

# Contents

<b>1</b>	<b>Introduction to Evolutionary Computation and Genetic Programming</b> .....	1
	Julian F. Miller	
1.1	Evolutionary Computation .....	1
1.1.1	Origins .....	1
1.1.2	Illustrating Evolutionary Computation: The Travelling Salesman Problem .....	2
1.2	Genetic Programming .....	4
1.2.1	GP Representation in LISP .....	5
1.2.2	Linear or Machine Code Genetic Programming .....	6
1.2.3	Grammar-Based Approaches .....	8
1.2.4	PushGP .....	10
1.2.5	Cartesian Graph-Based GP .....	11
1.2.6	Bloat .....	14
	References .....	14
<b>2</b>	<b>Cartesian Genetic Programming</b> .....	17
	Julian F. Miller	
2.1	Origins of CGP .....	17
2.2	General Form of CGP .....	17
2.3	Allelic Constraints .....	19
2.4	Examples .....	20
2.4.1	A Digital Circuit .....	20
2.4.2	Mathematical Equations .....	20
2.4.3	Art .....	22
2.5	Decoding a CGP Genotype .....	24
2.5.1	Algorithms for Decoding a CGP Genotype .....	25
2.6	Evolution of CGP Genotypes .....	28
2.6.1	Mutation .....	28
2.6.2	Recombination .....	29
2.6.3	Evolutionary Algorithm .....	30

- 2.7 Genetic Redundancy in CGP Genotypes ..... 31
- 2.8 Parameter Settings for CGP ..... 31
- 2.9 Cyclic CGP ..... 33
- References ..... 33
  
- 3 Problem Decomposition in Cartesian Genetic Programming ..... 35**  
 James Alfred Walker, Julian F. Miller, Paul Kaufmann  
 and Marco Platzner
- 3.1 Introduction ..... 35
- 3.2 Embedded Cartesian Genetic Programming (ECGP) ..... 36
  - 3.2.1 Genotype Representation ..... 37
  - 3.2.2 Modules ..... 38
  - 3.2.3 Genotype Operators ..... 41
  - 3.2.4 Module Operators ..... 47
  - 3.2.5 Evolutionary Strategy ..... 49
  - 3.2.6 Benchmark Experiments ..... 50
- 3.3 Digital-Adders ..... 60
- 3.4 Symbolic Regression ..... 63
- 3.5 Lawnmower Problem ..... 66
- 3.6 Alternative ECGP Operators ..... 70
  - 3.6.1 Cone-Based and Age-Based Module Creation ..... 70
  - 3.6.2 Cone-Based Crossover ..... 76
- 3.7 Modular Cartesian Genetic Programming (MCGP) ..... 78
  - 3.7.1 Multi-level Module Hierarchy Representation ..... 78
  - 3.7.2 Benchmark Experiments ..... 82
- 3.8 Multi-chromosome Cartesian Genetic Programming (MC-CGP) .. 88
  - 3.8.1 Multi-chromosome Representation ..... 88
  - 3.8.2 Multi-chromosome Evolutionary Strategy ..... 90
  - 3.8.3 Benchmark Experiments ..... 91
- References ..... 97
  
- 4 Self-Modifying Cartesian Genetic Programming ..... 101**  
 Simon L. Harding, Julian F. Miller and Wolfgang Banzhaf
- 4.1 Introduction ..... 101
  - 4.1.1 Discovering Mathematical Results Using Genetic  
 Programming ..... 102
- 4.2 Overview of Self-Modification ..... 102
- 4.3 SMCGP and Its Relation to CGP ..... 103
  - 4.3.1 Self-Modification Operators ..... 104
  - 4.3.2 Computational Functions ..... 104
  - 4.3.3 Arguments ..... 106
  - 4.3.4 Relative Addressing ..... 107
  - 4.3.5 Input and Output Nodes ..... 107
  - 4.3.6 A Simple Example ..... 108
  - 4.3.7 Discussion ..... 108
  - 4.3.8 And Back to CGP ..... 110

