

University of Exeter
Department of Computer Science

Evolution of Robotic Behaviour Using Gene Expression Programming

Jonathan Mwaura

December 2011

Submitted by Jonathan Mwaura, to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Computer Science, December 2011.

This thesis is available for Library use on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

(signature)

Publications

Some of the material in chapter 2 has been published in:

Mwaura, J. and Keedwell, E. (2009). Adaptive Gene Expression Programming Using a Simple Feedback Heuristic In *Proceedings of the Artificial Intelligence and Simulation of Behaviour(AISB2009)*, Edinburgh, April 2009.

Some of the material in chapter 3 has been published in:

Mwaura, J. and Keedwell, E. (2010). Evolution of Robotic Behaviours Using Gene Expression Programming. In *Proceedings of the Congress on Evolutionary Computation (CEC2010)*, Barcelona, 19th-23rd July 2010.

Some of the material in chapter 4 has been published in:

Mwaura, J. and Keedwell, E. (2011). Evolving modularity in robot behaviour using gene expression programming. *Proceedings of: Towards Autonomous Robotic Systems - 12th Annual Conference (TAROS 2011)*, Sheffield, UK, August 31 - September 2, 2011.

Acknowledgements

I would like to thank my supervisor, Dr. Ed Keedwell, for his help and support throughout my studies. I highly appreciate your precise criticism, timely advise and a genuine interest in my welfare throughout my studies as well as during my stay in England.

I would like to acknowledge my mother, Elizabeth Njeri Mwaura (deceased), for her conviction and insistence that education would lead to a bright future. You were right on the point. Similarly, I would like to thank Mr. John Gathinji Mucheru (my primary school teacher) and Mrs. Ann Njuguna (high school teacher) for it is only through them that I was able to access both high school and university education. Today I am educated because of their goodwill, support, encouragement and self sacrifice.

I would like to offer my gratitudes to Dr. Abel Nyamapfene and Dr. Richard Fredlund for their encouragement and friendship as well as helping with the proof reading.

I would also like to thank my siblings, Esther, Patrick and Margaret Mwaura for their unwavering love, friendship and encouragement. Special gratitudes goes particularly to Esther Mwaura for her humour and optimism. It has kept me going.

I would also like to offer my gratitudes to Susan Munene-Karuthui for her friendship, encouragements and unrelenting support before and during my research work.

Finally, I would like to offer my heartfelt gratitudes to Miss. Tutti Kandukira for her friendship, continuous encouragement and for listening to my ideas even when she did not exactly understand my research.

Abstract

The main objective in automatic robot controller development is to devise mechanisms whereby robot controllers can be developed with less reliance on human developers. One such mechanism is the use of evolutionary algorithms (EAs) to automatically develop robot controllers and occasionally, robot morphology. This area of research is referred to as evolutionary robotics (ER). Through the use of evolutionary techniques such as genetic algorithms (GAs) and genetic programming (GP), ER has shown to be a promising approach through which robust robot controllers can be developed

The standard ER techniques use monolithic evolution to evolve robot behaviour: monolithic evolution involves the use of one chromosome to code for an entire target behaviour. In complex problems, monolithic evolution has been shown to suffer from bootstrap problems; that is, a lack of improvement in fitness due to randomness in the solution set [103, 105, 100, 90]. Thus, approaches to dividing the tasks, such that the main behaviours emerge from the interaction of these simple tasks with the robot environment have been devised. These techniques include the subsumption architecture in behaviour based robotics, incremental learning and more recently the layered learning approach [55, 103, 56, 105, 136, 95]. These new techniques enable ER to develop complex controllers for autonomous robots.

Work presented in this thesis extends the field of evolutionary robotics by introducing Gene Expression Programming (GEP) to the ER field. GEP is a newly developed evolutionary algorithm akin to GA and GP, which has shown great promise in optimisation problems. The presented research shows through experimentation that the unique formulation of GEP genes is sufficient for robot controller representation and development. The obtained results show that GEP is a plausible technique for ER problems. Additionally, it is shown that controllers evolved using GEP algorithm are able to adapt when introduced to new environments.

