. In this article we present our chess engine Tempo. One of the major difficulties for this type of program lies in the function for evaluating game positions. This function is composed of a large number of parameters which have to be determined and then adjusted. We propose an alternative which consists in replacing this function by an artificial neuron network (ANN). Without topological knowledge of this complex network, we use the evolutionist methods for its inception, thus enabling us to obtain, among other things, a modular network. Finally, we present our results:

- reproduction of the XOR function which validates the method used
- generation of an evaluation function

1 Introduction

The game position evaluation function is a key part in a chess engine. It is composed of a long list of parameters [1], and using a genetic algorithm (GA) [8] to optimise them is relatively efficient, these parameters being obtained from extensive game experience. Another method, which consists of substituting an ANN for this list seems more interesting, because of the generalising capabilities of this model. In practice it is more difficult to implement. Network topology determination is the biggest problem. Evolutionist methods may help us, by evolving a network population which codes this function. We could code the matrix just as it is, in the chromosome [9], connections evolving through successive generations. This coding turns out to be unsatisfactory: the matrix size has to be prefixed and it is hard to predict it a priori. Network encoding then becomes the main problem of our study.

Boers and Kuiper's work [2] allows us, by using L-Systems, to generate modular neural networks whose size is independent of that of the chromosome, and crossover tolerant. We generate a population of L-System construction rules and then mark the resulting networks according to their capabilities to learn game position evaluations from real games. These positions are evaluated more and more deeply in the game tree, along with the increased complexity of the network.

M. Keijzer et al. (Eds.): EuroGP 2004, LNCS 3003, pp. 1–10, 2004.

2 Chess Engine

A chess engine contains three distinct parts: the management of the rules, investigation of the different variant pathways using a search algorithm, and the evaluation function.

2.1 Chess Rules

All chess engines need to know the rules to generate the legal moves or referee a game between two people. All legal moves are pre-calculated in the tables: the engine just needs to confirm that this move is one of them.

2.2 Search Algorithm

The search algorithm explores all the moves from a position and tries to find the best move. In our programme, we use the alphabeta algorithm with various heuristics [10], which are not mentioned here. The values of tree leaves are computed using an evaluation function, which is an ANN in this case.

2.3 Evaluation Function

The evaluation function is very important in all chess engines. It is very complex because it gives the final mark which it uses to select the move to play. The main operation of this function is to count the values of the pieces. After that, it is possible to refine the function, using a series of parameters to define:

- the king's safety
- maintenance of the bishop pair
- domination of the centre
- occupation of the open columns by the rooks.
- ...

We then calculate the sum of all the parameters to obtain the final mark. One of the main limitations of this technique is that we have to define the list of parameters and to set them up correctly, knowing that certain of these values will change during the game and are not self-compensating.

3 Neural Network

3.1 Presentation

Introduction. The human brain is certainly a most amazing organ. It is not a surprise if people try to pierce the secrets of its functioning, to recreate certain of its mechanisms artificially. Neural networks are directly inspired by this vision. The ANN are inspired directly by the structure of the human brain, which is schematically a field of neurons linked together. A Neuron is composed of three parts: dendrites, body and axon. The dendrites get the information (electrical impulses) from the other neurons. The body makes the sum of all this electrical information and if it goes beyond a certain level , the axon is activated, that is to say the neuron sends an electrical impulse to its successors.

The artificial neuron. By starting from these biological considerations, we can find an artificial neural model [12]. The artificial neuron is composed of n inputs which are real numbers and one output (that can be duplicated to power different successor neurons) which is the weighted sum of all the entries. The weights correspond to each one of the connections. It is these weights which will be modified during the learning process.

Learning method: back-propagation. *Principle.* Once a network is created it possesses inputs and outputs. The goal of the learning process is to make sure that, for a given problem, the ANN selects the "correct" outputs as a function of the entries which will be presented. A level is fixed and the learner tries to obtain it.

It is important to note that this learning is supervised. This is like having an expert give a part of some solutions for the entries and the outputs. However, it is impossible to have all the solutions, and one of the properties of the ANN is to generalise. After the learning, it can find the solution for entries which it has never seen (if the structure of the ANN is well adapted). Some other techniques could also be used. Various other learning methods exist. It is also possible to process the learning of a network by a GA. This method could be used once a good topology has beenfound for the network.

Remark. This learning process does not correspond to the humain brain learning process.

3.2 An Example: XOR Function

We have tested a simple example: the learning of XOR function by back-propagation. The particularity of this simple logical function is that it which cannot be learnt by an ANN composed of only one hidden layer.

The logical XOR function is a binary function with two variables:

```
F(0,0) = 0
F(0,1) = 1
F(1,0) = 1
F(1,1) = 0
```

One of the possible topologies for the network is the following matrix (Fig. 1). If we add one connection to it (from 0 to 4), we obtain another topology (Fig. 2) which can resolve the problem.

```
Matrix:       0 1 2 3 4

            0 0 0 1 1 0
            1 0 0 1 1 0
            2 0 0 0 0 1
            3 0 0 0 0 1
            4 0 0 0 0 0
```

