

**PAIR BONDS IN GENETIC ALGORITHM**

**LIM TING YEE**

**UNIVERSITI SAINS MALAYSIA**

**2015**

# **PAIR BONDS IN GENETIC ALGORITHM**

by

**LIM TING YEE**

**Thesis submitted in fulfilment of the requirements  
for the degree of  
Doctor of Philosophy**

**July 2015**

## ACKNOWLEDGEMENTS

This work could not have been written without the love, and support from my heroic and perfect parents. To my brother and sisters, my profuse and sincere thanks for their loving kindness and tireless support.

I would also like to acknowledge here the unflagging efforts of my supervisor, Professor Dr. Ahamad Tajudin Khader and co-supervisor, Dr Mohammed Azmi Al-Betar from the School of Computer Science at Universiti Sains Malaysia (USM) and Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Jordan, respectively in all aspects - from supervision, counselling, advice, to encouragement throughout my research; without whom the road of research would not have been sailing smoothly.

I am wildly indebted to Mr Chen Chin Peng, the Senior Managing Director of Promserv Sdn Bhd, for his incredible support and kind consideration. Also, not a day goes by without my reflecting on my good fortune in having so many generous and loyal friends, many of whom provided moral support and practical assistance.

Finally, I would like to extend my warmest thanks to all the people who in various ways, directly or indirectly made this path possible for me.

# TABLE OF CONTENTS

	<b>Page</b>
<b>ACKNOWLEDGEMENTS</b> .....	ii
<b>TABLE OF CONTENTS</b> .....	iii
<b>LIST OF TABLES</b> .....	xi
<b>LIST OF FIGURES</b> .....	xv
<b>LIST OF ABBREVIATIONS</b> .....	xxvii
<b>LIST OF SYMBOLS</b> .....	xxix
<b>LIST OF PUBLICATIONS</b> .....	xxx
<b>ABSTRAK</b> .....	xxxii
<b>ABSTRACT</b> .....	xxxiii

## **CHAPTER 1 – INTRODUCTION**

1.1 Preliminaries .....	1
1.2 Background .....	2
1.2.1 Exploration and Exploitation.....	2
1.2.2 Convergence Behaviour.....	4
1.2.2(a) Dilemma .....	5
1.2.3 Parameter Setting.....	6
1.3 Pair Bond .....	7
1.4 Inspiration and Motivation.....	9
1.5 Research Problem .....	10
1.6 Research Objectives .....	11
1.7 Research Contributions .....	12
1.8 Scope of the Research.....	13
1.9 Organization of Thesis.....	13

## **CHAPTER 2 – LITERATURE REVIEW**

2.1	Preliminaries .....	15
2.2	An Overview of the Standard Genetic Algorithm .....	17
2.2.1	Encoding.....	18
2.2.2	Fitness Function.....	20
2.2.3	Selection .....	20
2.2.4	Crossover .....	21
2.2.4(a)	Roles and Characteristics of Crossover in GAs .....	23
2.2.5	Mutation.....	25
2.2.6	Population Size.....	26
2.3	Schemata, Schema Theorem and Building Block Hypothesis.....	27
2.4	Related Work.....	28
2.4.1	Multiple Offspring Approaches.....	32
2.4.1(a)	Crossover with Multiple Offspring (XMO) .....	33
2.4.1(b)	Multiple Applications of Crossover (MAX) .....	34
2.4.1(c)	Critical Analysis I: Multiple Offspring Approaches .....	37
2.4.2	Parameter Setting.....	39
2.4.2(a)	Critical Analysis II: Parameter Setting .....	42
2.5	Summary .....	43

### **CHAPTER 3 – PROPOSED METHODOLOGY**

3.1	Preliminaries .....	45
3.2	Theoretical Foundation .....	46
3.2.1	Motivation for Pair Bonding in GA.....	46
3.3	The Basic Monogamous Pairs GA (MopGA).....	49
3.3.1	Artificial Pair Bond and Rebond .....	53
3.3.2	Infidelity.....	54
3.3.3	Family and Offspring Competition.....	55

3.3.4	Crossover .....	56
3.3.5	Mutation.....	58
3.3.6	Fitness Evaluation.....	58
3.4	Enhanced Version of the MopGA (AMopGA).....	59
3.4.1	Adaptive Mutation Rate.....	60
3.4.2	Adaptive Pair Bond Rate .....	60
3.4.2(a)	Factors Influencing the Design of Adaptive $G_p$ .....	62
3.4.2(b)	Mathematical Functions .....	64
3.4.2(c)	Formulating Adaptive $G_p$ .....	65
3.4.3	Adaptive Infidelity Rate.....	67
3.4.4	Secondary Mutation.....	67
3.4.5	Adaptive Selection of Crossover Operators.....	70
3.5	Convergence Analysis.....	73
3.6	A Comparison of the Proposed Methodologies to the Literature .....	74
3.6.1	MopGA and AMopGA versus SGA.....	74
3.6.2	MopGA and AMopGA versus State-of-the-art Approaches .....	75
3.7	Salient Features of Proposed Methodologies.....	77
3.7.1	Exploitative.....	77
3.7.2	Diversity Maintenance.....	77
3.7.3	Protecting and Shielding The Weak .....	78
3.7.4	Favouring The Fit .....	78
3.7.5	Competitive Complexity.....	79
3.8	Conclusion .....	80
 <b>CHAPTER 4 – TEST SUITES</b>		
4.1	Preliminaries .....	82
4.2	Test Suite I and Preliminary Test .....	83