Further, the capabilities of GEP chromosomes to code for more than one gene have been utilised to show that GEP can be used to evolve manually sub-divided robot behaviours. Additionally, this thesis extends the GEP algorithm by proposing two new evolutionary techniques named multigenic GEP with Linker Evolution (mgGEP-LE) and multigenic GEP with a Regulator Gene (mgGEP-RG). The results obtained from the proposed algorithms show that the new techniques can be used to automatically evolve modularity in robot behaviour. This ability to automate the process of behaviour sub-division and optimisation in a modular chromosome is unique to the GEP formulations discussed, and is an important advance in the development of machines that are able to evolve stratified behavioural architectures with little human intervention.

Contents

1	Introduction	15
1.1	Research questions, aims, and claims	17
1.1.1	Research questions	17
1.1.2	Aims	17
1.1.3	Claims	18
1.2	Thesis overview	18
2	Autonomous Robotics: Design and Control	20
2.1	Robot control system	21
2.2	Robot control approaches	22
2.2.1	Hierarchical/Deliberative paradigm	22
2.2.2	Reactive paradigm	23
2.2.3	Hybrid deliberative/reactive paradigm	26
2.2.4	Learning robot control	29
2.2.5	Evolutionary robotics	31
2.3	Evolutionary algorithms	31
2.3.1	Evolutionary algorithm model	31
2.3.2	Components of evolutionary algorithms	32
2.3.3	Genetic Algorithms	35
2.3.4	Genetic programming	35
2.3.5	Gene expression programming	38
2.4	Evolutionary robotics	47
2.4.1	Evolving neural networks	48
2.4.2	Evolving control programs	52
2.4.3	Evolving intelligent behaviours using GEP	54
2.4.4	Evolution platform	55
2.5	Improving evolutionary robotics	57
2.5.1	Incremental evolution	57
2.5.2	Layered learning	59
2.5.3	Co-evolution	59
2.6	Conclusions	60
3	Evolving Behaviours using GEP	63
3.1	Obstacle avoidance with straight line navigation	64
3.1.1	Related work	64
3.2	Using GEP to evolve an obstacle avoidance behaviour	65
3.2.1	Robot and environment implementation	66

3.2.2	Algorithm parameters	67
3.2.3	Experimental results	69
3.3	Adaptation to new environments	75
3.3.1	Effect of training and testing controllers in similar environments	76
3.3.2	Effect of training controllers in simple environment and testing in a complex environment	77
3.3.3	Testing generalisation using the last population	79
3.4	Sensor based velocity control	81
3.4.1	Experimental results	82
3.5	Evolving monolithic controllers using multigenic GEP	85
3.5.1	Experimental results	85
3.6	Conclusion	94
4	Using Multigenic GEP in Robot Behaviour Sub-division	96
4.1	Modular architectures in ER	97
4.2	Linking functions in multigenic GEP	98
4.3	Evolution of a wall following behaviour	98
4.3.1	Program implementation	100
4.3.2	Fitness function	100
4.3.3	Parameter settings and algorithm parameters	101
4.3.4	Algorithm primitives	102
4.3.5	Experimental results	102
4.3.6	Discussion	108
4.4	Evolving in a 3D world model	109
4.4.1	Experimental set up	109
4.4.2	Fitness function	111
4.4.3	Experimental results	112
4.5	Effect of behaviour coordination mechanism	116
4.5.1	Experimental set up	117
4.5.2	Experimental results	119
4.5.3	Discussion	122
4.6	Conclusion	125
5	Evolving Modularity in Robot Behaviour	127
5.1	Multigenic GEP with Linker Evolution	128
5.1.1	Motivation	129
5.1.2	Experimental set up	129
5.1.3	Results and Discussion	131
5.2	Multigenic GEP with a regulatory gene	133
5.2.1	Implementation	134
5.2.2	Experimental set up	135
5.2.3	Results and Discussion	136
5.3	Algorithm performance analysis	140
5.3.1	Performance comparison	141
5.3.2	Statistical significance	146