4.4	Solving Computational Problems with SMCGP: Parity . . . . .	110
4.4.1	Definition of Fitness . . . . .	111
4.4.2	Results . . . . .	112
4.4.3	A General Solution to Computing Even-Parity . . . . .	113
4.4.4	Why GP Cannot Solve General Parity Without Iteration . . . . .	116
4.5	SM vs GP vs GA . . . . .	118
4.6	Implementing Incremental Fitness Functions . . . . .	120
4.7	Conclusions . . . . .	122
4.8	Acknowledgements . . . . .	123
	References . . . . .	123
<b>5</b>	<b>Evolution of Electronic Circuits . . . . .</b>	<b>125</b>
	Lukas Sekanina, James Alfred Walker, Paul Kaufmann and Marco Platzner	
5.1	Introduction . . . . .	125
5.2	Direct Evolution of Small Combinational Circuits . . . . .	126
5.2.1	Evolutionary vs Conventional Synthesis of Combinational Circuits . . . . .	126
5.2.2	CGP for Logic Synthesis . . . . .	127
5.2.3	Benchmark Problems . . . . .	128
5.2.4	Summary . . . . .	130
5.3	Multi-objective Evolution of Combinational Circuits . . . . .	131
5.3.1	Multi-objective Fitness Function . . . . .	131
5.3.2	Benchmarks . . . . .	132
5.3.3	Summary . . . . .	136
5.4	Evolution of Polymorphic Circuits . . . . .	136
5.4.1	Polymorphic Electronics . . . . .	137
5.4.2	Gate-Level Evolution of Polymorphic Circuits . . . . .	138
5.4.3	CGP as Optimizer . . . . .	139
5.4.4	REPOMO32: CGP on a Chip . . . . .	140
5.5	Evolution of Multiple-Constant Multipliers . . . . .	143
5.5.1	Multiplierless Multiplication . . . . .	143
5.5.2	Results of CGP . . . . .	144
5.6	CMOS-Level Circuit Evolution . . . . .	145
5.6.1	Intrinsic Parameter Fluctuations . . . . .	146
5.6.2	Modifying CGP for CMOS Design . . . . .	148
5.6.3	Experiments . . . . .	154
5.6.4	Conclusions and Future Work . . . . .	158
5.7	Evolution of Classification Hardware Using a Modular Approach . . . . .	159
5.7.1	Classifier Architecture . . . . .	159
5.7.2	EMG Signal Domain . . . . .	160
5.7.3	Classifier Hardware Representation Model . . . . .	162
5.7.4	Fitness Assignment and Evolutionary Algorithm . . . . .	163
5.7.5	Experiments and Results . . . . .	164
5.8	EvoCaches: Application-Specific Adaptation of Cache Mappings . . . . .	165

- 5.8.1 The EvoCache Concept ..... 166
- 5.8.2 System Simulation and Metrics ..... 168
- 5.8.3 Experiments and Results ..... 170
- 5.8.4 Conclusion ..... 175
- 5.9 Acknowledgements ..... 176
- References ..... 176
  
- 6 Image Processing and CGP ..... 181**  
 Lukas Sekanina, Simon L. Harding, Wolfgang Banzhaf  
 and Taras Kowaliw
- 6.1 Introduction ..... 181
- 6.2 Evolutionary Design of Image Filters for FPGAs ..... 181
  - 6.2.1 Sliding-Window Function ..... 182
  - 6.2.2 Types of Noise ..... 183
  - 6.2.3 Conventional Filters ..... 184
  - 6.2.4 Edge Detectors ..... 186
  - 6.2.5 Basic Approach to Filter Evolution ..... 186
  - 6.2.6 Bank of Evolved Filters ..... 187
  - 6.2.7 Extended Kernel ..... 188
  - 6.2.8 Experimental Results ..... 188
  - 6.2.9 Summary ..... 196
- 6.3 Evolving Advanced Image Filters ..... 196
  - 6.3.1 Fitness Function ..... 198
  - 6.3.2 Changes to the Standard CGP Genotype ..... 198
  - 6.3.3 Evolutionary Algorithm, Parameters and Function Set ... 198
  - 6.3.4 Results ..... 199
- 6.4 The Automated Design of Features for Image Classification ..... 205
  - 6.4.1 Motivation ..... 205
  - 6.4.2 The Model ..... 206
  - 6.4.3 Transform Evolution ..... 209
  - 6.4.4 Future Directions ..... 212
- 6.5 Acknowledgements ..... 212
- References ..... 213
  
- 7 CGP Acceleration Using Field-Programmable Gate Arrays ..... 217**  
 Lukas Sekanina and Zdenek Vasicek
- 7.1 Reconfigurable Chips ..... 217
- 7.2 Field-Programmable Gate Arrays ..... 218
- 7.3 Hardware Accelerators for CGP ..... 219
  - 7.3.1 Architecture Overview ..... 220
  - 7.3.2 VRC for Symbolic Regression Problems ..... 223
  - 7.3.3 VRC for Combinational-Circuit Evolution ..... 225
- 7.4 Performance Evaluation ..... 226
  - 7.4.1 Evolution of Combinational Circuits ..... 226
  - 7.4.2 Symbolic Regression Problems ..... 228
- 7.5 Summary ..... 229