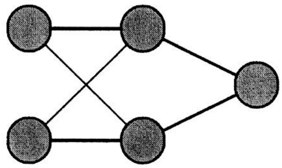


Fig. 1. Topology represented by the matrix

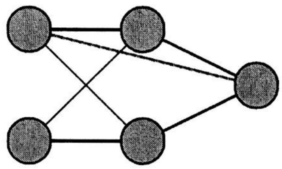


Fig. 2. Topology with one more connection

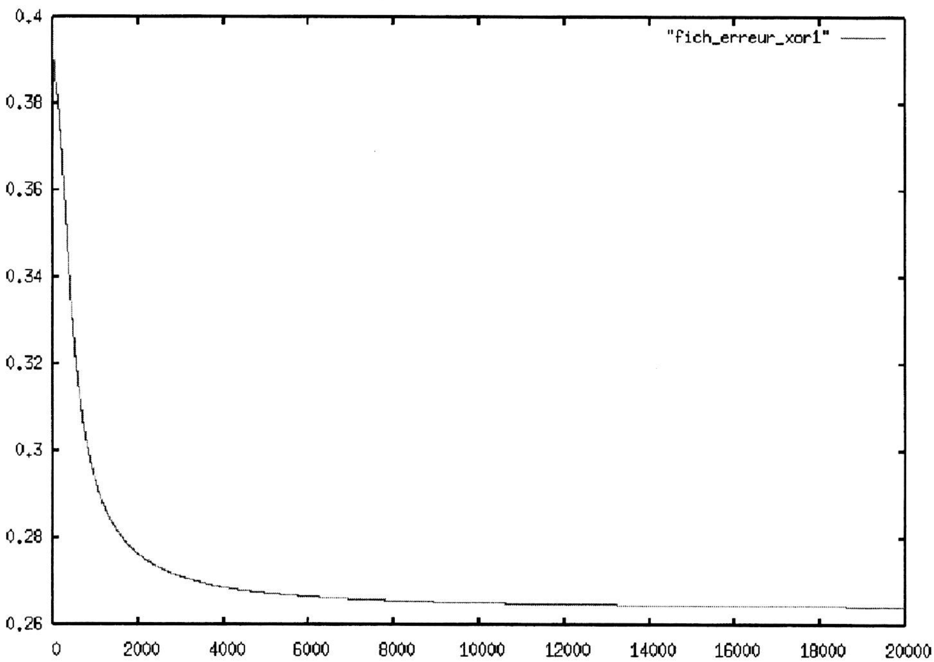


Fig. 3. Learning by the topology of Fig. 1: the graph represents the error on the function for each step of calcul

For the topology of Fig. 1, the error (Fig. 3) does not fall below the local minimum 0.26. In contrast, with the topology of Fig. 2 the learning is better (Fig. 4). This simple example shows how sensitve the learning is to the topology of the neuron.

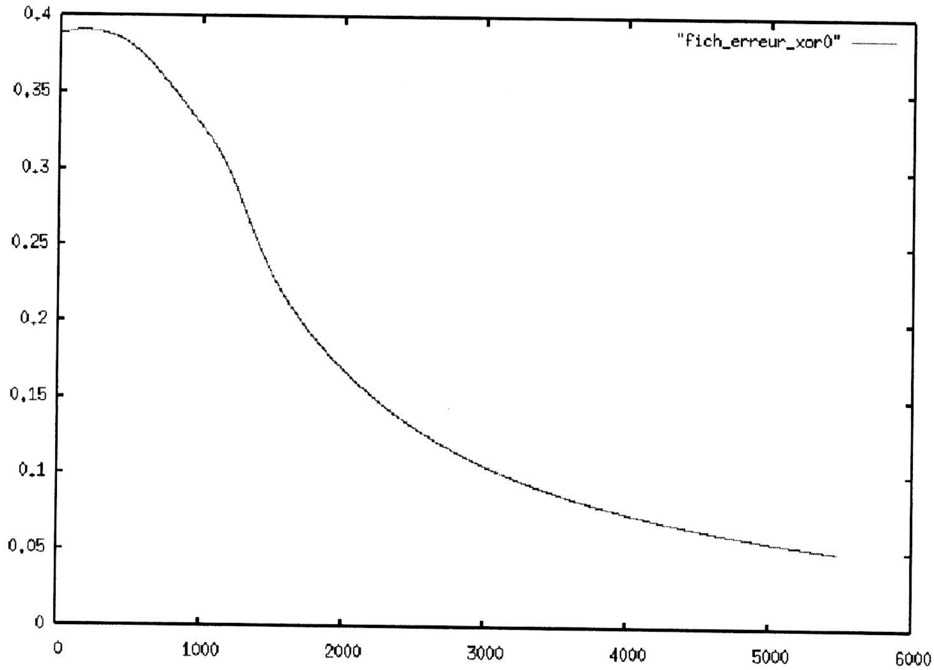


Fig. 4. Learning by the topology of Fig. 2: the graph represents the error on the function for each step of calculation

3.3 Back-Propagation: Limits and Solutions

Problems linked to back-propagation are essentially due to the structure of the network. Regarding this structure, a network will be able to learn, unable to learn, able to learn but need a long time, able to learn but unable to generalise. Another thing is that the net may potentially be able to learn but converges to a local minimum when initial weights are not correctly chosen. It is always the network structure that is responsible for the initial weights sensitivity: some nets are convergent for any set of initial weights, others do not converge every time. We noticed that nets with a modular topology were better for learning. By modularity, we mean the network is made up of many other sub-networks. A connection between two networks is a connection between each output of the first to each input of the second. The next paragraph explains how to create networks with strong modularity.

4 Neural Net Generation with L-Systems

L-Systems [11] are grammar systems generally used for generating artificial plants. They are based on cellular development. We use them to generate modular neural networks. These grammar rules are obtained from a string [4].