4.2.1	Onemax.....	85
4.2.2	Royal Road .....	86
4.2.3	Hierarchical If-And-Only-If .....	87
4.2.4	Deceptive Problem.....	88
4.3	Test Suite II and Case Study .....	89
4.3.1	0/1 Knapsack Problem.....	89
4.3.2	Real-parameter Optimization Problems .....	91
4.4	Experimental Environment .....	93
4.5	Statistical Analysis.....	93
4.6	Performance Criteria.....	94
4.6.1	Solution Quality.....	95
4.6.2	Search Efficiency and Convergence .....	96
4.6.3	Population Diversity .....	97
4.7	Summary .....	97

**CHAPTER 5 – PRELIMINARY TEST I: MONOGAMOUS PAIRS GENETIC ALGORITHM (MopGA)**

5.1	Preliminaries .....	99
5.2	Experiments and Discussion.....	100
5.2.1	Experiment I: Understanding MopGA Convergence Behaviour.....	100
5.2.1(a)	Results and Discussion I .....	101
5.2.2	Experiment II: The Effect of Crossover Rate on MopGA Performance ....	105
5.2.2(a)	Results and Discussion II .....	106
5.2.3	Experiment III: Effect of $g_p$ and $P_i$ on MopGA performance .....	108
5.2.3(a)	Results and Discussion III: RR .....	110
5.2.3(b)	Results and Discussion III: HIFF .....	111
5.2.3(c)	Results and Discussion III: DP.....	113
5.2.3(d)	Summary of Experimental Results III .....	115

5.2.4	Experiment IV: Convergence Behaviour .....	116
5.2.4(a)	Results and Discussion IV .....	116
5.2.5	Experiment V: Diversity Maintenance .....	120
5.2.5(a)	Results and Discussion V .....	120
5.2.6	Experiment VI: Computational Effort.....	123
5.2.6(a)	Results and Discussion VI .....	124
5.3	Conclusion .....	126

**CHAPTER 6 – PRELIMINARY TEST II: ADAPTIVE MONOGAMOUS  
PAIRS GENETIC ALGORITHM (AMopGA)**

6.1	Preliminaries .....	129
6.2	Experiments and Discussion.....	130
6.2.1	Experiment I: Understanding AMopGA Convergence Behaviour.....	130
6.2.1(a)	Results and Discussion I .....	131
6.2.2	Experiment II: Multiple Crossover Operators .....	133
6.2.2(a)	Results and Discussion II .....	134
6.2.3	Experiment III: Characteristics of AMopGA.....	137
6.2.3(a)	Results and Discussion III: Solution Quality .....	137
6.2.3(b)	Results and Discussion III: Convergence Analysis .....	138
6.2.3(c)	Results and Discussion III: Adaptive Pair Bond and Infidelity .	140
6.2.3(d)	Results and Discussion III: Adaptive Multiple Crossover Operators .....	141
6.2.3(e)	Results and Discussion III: Effect of Secondary Mutation .....	145
6.2.4	Experiment IV: Diversity Maintenance .....	147
6.2.4(a)	Results and Discussion IV .....	148
6.2.5	Experiment V: Computational Effort.....	150
6.2.5(a)	Results and Discussion V .....	150
6.3	Conclusion .....	153



## **CHAPTER 7 – CASE STUDY I: 0/1 KNAPSACK PROBLEMS**

7.1	Preliminaries .....	156
7.2	Experimental Setup.....	156
7.2.1	Problem-specific Constraint Handling.....	157
7.2.2	Encoding.....	157
7.2.3	MopGA.....	157
7.2.4	AMopGA.....	158
7.2.5	SGA.....	158
7.3	Experiments and Discussion.....	158
7.3.1	Experiment I: Performance Analysis on MopGA and AMopGA .....	159
7.3.1(a)	Results and Discussion I: Low-dimensional Problems .....	159
7.3.1(b)	Results and Discussion I: High-dimensional Problems .....	163
7.3.2	Experiment II: Convergence Behaviour .....	167
7.3.2(a)	Results and Discussion II .....	167
7.3.3	Experiment III: Diversity Maintenance .....	169
7.3.3(a)	Results and Discussion III.....	169
7.4	Conclusion .....	171

## **CHAPTER 8 – CASE STUDY II: IEEE-CEC’13 REAL-PARAMETER SINGLE OBJECTIVE OPTIMIZATION PROBLEMS**

8.1	Preliminaries .....	173
8.2	Experimental Setup.....	174
8.2.1	Encoding.....	174
8.2.2	MopGA.....	174
8.2.3	AMopGA.....	175
8.3	Experiments and Discussion.....	178
8.3.1	Experiment I: Behaviour of Adaptive Parameters in AMopGA.....	178
8.3.1(a)	Results and Discussion I .....	178

8.3.2	Experiment II: The Effect of Secondary Levy Mutation on AMopGA Performance.....	180
8.3.2(a)	Results and Discussion II .....	180
8.3.3	Experiment III: The Effect of Multiple Crossover Operators on AMopGA Performance.....	181
8.3.3(a)	Results and Discussion III.....	182
8.3.4	Experiment IV: Performance Analysis on MopGA and AMopGA.....	183
8.3.4(a)	Results and Discussion IV .....	184
8.3.5	Experiment V: Algorithm Complexity .....	185
8.3.5(a)	Results and Discussion V .....	186
8.3.6	Experiment VI: Comparison with State-of-the-art Approaches .....	186
8.3.6(a)	Results and Discussion VI .....	187
8.4	Conclusion .....	193

## **CHAPTER 9 – CONCLUSION**

9.1	Preliminaries .....	195
9.2	Contribution Summary.....	196
9.3	Future Work .....	198
9.4	Final Remarks .....	200

<b>REFERENCES</b> .....	201
-------------------------	-----

<b>APPENDICES</b> .....	214
-------------------------	-----

<b>APPENDIX A – TEST SUITES</b> .....	215
---------------------------------------	-----

A.1	Data Set for 0/1 Knapsack Problems .....	215
-----	--	-----

<b>APPENDIX B – PRELIMINARY TEST I: MopGA</b> .....	223
---	-----

B.1	Experimental Results II.....	223
-----	------------------------------	-----

B.2	Experimental Results III .....	224
-----	--------------------------------	-----