5.3.3	Solution convergence	149
5.3.4	Controller performance	152
5.3.5	Mutation effect	164
5.3.6	Discussion	166
5.4	Evolving in 3D environments	167
5.4.1	Experimental set up	168
5.4.2	Results and Discussion	169
5.5	Conclusion	173
6	Conclusions and Further Work	175
6.1	GEP for automatic development of robot controllers	175
6.2	GEP for automatic development of modular robot controllers	176
6.3	Further work	179
6.3.1	Simulator to on-board evolution	179
6.3.2	RoboCup	179
6.3.3	Complex robots	180
	Bibliography	181

List of Tables

3.1	Terminal set and sensor positions	66
3.2	General algorithm parameter settings	68
3.3	Adaptation test: algorithm parameter settings	76
3.4	Phenotype lengths in the last generation controllers	80
3.5	Statistical comparison of the performance in the training and test environment	80
3.6	Sensor based velocity control: Parameter settings	82
3.7	mgGEP-multiple out: Parameter settings	86
3.8	Mann-Whitney U test between mgGEP-multiple out and ugGEP perfor- mances	93
4.1	Success predicate	101
4.2	Parameter settings	101
4.3	Comparison of average evaluations of the GP, ugGEP and mgGEP	103
4.4	3D experiments: Algorithm parameter settings	111
4.5	Robot foraging behaviour: Algorithm parameter settings	119
5.1	mgGEP-LE: Algorithm parameter settings	129
5.2	mgGEP-RG: Algorithm parameter settings	135
5.3	Algorithm parameter settings	141
5.4	mgGEP vs ugGEP	146
5.5	mgGEP-LE vs ugGEP	147
5.6	mgGEP-RG vs ugGEP	147
5.7	mgGEP vs mgGEP-LE	147
5.8	mgGEP vs mgGEP-RG	147
5.9	mgGEP-RG vs mgGEP-LE	148
5.10	Analysing the linking set	159
5.11	Mutation effect on GEP algorithms	164
5.12	Algorithm parameter settings	169

List of Figures

2.1	Hierarchical model	23
2.2	Reactive model	24
2.3	Hybrid deliberative/reactive model	27
2.4	Example of uniform crossover	34
2.5	GP representation of a problem	36
2.6	Flowchart of a GEP algorithm	39
2.7	GEP chromosome, phenotype/expression tree (ET) and resulting coding and non-coding regions	41
2.8	sub-ETs in a multigenic chromosome	42
2.9	Expression Tree (ET) for a multigenic GEP	42
2.10	Biological structure of a Gene Regulation Network.	43
2.11	An example of IS Transposition. The marks the end of head region.	45
2.12	An example of RIS Transposition. The marks the end of head region.	45
2.13	An example of Gene Transposition. The marks the end of head region.	46
2.14	Evolutionary robotics methodology	48
3.1	Training and test environments	67
3.2	Example of a potential robot avoidance controller using terminals and func- tions as reported on Table 3.1.	68
3.3	Progression of the median fitness of the best individual and the median of the population mean fitness over the generations	70
3.4	Evolved obstacle avoidance controller with parameters listed on Table 3.2. The controller was evolved in environment test 2 as shown in Figure 3.1	70
3.5	Success rate with varying chromosome length.	72
3.6	Variation of success rate with number of generations. A comparison using different chromosome lengths is shown	73
3.7	Progression of best individuals in the population over the number of gener- ations using different chromosome lengths.	74
3.8	A comparison of the average population mean fitness across different chro- mosome lengths.	75
3.9	Comparison of frequencies with different phenotypical lengths when con- trollers are tested in similar environments	77
3.10	Comparison of frequencies of different controller lengths when tested in complex environment	78
3.11	Performance of all the final generation controllers in the 20 GEP runs in training environment and the resultant performance in test environment	79