7.6	Acknowledgements .....	229
	References .....	229
<b>8</b>	<b>Hardware Acceleration for CGP: Graphics Processing Units</b> .....	<b>231</b>
	Simon L. Harding and Wolfgang Banzhaf	
8.1	Graphics Processing Units .....	231
8.2	The Architecture of Graphics Processing Units .....	231
8.3	Programming a GPU .....	233
8.4	Parallel Implementation of GP .....	234
8.5	Initial GPU Results .....	235
	8.5.1 Floating-Point-Based Expressions .....	236
	8.5.2 Binary .....	237
	8.5.3 Regression and Classification .....	237
8.6	Image Processing on the GPU .....	240
	8.6.1 Evolving Image Filters Using Accelerator .....	243
	8.6.2 Results .....	243
8.7	CUDA Implementation .....	245
	8.7.1 Algorithm .....	246
	8.7.2 Compilation and Code Generation .....	247
	8.7.3 Fitness Function .....	251
	8.7.4 Results .....	251
8.8	Conclusions .....	252
8.9	Acknowledgements .....	252
	References .....	253
<b>9</b>	<b>The CGP Developmental Network</b> .....	<b>255</b>
	Gul Muhammad Khan and Julian F. Miller	
9.1	Introduction .....	255
9.2	Biology of Neurons .....	257
9.3	The CGP Developmental Network .....	258
	9.3.1 Health, Resistance, Weight and State-Factor .....	260
	9.3.2 Cartesian Genetic Program (Chromosome) .....	260
	9.3.3 Inputs and Outputs .....	263
9.4	CGP Model of Neuron .....	263
	9.4.1 Electrical Processing .....	263
	9.4.2 Weight Processing .....	268
	9.4.3 Life Cycle of Neuron .....	268
9.5	Applications .....	270
	9.5.1 Wumpus World Problem .....	271
	9.5.2 Competitive Learning Scenario .....	277
9.6	Learning ‘How to Play’ Checkers .....	285
	9.6.1 Coevolution of Two Agents Playing Checkers .....	285
	9.6.2 An Agent Plays Against a Minimax-Based Checkers Program .....	287
9.7	Conclusions .....	288
	References .....	289

- 10 CGP, Creativity and Art** ..... 293
  - Steve DiPaola and Nathan Sorenson
  - 10.1 Introduction ..... 293
  - 10.2 Creativity and Art ..... 294
  - 10.3 Evolutionary Systems and Creativity ..... 295
  - 10.4 Evolutionary Art ..... 295
  - 10.5 Genetic Programming and Creativity ..... 296
    - 10.5.1 Advantages of CGP in Creative Systems ..... 297
  - 10.6 Implementation ..... 298
    - 10.6.1 Fitness Function ..... 300
    - 10.6.2 Contextual Focus ..... 301
  - 10.7 Results ..... 302
  - 10.8 Conclusions and Future Directions ..... 305
  - 10.9 Acknowledgements ..... 306
  - References ..... 306
  
- 11 Medical Applications of Cartesian Genetic Programming** ..... 309
  - Stephen L. Smith, James Alfred Walker, Julian F. Miller
  - 11.1 Introduction ..... 309
  - 11.2 CGP Applied to the Diagnosis of Breast Cancer ..... 309
    - 11.2.1 Classification of Mammograms Using a Vector  
of Conventional Statistical Features ..... 312
    - 11.2.2 Classification of Mammograms Using Raw Pixel Values . 312
    - 11.2.3 Classification of Mammograms Using Multi-  
chromosome CGP ..... 314
    - 11.2.4 Summary ..... 319
  - 11.3 CGP Applied to the Diagnosis of Parkinson’s Disease ..... 319
  - 11.4 CGP Applied to the Diagnosis of Alzheimer’s Disease ..... 325
  - 11.5 Conclusions ..... 333
  - References ..... 334
  
- Appendix A Resources for Cartesian Genetic Programming** ..... 337
  - A.1 General Advice ..... 337
  - A.2 Web Sites ..... 337
  - A.3 Tutorial Material ..... 337
  - A.4 Software ..... 338
    - A.4.1 CGP in C ..... 338
    - A.4.2 CGP in Java ..... 338
    - A.4.3 CGP in MATLAB ..... 339
    - A.4.4 Evolutionary Art with Laurence Ashmore’s CGP  
Program (in Java) ..... 339
  
- Index** ..... 341



<http://www.springer.com/978-3-642-17309-7>

Cartesian Genetic Programming

Miller, J.F. (Ed.)

2011, XXII, 346 p., Hardcover

ISBN: 978-3-642-17309-7