B.3	Experimental Results VI.....	227
-----	------------------------------	-----

<b>APPENDIX C – PRELIMINARY TEST II: AMopGA</b> .....	228
C.1 Experimental Results II.....	228
C.2 Experimental Results III .....	229
C.3 Experimental Results V .....	230
<b>APPENDIX D – CASE STUDY I: 0/1 KNAPSACK PROBLEMS</b> .....	231
D.1 Experimental Results I.....	231
D.2 Experimental Results II & III .....	232
<b>APPENDIX E – CASE STUDY II: IEEE-CEC’13</b> .....	238
E.1 Crossover Operators.....	238
E.1.1 BLX-alpha Crossover .....	238
E.1.2 Convex Crossover .....	239
E.1.3 Arithmetic Crossover.....	239
E.1.4 Local Crossover .....	240
E.1.5 Laplace Crossover .....	240
E.2 Mutation Operator.....	241
E.2.1 Experimental Results IV.....	241
E.3 Comparative Methodologies and References .....	242
E.4 Experimental Results IV .....	243
E.4.1 10D Problems .....	243
E.4.2 30D Problems .....	246
E.4.3 50D Problems .....	249

## LIST OF TABLES

		Page
Table 3.1	Notations used in MopGA and AMopGA.....	50
Table 3.2	Summary of differences in pair bond tenure $g_p$ (MopGA) and pair bond rate $G_p$ (AMopGA). Both parameters have the effect of controlling the length of pair bond. ....	62
Table 3.3	A comparison of features among SGA, MopGA, and AMopGA.....	75
Table 3.4	A comparison of features between MopGA, AMopGA and State-of-the-art approaches.....	77
Table 3.5	Computational complexity.....	79
Table 5.1	Outline of the series of experiments to be conducted on MopGA.....	99
Table 5.2	Onemax: a sample reproduction progress for family $\mathcal{U}_1^{(1-14)}$ during the pair bond session from generations 1 to 14. Columns $t$ denotes generation number and $F$ refers to the family fitness. The chromosomes of parents and their two best offspring are shown in the last two columns with values in brackets denoting their fitness. ....	104
Table 5.3	Parameter setup for determining the effect of $g_p$ and $P_i$ on MopGA.....	109
Table 6.1	Outline of the series of experiments to be conducted on AMopGA.....	129
Table 7.1	Summary of program parameters for MopGA, AMopGA and SGA .....	159
Table 7.2	Mean ( <i>Mean</i> ) and standard deviations ( <i>SD</i> ) best fitness of AMopGA, MopGA and SGA in solving low-dimensional 0/1-KPs with hypothesis test using multiple comparison ANOVA and post-hoc Dunnett's test, where applicable. ....	160
Table 7.3	Mean ( <i>Mean</i> ) and standard deviations ( <i>SD</i> ) processing time (in milliseconds) to completion for AMopGA, MopGA and SGA on low-dimensional 0/1-KPs with hypothesis test using multiple comparison ANOVA and post-hoc Dunnett's test, where applicable. ....	161
Table 7.4	Comparing the experimental results for low-dimensional 0/1-KPs against other state-of-the-art approaches.....	163
Table 7.5	Mean ( <i>Mean</i> ) and standard deviations ( <i>SD</i> ) best fitness of AMopGA, MopGA and SGA in solving low-dimensional 0/1-KPs with hypothesis test using multiple comparison ANOVA and post-hoc Dunnett's test, where applicable. ....	164

Table 7.6	Mean ( <i>Mean</i> ) and standard deviations ( <i>SD</i> ) processing time (in milliseconds) to completion for AMopGA, MopGA and SGA on high-dimensional 0/1-KPs with hypothesis test using multiple comparison ANOVA and post-hoc Dunnett’s test, where applicable. ....	165
Table 8.1	Parameter settings for MopGA on IEEE-CEC’ 13 test functions .....	175
Table 8.2	Parameter settings for AMopGA on IEEE-CEC’ 13 test functions.....	176
Table 8.3	Wilcoxon test for AMopGA with and without secondary Levy mutation. The best results are highlighted in <b>bold</b> . .....	181
Table 8.4	Average rankings of AMopGA with different crossover operators.....	182
Table 8.5	Algorithm complexity for AMopGA and MopGA in solving IEEE-CEC’ 13 test problems.....	186
Table 8.6	Results of Friedman tests across all dimensions ( $\alpha = 0.05$ ) .....	187
Table 8.7	$p$ -values on all functions (Control algorithm = AMopGA).....	191
Table 8.8	$p$ -values on unimodal functions, $f_1$ - $f_5$ (Control algorithm = AMopGA)	191
Table 8.9	$p$ -values on multimodal functions, $f_6$ - $f_{20}$ (Control algorithm = AMopGA).....	192
Table 8.10	$p$ -values on composite functions, $f_{21}$ - $f_{28}$ (Control algorithm = AMopGA).....	192
Table A.1	Dataset for Low-dimensional 0/1 Knapsack Problems.....	215
Table A.2	Dataset for High-dimensional 0/1 Knapsack Problems.....	216
Table B.1	Mean cut-off generation ( <i>Cutoff</i> ), best fitness (Best), and percentage success rate to optimum ( <i>SR</i> ) required by MopGA ( $g_p = 0.04$ , $P_i = 0.3$ ) to solve the RR, HIFF, and DP functions using various $P_c$ rates. The best cut-off generation per function is highlighted in <b>bold</b> . ...	223
Table B.2	Mean cut-off generation ( <i>Cutoff</i> ), best fitness (Best), and percentage success rate to optimum ( <i>SR</i> ) required by MopGA to solve the RR function using various combinations of $g_p$ and $P_i$ . Best cut-off generation is highlighted in <b>bold</b> . .....	224
Table B.3	Mean cut-off generation ( <i>Cutoff</i> ), best fitness (Best), and percentage success rate to optimum ( <i>SR</i> ) required by MopGA to solve HIFF function using various combinations of $g_p$ and $P_i$ . Best cut-off generation is highlighted in <b>bold</b> . .....	225