3.12	Progression of the average mean fitness in the population as achieved using ugGEP and ugGEP with sensor based velocity control.	83
3.13	Obstacle avoidance controller evolved using ugGEP with sensor based velocity control.	84
3.14	Comparison of performance of best individuals in new environments.	84
3.16	Comparison of success rate over generations as achieved using ugGEP, ugGEP with sensor based velocity control and with multiple output gene.	86
3.15	Comparison of success rates achieved using ugGEP, ugGEP with sensor based velocity control and with mgGEP multiple out.	87
3.17	Progression of best individual in the population as achieved using ugGEP, ugGEP with sensor based velocity control and with multiple output gene in environment test 2.	88
3.18	Progression of the mean population mean fitness as achieved using ugGEP, ugGEP with sensor based velocity control and mgGEP with multiple output gene in environment test 2.	89
3.19	Progression of best individual in the population as achieved using ugGEP and mgGEP with multiple output.	90
3.20	Mean of the population mean fitness in the population as achieved using ugGEP, ugGEP with sensor based velocity control and mgGEP with multiple output.	91
3.21	Comparison of performance of best individuals in new environments.	92
3.22	Comparison of performance of best individuals in new environments.	93
4.1	A behaviour modularity model	97
4.2	mgGEP behaviour sub-division using 2 genes and a linking function	98
4.3	Five different room types used in the experiment. These room types have been adapted from Lazarus and Hu [78].	99
4.4	Comparison of ugGEP and mgGEP success rates across the five different room types.	104
4.5	Progression of the average fitness of the best individual in the population as achieved through ugGEP and mgGEP in the different room types.	105
4.6	Progression of the average population mean fitness as achieved through ugGEP and mgGEP in the different room types	106
4.7	An example of a controller generated using ugGEP while solving room type 4.	107
4.8	Simplified version of figure 4.7	107
4.9	Example of mgGEP controller evolved while solving room type 4	108
4.10	Robot environments as set up in simbad simulator	110
4.11	Comparison of ugGEP and mgGEP success rates across the five different room types.	112
4.12	Progression of success rates with number of generations using ugGEP and mgGEP in the four room types.	113
4.13	Progression of the average fitness of the best individual in the population as achieved through ugGEP and mgGEP in the different room types.	114

4.14	Progression of the average population mean fitness as achieved through ugGEP and mgGEP in the different room types	115
4.15	Progression of standard fitness with number of generations.	116
4.16	Robot world used in the foraging experiments. The round objects represents “food sources” that the robot picks to improve its energy while navigating.	117
4.17	A behaviour coordination model that relies on robot proximity to obstacles in order to choose a robot task.	120
4.18	Progression of the mean of the best individual in the population, and the average of the population mean fitness over generations as achieved through ugGEP and mgGEP (OA priority).	121
4.19	A behaviour coordination model that relies on robots energy level in order to select a robot task.	121
4.20	A behaviour coordination model that uses robots energy to determine behaviour selection. In this case sub-behaviour selection is determined via cascading priorities.	122
4.21	Progression of the mean fitness of the best individual in the population over generations.	123
4.22	Progression of average performance of the population over generations, with different behaviour coordination mechanisms.	124
5.1	Mean fitness of the best individuals along 200 generations for the population of mgGEP-LE and mgGEP chromosomes.	131
5.2	Mean of the population mean fitness as achieved through mgGEP and mgGEP-LE in the different room types.	132
5.3	mgGEP-RG implementation model. The output of the regulatory gene is a symbol that is then used for decision making	134
5.4	Comparison of success rates achieved in the four different room types using standard mgGEP, mgGEP-LE and mgGEP-RG algorithms.	136
5.5	Comparison of best individual performances in the population as achieved through standard mgGEP and mgGEP-RG in the different room types.	137
5.6	Comparison of progression of average fitness of the population as achieved through mgGEP and mgGEP-RG in the different room types.	138
5.7	an example of a controller evolved using mgGEP-RG	139
5.8	Comparison of success rates achieved using ugGEP, standard mgGEP, mgGEP-LE and mgGEP-RG algorithms.	142
5.9	Progression of the mean fitness of the best individual in the population as achieved through ugGEP, standard mgGEP, mgGEP-LE and mgGEP-RG in the different room types.	143
5.10	Progression of the mean of the population mean fitness as achieved through ugGEP, mgGEP, mgGEP-LE and mgGEP-RG in the different room types.	144
5.11	Progression of the median of the population mean fitness as achieved through ugGEP, mgGEP, mgGEP-LE and mgGEP-RG in the different room types.	145