Table B.4	Mean cut-off generation ( <i>Cutoff</i> ), best fitness (Best), and percentage success rate to optimum ( <i>SR</i> ) required by MopGA to solve the DP function using various combinations of $g_p$ and $P_i$ . Best cut-off generation is highlighted in <b>bold</b> .....	226
Table B.5	Simulation details in terms of operation time for RR, HIFF, and DP functions through MopGA and SGA. ....	227
Table B.6	Comparing MopGA and SGA using student-t tests with 95% confidence interval for the means of processing time to completion. The symbol '+' denotes MopGA better than SGA, '-' if it is worse, and '=' if they do not have statistical difference.....	227
Table C.1	Mean cut-off generations ( <i>Cutoff</i> ) required for AMopGA to solve the 64-bit onemax problem, standard deviation (in brackets), and percentage success rate ( <i>SR</i> ) over 50 runs for various crossover operators. ....	228
Table C.2	Comparing the mean cut-off generation for RR, HIFF and DP functions on AMopGA with secondary mutation at infidelity on elite or random individuals and without secondary mutation. Each value is a mean of over 50 runs. Values in parenthesis denote standard deviations. ....	229
Table C.3	Student t-test for AMopGA with secondary mutation on elite individual (Elite) against random individual (Random) and against without secondary mutation (None) considering RR, HIFF and DP functions. ....	230
Table C.4	Simulation details in terms of operation time for RR, HIFF, and DP functions through AMopGA and MopGA.....	230
Table C.5	Comparing AMopGA and MopGA using student-T tests with 95% confidence interval for the means of processing time to completion. The symbol '+' denotes AMopGA better than MopGA, '-' if it is worse, and '=' if they do not have statistical difference. ....	230
Table D.1	Results from hypothesis test using multiple comparison ANOVA and post-hoc Dunnett's test, where applicable to compare AMopGA against MopGA and SGA on solving low- and high-dimensional 0/1-KPs. ....	231
Table D.2	Average processing time per operation on low- and high-dimensional 0/1-KPs .....	232
Table E.1	Simulation details in terms of operation time for $f_{14}$ through MopGA and AMopGA. ....	241
Table E.2	Algorithms from the IEEE-CEC'13 Special Session & Competition on Real-Parameter Single Objective Optimization.....	242

Table E.3	Function error values for IEEE-CEC'13 benchmarks on $D = 10, 30, 50$ over 51 runs.....	<b>252</b>
-----------	---	------------

## LIST OF FIGURES

		Page
Figure 1.1	Striatopallidal regulation of pair bonding - a simplified version (Resendez and Aragona, 2013) .....	8
Figure 2.1	An example of mapping between binary genotypic space and phenotypic space. $L$ represents the chromosome length. ....	19
Figure 2.2	Examples of 1-point and 2-point crossover operators .....	22
Figure 2.3	Example of a bit-flip mutation operator. Shaded blocks represent mutated genes. ....	26
Figure 2.4	The different areas of enhancement that take place in GAs. Shaded blocks represent areas of focus in this thesis. ....	29
Figure 2.5	Variants of GA with multiple offspring. ....	32
Figure 2.6	Parameter setting taxonomy adopted from Eiben et al. (1999) .....	39
Figure 3.1	A schematic overview of the two proposed methodologies, MopGA and AMopGA with SGA as the core algorithm.....	46
Figure 3.2	Applying 1-point crossovers on three different positions for complementary parents with common schema $\hat{H} = ****$ and $o(\hat{H}) = 0$ . "x" indicates a 1-point crossover operation.....	48
Figure 3.3	Applying 1-point crossovers on three different positions for parents with common schema $\hat{H} = *0**$ and $o(\hat{H}) = 1$ . "x" indicates a 1-point crossover operation .....	48
Figure 3.4	MopGA framework.....	50
Figure 3.5	Sample graphs showing how $P_m$ changes inversely proportional to $t$ with various $C_1 = \{1, 2, 3, 6\}$ where $G = 5000$ .....	61
Figure 3.6	Sample results of family performance during a pair bond session from generations 1 to 14 for solving the 64-bit Onemax problem. $t$ denotes the generation numbers.....	62
Figure 3.7	Sample plots of functions $y_a = \frac{x}{C}$ and $y_b = \frac{x}{(x+C)}$ on various $C$ values ranging from 16, 32, and 64.....	64
Figure 3.8	Sample plot showing how $G_p$ changes in relation to $F$ for solving the 64-bit Onemax problem.....	66
Figure 3.9	Sample plot showing how $P_i$ changes inversely proportional to $V$ for solving the 64-bit Onemax problem. ....	68



Figure 3.10	The valley landscape of a maximization objective function.....	69
Figure 3.11	Sample distribution of 5 crossover operators used in solving the function $f_{21}$ ( $D = 30$ ) of IEEE-CEC'13 following the adaptive selection of crossover operators in Eq. (3.17) .....	72
Figure 4.1	Test suite for evaluating MopGA and AMopGA.....	83
Figure 5.1	Convergence plot of MopGA on 64-bit Onemax.....	101
Figure 5.1(a)	Overview .....	101
Figure 5.1(b)	Zoom-in .....	101
Figure 5.2	Onemax: close-up view of individual family fitness during each pair bond session. $t$ denotes generation of evolution and $\mathcal{U}_i^{(t)}$ represents a family where $i \in \{1, \frac{N}{2}\}$ .....	103
Figure 5.2(a)	$t : 1 - 14: \mathcal{U}_1^{(1-14)}$ .....	103
Figure 5.2(b)	$t : 1 - 14: \mathcal{U}_2^{(1-14)}$ .....	103
Figure 5.2(c)	$t : 15 - 29: \mathcal{U}_1^{(15-29)}$ .....	103
Figure 5.2(d)	$t : 15 - 29: \mathcal{U}_2^{(15-29)}$ .....	103
Figure 5.2(e)	$t : 30 - 44: \mathcal{U}_1^{(30-44)}$ .....	103
Figure 5.2(f)	$t : 30 - 44: \mathcal{U}_2^{(30-44)}$ .....	103
Figure 5.3	Box plots illustrating the cut-off generation for MopGA with various $P_c$ rates. The numeric value within each box represents the mean cut-off generation.....	107
Figure 5.3(a)	RR .....	107
Figure 5.3(b)	HIFF .....	107
Figure 5.3(c)	DP .....	107
Figure 5.4	Interaction plot of $g_p$ and $P_i$ for the mean cut-off generation of RR function solved using MopGA. ....	110
Figure 5.5	Interaction plot of $g_p$ and $P_i$ for the response mean cut-off generation of HIFF function solved using MopGA. ....	112
Figure 5.6	Interaction plot of $g_p$ and $P_i$ for the response mean cut-off generation of DP function solved using MopGA. ....	114

Figure 5.7	Sample convergence plots for the first 500 generations of MopGA and its variants on RR, HIFF and DP functions. The SGA acts as a control algorithm.....	117
Figure 5.7(a)	RR .....	117
Figure 5.7(b)	HIFF.....	117
Figure 5.7(c)	DP .....	118
Figure 5.8	Sample convergence plot of MopGA and its variants on Onemax function.....	120
Figure 5.9	Population Hamming distance averaged over 50 runs on SGA and MopGA for the first 500 generations. Vertical lines marked the generations at which optimum solutions are found. ....	121
Figure 5.9(a)	RR .....	121
Figure 5.9(b)	HIFF.....	121
Figure 5.9(c)	DP .....	121
Figure 5.10	Comparing the simulation time in milliseconds between MopGA and SGA on RR, HIFF and DP functions using bar chart.....	124
Figure 5.11	Pie charts of operation time in RR, HIFF and DP functions for MopGA and SGA. ....	125
Figure 5.11(a)	RR: MopGA .....	125
Figure 5.11(b)	RR: SGA .....	125
Figure 5.11(c)	HIFF: MopGA .....	125
Figure 5.11(d)	HIFF: SGA.....	125
Figure 5.11(e)	DP: MopGA .....	125
Figure 5.11(f)	DP: SGA .....	125
Figure 6.1	Onemax: average families fitness in the entire population, $\bar{F}$ on AMopGA.....	131
Figure 6.2	Onemax: sample plot of family fitness values and $G_p$ rate during a pair bond session between generation $t$ 27 to 38 of family $U_1^{27-38}$ on AMopGA.....	132
Figure 6.3	Onemax: sample plot of survival rate, $V$ and $P_i$ rate during a pair bond session between generation ( $t$ ) 27 to 38 of family $U_2^{27-38}$ on AMopGA.....	133

Figure 6.4	Onemax: sample distribution plot of infidelity, rebond, and monogamy occurrences at every 50-th generation for AMopGA averaged over 50 runs. $t$ denotes generation number.....	133
Figure 6.4(a)	Onemax: distribution plot .....	133
Figure 6.4(b)	Onemax: pie chart .....	133
Figure 6.5	Box plots illustrating the distribution of cut-off generation to solving the Onemax function with AMopGA using various combinations of crossover operators. Values marked by triangular shapes indicate the mean cut-off generations of 50 runs. ....	134
Figure 6.6	Distribution of crossover operators when using multiple choice crossover operator in AMopGA under Test OM4 and OM5, respectively. ....	136
Figure 6.6(a)	OM4.....	136
Figure 6.6(b)	OM5.....	136
Figure 6.7	Comparing AMopGA performance against MopGA for solving the RR, HIFF, and DP functions.....	138
Figure 6.8	Sample convergence plots for the first 500 generations of AMopGA on RR, HIFF and DP functions. Graphs on the left denote comparisons of the best fitness between AMopGA and MopGA, while graphs on the right represent the population best, average and worst fitness of AMopGA over time (generation, $t$ ) sampled at every 10-th generation. ....	139
Figure 6.8(a)	RR: AMopGA vs MopGA .....	139
Figure 6.8(b)	RR: Population best, average and worst fitness (AMopGA).....	139
Figure 6.8(c)	HIFF: AMopGA vs MopGA .....	139
Figure 6.8(d)	HIFF: Population best, average and worst fitness (AMopGA). ....	139
Figure 6.8(e)	DP: AMopGA vs MopGA .....	139
Figure 6.8(f)	DP: Population best, average and worst fitness (AMopGA). ....	139
Figure 6.9	Distribution plots (at every 10-th generation) and pie charts of infidelity, rebond, and monogamy occurrences in AMopGA on RR, HIFF and DP functions. $t$ denotes the generation number. ....	140
Figure 6.9(a)	RR: distribution plot.....	140
Figure 6.9(b)	RR: pie chart .....	140
Figure 6.9(c)	HIFF: distribution plot.....	140