5.12	P values generated using mann whitney two tailed test. The values are generated from 50 observations in each generation. Each curve shows the comparison with a different algorithms. A summary of all the parameters used in this experiment is shown on Table 5.3.	149
5.13	Q1, Q2 and Q3 values of the best individual in the population as achieved by ugGEP and mgGEP algorithms, plotted every generation.	150
5.14	Q1, Q2 and Q3 values of the best individual in the population as achieved by mgGEP and mgGEP-LE algorithms, plotted every generation	151
5.15	Q1, Q2 and Q3 values of the best individual in the population as achieved by mgGEP and mgGEP-RG algorithms, plotted every generation	152
5.16	An example of a multi-controller evolved using standard mgGEP	153
5.17	An example of a multi-controller evolved using mgGEP-LE.	155
5.18	A multi-controller evolved using mgGEP-LE	156
5.19	An evolved mgGEP-LE controller that replicates the behaviour of a controller evolved using mgGEP	157
5.20	Example of a multi-controller evolved using mgGEP-LE	158
5.21	An example of a multi-controller evolved using mgGEP-RG.	160
5.22	Example of a novel multi-controller evolved using mgGEP-RG	161
5.23	A strategy used by mgGEP-RG to solve wall following problem in room 5. .	162
5.24	A novel mgGEP-RG controller evolved to solve wall following problem in room 5	163
5.25	Mutation effect on ugGEP, mgGEP, mgGEP-LE and mgGEP-RG individuals	165
5.26	Average performance of the best individual in the population across rooms 2-5 evolved using mgGEP, mgGEP-LE and mgGEP-RG.	170
5.27	Average performance of the best individual in the population evolved using mgGEP-RG and different variations of mgGEP-LE algorithm.	171
5.28	Comparison of progression of the fitness of the best individual fitness in the population as achieved through mgGEP, mgGEP-RG and mgGEP-LE in the different room types.	172
5.29	Progression of the best individual in the population as achieved through mgGEP, mgGEP(L) and mgGEP-RG in room 2 and 4.	173
5.30	Progression of success rates with number of generations using mgGEP, mgGEP(L) and mgGEP-RG.	173

Acronyms

Acronym	Meaning
AI	Artificial Intelligence
ADF	Automatically Defined Functions
ANN	Artificial Neural Network
ARL	Adaptive Representation through Learning
AuRA	Autonomous Robot Architecture
BBR	Behaviour Based Robotics
BPGEP	Backtracking Parallel Gene Expression Programming
CGP	Cartesian Genetic Programming
CTRNN	Continuous Time Recurrent Neural Network
EA	Evolutionary Algorithms
EANN	Evolving Artificial Neural Networks
EC	Evolutionary Computation
EP	Evolutionary Programming
ER	Evolutionary Robotics
ES	Evolution Strategies
ET	Expression Tree
FFNN	Feed Forward Neural Network
FLC	Fuzzy Logic Control
GA	Genetic Algorithms
GEP	Gene Expression Programming
GP	Genetic Programming
GRNN	Global Recurrent Neural Network
HGP	Hierarchical Genetic Programming
IAS	Intelligent Autonomous Systems
IFLTE	If Less Than or Equal to
IS	Insertion Sequence Transposition
LGP	Linear Genetic Programming
LISP	LISt Processing
LRNN	Local Recurrent Neural Network
MA	Module Acquisition
MEP	Multi Expression Programming
mgGEP	Multigenic Gene Expression Programming
mgGEP (L)	Multigenic Gene Expression Programming (checking left sensor)
mgGEP - multiple out	Multigenic Gene Expression Programming with multiple output

Acronym	Meaning
mgGEP (OA Priority)	Multigenic Gene Expression Programming (with Obstacle Avoidance as Priority behaviour)
mgGEP-LE	Multigenic Gene Expression Programming with Linker Evolution
mgGEP-RG	Multigenic Gene Expression Programming with Regulator Gene
ORF	Open Reading Frame
RIS	Root Insertion Sequence
SFX	Sensor Fusion Effects
S-R	Stimulus-Response
TGP	Traceless Genetic Programming
ugGEP	Unigenic Gene Expression Programming