Figure 6.9(d)	HIFF: pie chart.....	140
Figure 6.9(e)	DP: distribution plot .....	141
Figure 6.9(f)	DP: pie chart .....	141
Figure 6.10	Distribution plots (at every 10-th generation) and pie charts for AMopGA with multiple and adaptive crossover operators on RR, HIFF, and DP problems. $t$ denotes the generation number.....	142
Figure 6.10(a)	RR: distribution plot .....	142
Figure 6.10(b)	RR: pie chart .....	142
Figure 6.10(c)	HIFF: distribution plot.....	142
Figure 6.10(d)	HIFF: pie chart.....	142
Figure 6.10(e)	DP: distribution plot .....	142
Figure 6.10(f)	DP: pie chart .....	142
Figure 6.11	Crossover operators distribution along with average population fitness plotted at intervals of 10 generations apart for the first 500 generations of evolution.....	144
Figure 6.11(a)	RR .....	144
Figure 6.11(b)	HIFF.....	144
Figure 6.11(c)	DP .....	144
Figure 6.12	Boxplots showing the cut-off generation for RR, HIFF and DP functions on AMopGA with secondary mutation at infidelity on elite or random individuals and without secondary mutation. Means cut-off generation are marked by points and numerical values.....	145
Figure 6.12(a)	RR .....	145
Figure 6.12(b)	HIFF.....	145
Figure 6.12(c)	DP .....	145
Figure 6.13	Population Hamming distance averaged over 50 runs for MopGA and AMopGA on various test problems. $t$ denotes the generation number. Vertical lines marked the generations at which optimum solutions are found.....	149
Figure 6.13(a)	RR .....	149
Figure 6.13(b)	HIFF.....	149
Figure 6.13(c)	DP .....	149

Figure 6.14	Comparing the simulation time in milliseconds between MopGA and AMopGA on RR, HIFF and DP functions using bar chart.....	151
Figure 6.15	Pie charts of operation time in RR, HIFF and DP functions for MopGA and AMopGA.....	152
Figure 6.15(a)	RR: MopGA .....	152
Figure 6.15(b)	RR: AMopGA .....	152
Figure 6.15(c)	HIFF: MopGA .....	152
Figure 6.15(d)	HIFF: AMopGA .....	152
Figure 6.15(e)	DP: MopGA .....	152
Figure 6.15(f)	DP: AMopGA .....	152
Figure 7.1	Comparing the mean processing time (ms) to completion among AMopGA, MopGA and SGA on low-dimensional 0/1-KPs.....	161
Figure 7.2	Comparing the mean processing time (in seconds) to completion among AMopGA, MopGA and SGA on high-dimensional 0/1-KPs.....	166
Figure 7.3	Percentage processing time per operation on average for high-dimensional 0/1-KPs. The corresponding data can be found in Table D.2.....	167
Figure 7.3(a)	AMopGA .....	167
Figure 7.3(b)	MopGA .....	167
Figure 7.3(c)	SGA .....	167
Figure 7.4	Sample convergence and diversity plots averaged over 50 runs for selective 0/1-KPs. $t$ denotes generation number.....	168
Figure 7.4(a)	Instance 14: Solution quality .....	168
Figure 7.4(b)	Instance 14: Diversity .....	168
Figure 7.4(c)	Instance 15: Solution quality .....	168
Figure 7.4(d)	Instance 15: Diversity .....	168
Figure 7.4(e)	Instance 25: Solution quality .....	168
Figure 7.4(f)	Instance 25: Diversity .....	168
Figure 8.1	Parameter adaptation process of AMopGA on $f_4$ , $f_{10}$ and $f_{21}$ .....	179
Figure 8.1(a)	$f_4$ : Adaptation of $G_p$ .....	179

Figure 8.1(b)	$f_4$ : Adaptation of $P_i$ .....	179
Figure 8.1(c)	$f_4$ : Adaptation of $P_x^k$ .....	179
Figure 8.1(d)	$f_{10}$ : Adaptation of $G_p$ .....	179
Figure 8.1(e)	$f_{10}$ : Adaptation of $P_i$ .....	179
Figure 8.1(f)	$f_{10}$ : Adaptation of $P_x^k$ .....	179
Figure 8.1(g)	$f_{21}$ : Adaptation of $G_p$ .....	179
Figure 8.1(h)	$f_{21}$ : Adaptation of $P_i$ .....	179
Figure 8.1(i)	$f_{21}$ : Adaptation of $P_x^k$ .....	179
Figure 8.2	Bonferroni-Dunn’s graphic corresponding to the results for all functions on $D = 30$ .....	184
Figure 8.3	Bonferroni-Dunn’s graphic corresponding to the results for all functions across all problem dimensions.....	188
Figure 8.4	Bonferroni-Dunn’s graphic corresponding to the results for unimodal functions, $f_1$ - $f_5$ . ....	188
Figure 8.5	Bonferroni-Dunn’s graphic corresponding to the results for multimodal functions, $f_6$ - $f_{20}$ . ....	189
Figure 8.6	Bonferroni-Dunn’s graphic corresponding to the results for composite functions, $f_{21}$ - $f_{28}$ . ....	189
Figure C.1	Output of post-hoc Tukey Honesty Test for Experiment II Section 6.2.2. When the confidence intervals for mean differences contain zero, the test pairs do not differ among themselves. For instance, the mean difference for the pair OM2-OM1 is not significant. ....	228
Figure C.2	Distribution of crossover operators when using multiple choice crossover operator in AMopGA under Test OM6 and OM7, respectively. ....	229
Figure C.2(a)	OM6.....	229
Figure C.2(b)	OM7.....	229
Figure D.1	Convergence and diversity plots averaged over 50 runs for low-dimensional 0/1-KPs. $t$ denotes generation number.....	232
Figure D.1(a)	Instance 1: Solution quality.....	232
Figure D.1(b)	Instance 1: Diversity.....	232

Figure D.1(c)	Instance 2: Solution quality .....	232
Figure D.1(d)	Instance 2: Diversity .....	232
Figure D.1(e)	Instance 3: Solution quality .....	233
Figure D.1(f)	Instance 3: Diversity .....	233
Figure D.1(g)	Instance 4: Solution quality .....	233
Figure D.1(h)	Instance 4: Diversity .....	233
Figure D.1(i)	Instance 5: Solution quality .....	233
Figure D.1(j)	Instance 5: Diversity .....	233
Figure D.1(k)	Instance 6: Solution quality .....	233
Figure D.1(l)	Instance 6: Diversity .....	233
Figure D.1(m)	Instance 7: Solution quality .....	234
Figure D.1(n)	Instance 7: Diversity .....	234
Figure D.1(o)	Instance 8: Solution quality .....	234
Figure D.1(p)	Instance 8: Diversity .....	234
Figure D.1(q)	Instance 9: Solution quality .....	234
Figure D.1(r)	Instance 9: Diversity .....	234
Figure D.1(s)	Instance 10: Solution quality .....	234
Figure D.1(t)	Instance 10: Diversity .....	234
Figure D.1(u)	Instance 11: Solution quality .....	235
Figure D.1(v)	Instance 11: Diversity .....	235
Figure D.1(w)	Instance 12: Solution quality .....	235
Figure D.1(x)	Instance 12: Diversity .....	235
Figure D.1(y)	Instance 13: Solution quality .....	235
Figure D.1(z)	Instance 13: Diversity .....	235
Figure D.1(aa)	Instance 16: Solution quality .....	235
Figure D.1(bb)	Instance 16: Diversity .....	235
Figure D.2	Convergence and diversity plots averaged over 50 runs for high-dimensional 0/1-KPs. $t$ denotes generation number.....	236

Figure D.2(a)	Instance 17: Solution quality .....	236
Figure D.2(b)	Instance 17: Diversity .....	236
Figure D.2(c)	Instance 18: Solution quality .....	236
Figure D.2(d)	Instance 18: Diversity .....	236
Figure D.2(e)	Instance 19: Solution quality .....	236
Figure D.2(f)	Instance 19: Diversity .....	236
Figure D.2(g)	Instance 20: Solution quality .....	236
Figure D.2(h)	Instance 20: Diversity .....	236
Figure D.2(i)	Instance 21: Solution quality .....	237
Figure D.2(j)	Instance 21: Diversity .....	237
Figure D.2(k)	Instance 22: Solution quality .....	237
Figure D.2(l)	Instance 22: Diversity .....	237
Figure D.2(m)	Instance 23: Solution quality .....	237
Figure D.2(n)	Instance 23: Diversity .....	237
Figure D.2(o)	Instance 24: Solution quality .....	237
Figure D.2(p)	Instance 24: Diversity .....	237
Figure E.1	10D Convergence plots .....	243
Figure E.1(a)	$f_1$ .....	243
Figure E.1(b)	$f_2$ .....	243
Figure E.1(c)	$f_3$ .....	243
Figure E.1(d)	$f_4$ .....	243
Figure E.1(e)	$f_5$ .....	243
Figure E.1(f)	$f_6$ .....	243
Figure E.1(g)	$f_7$ .....	243
Figure E.1(h)	$f_8$ .....	243
Figure E.1(i)	$f_9$ .....	243
Figure E.1(j)	$f_{10}$ .....	244



Figure E.1(k)	$f_{11}$	244
Figure E.1(l)	$f_{12}$	244
Figure E.1(m)	$f_{13}$	244
Figure E.1(n)	$f_{14}$	244
Figure E.1(o)	$f_{15}$	244
Figure E.1(p)	$f_{16}$	244
Figure E.1(q)	$f_{17}$	244
Figure E.1(r)	$f_{18}$	244
Figure E.1(s)	$f_{19}$	244
Figure E.1(t)	$f_{20}$	244
Figure E.1(u)	$f_{21}$	244
Figure E.1(v)	$f_{22}$	245
Figure E.1(w)	$f_{23}$	245
Figure E.1(x)	$f_{24}$	245
Figure E.1(y)	$f_{25}$	245
Figure E.1(z)	$f_{26}$	245
Figure E.1(aa)	$f_{27}$	245
Figure E.1(bb)	$f_{28}$	245
Figure E.2	30D Convergence plots	246
Figure E.2(a)	$f_1$	246
Figure E.2(b)	$f_2$	246
Figure E.2(c)	$f_3$	246
Figure E.2(d)	$f_4$	246
Figure E.2(e)	$f_5$	246
Figure E.2(f)	$f_6$	246
Figure E.2(g)	$f_7$	246
Figure E.2(h)	$f_8$	246

Figure E.2(i)	$f_9$ .....	246
Figure E.2(j)	$f_{10}$ .....	246
Figure E.2(k)	$f_{11}$ .....	246
Figure E.2(l)	$f_{12}$ .....	246
Figure E.2(m)	$f_{13}$ .....	247
Figure E.2(n)	$f_{14}$ .....	247
Figure E.2(o)	$f_{15}$ .....	247
Figure E.2(p)	$f_{16}$ .....	247
Figure E.2(q)	$f_{17}$ .....	247
Figure E.2(r)	$f_{18}$ .....	247
Figure E.2(s)	$f_{19}$ .....	247
Figure E.2(t)	$f_{20}$ .....	247
Figure E.2(u)	$f_{21}$ .....	247
Figure E.2(v)	$f_{22}$ .....	247
Figure E.2(w)	$f_{23}$ .....	247
Figure E.2(x)	$f_{24}$ .....	247
Figure E.2(y)	$f_{25}$ .....	248
Figure E.2(z)	$f_{26}$ .....	248
Figure E.2(aa)	$f_{27}$ .....	248
Figure E.2(bb)	$f_{28}$ .....	248
Figure E.3	50D Convergence plots.....	249
Figure E.3(a)	$f_1$ .....	249
Figure E.3(b)	$f_2$ .....	249
Figure E.3(c)	$f_3$ .....	249
Figure E.3(d)	$f_4$ .....	249
Figure E.3(e)	$f_5$ .....	249
Figure E.3(f)	$f_6$ .....	249

Figure E.3(g)	$f_7$	249
Figure E.3(h)	$f_8$	249
Figure E.3(i)	$f_9$	249
Figure E.3(j)	$f_{10}$	249
Figure E.3(k)	$f_{11}$	249
Figure E.3(l)	$f_{12}$	249
Figure E.3(m)	$f_{13}$	250
Figure E.3(n)	$f_{14}$	250
Figure E.3(o)	$f_{15}$	250
Figure E.3(p)	$f_{16}$	250
Figure E.3(q)	$f_{17}$	250
Figure E.3(r)	$f_{18}$	250
Figure E.3(s)	$f_{19}$	250
Figure E.3(t)	$f_{20}$	250
Figure E.3(u)	$f_{21}$	250
Figure E.3(v)	$f_{22}$	250
Figure E.3(w)	$f_{23}$	250
Figure E.3(x)	$f_{24}$	250
Figure E.3(y)	$f_{25}$	251
Figure E.3(z)	$f_{26}$	251
Figure E.3(aa)	$f_{27}$	251
Figure E.3(bb)	$f_{28}$	251

## LIST OF ABBREVIATIONS

<b>0/1-KP</b>	0/1 Knapsack Problem
<b>AMopGA</b>	Adaptive Monogamous Pairs Genetic Algorithm
<b>DP</b>	Deceptive Problem
<b>EA</b>	Evolutionary Algorithm
<b>EP</b>	Evolutionary Programming
<b>ES</b>	Evolutionary Strategies
<b>G3</b>	Generalized Generation Gap
<b>GA</b>	Genetic Algorithms
<b>GIGA</b>	Gene Invariant Genetic Algorithm
<b>GMD+STB</b>	Generation of Multiple Descendants and Selection of the Two Best
<b>HIFF</b>	Hierarchical-If-And-Only-If function
<b>IEEE-CEC'13</b>	Special Session on Real-Parameter Single Objective Optimization at IEEE Congress on Evolutionary Computation 2013
<b>MAX</b>	Multiple Applications of Crossover operation
<b>MCPC</b>	Multiple Crossovers Per Couple
<b>MCPC-FPCS</b>	Multiple Crossovers Per Couple with Fitness Proportional Couple Selection
<b>MCPMA</b>	Multiple Crossovers Per Mating Action
<b>MGG</b>	Minimal Generation Gap

<b>MopGA</b>	Monogamous Pairs Genetic Algorithm
<b>MPC</b>	Multi-parent Crossover
<b>RR</b>	Royal Road function
<b>XMO</b>	Crossover with Multiple Offspring

## LIST OF SYMBOLS

$P_c$  Crossover rate

$P_m$  Mutation rate

$P_i$  Infidelity rate

$P_x^k$  Probability of selecting the  $k$ -th crossover operator from a set of operators

$G$  Maximum number of generations

$G_p$  Adaptive pair bond rate

$g_p$  Pair bond tenure

$L$  Chromosome length

$N$  Population size

$D$  Problem dimensions

$\mathfrak{P}^{(t)}$  A population at generation  $t$

$\rho$  An individual

$\sigma$  A set of parents

$\sigma^o$  A set of temporary mating pair

$\tau$  An offspring set of two individuals

$\mathcal{U}_k^{(t)}$  A set of family comprises parents, offspring, and fitness values

$t$  Generation step

## LIST OF PUBLICATIONS

- Lim, T. Y., Al-Betar, M. A., and Khader, A. T. (2015a). Adaptive pair bonds in genetic algorithm: An application to real-parameter optimization. *Applied Mathematics and Computation*, 252:503–519.
- Lim, T. Y., Al-Betar, M. A., and Khader, A. T. (2015b). Monogamous pair bonding in genetic algorithm. In *Evolutionary Computation (CEC), 2015 IEEE Congress*.
- Lim, T. Y. (2014). Structured population genetic algorithms: a literature survey. *Artificial Intelligence Review*, 41(3):385–399.
- Lim, T. Y. and Khader, A. T. (2013). Monogamous pairs genetic algorithm (mopga). *International Journal of Engineering and Advanced Technology (IJEAT)*, 2(5):143–149.

# IKATAN PASANGAN DALAM ALGORITMA GENETIK

## ABSTRAK

Tesis ini membentangkan siasatan komprehensif berasaskan konsep ikatan pasangan (pasangan monogami) yang akan dilaksanakan dalam fasa rekombinasi algoritma genetik (GA). GA merupakan teknik pencarian heuristik berdasarkan prinsip dan mekanisma pilihan semula jadi dan teori "survival of the fittest". Biasanya kromosom ibubapa akan dipilih pada setiap generasi bagi menghasilkan kromosom anak melalui operasi percantuman (crossover) dan mutasi. Proses ini diulangi sehingga syarat berhenti dipenuhi. Tetapi kadang-kala alam semula jadi mempamerkan pembentukan hubungan yang berkekalan antara pasangan mengawan. Dalam masyarakat manusia moden, sesetengah burung, ikan, tikus, dan cicak, ikatan pasangan merupakan aspek penting dalam tingkah laku sosial mereka. Mereka biasanya mengekalkan pasangan yang sama sepanjang hidup - monogami sosial. Oleh itu, tesis ini mengkaji kesesuaian aplikasi ikatan pasangan dalam GA. Dua kaedah GA baru akan dibentangkan: Kaedah pertama dikenali sebagai *Algoritma Genetik Ikatan Monogami (MopGA)*. Dalam MopGA, kromosom ibubapa akan berkekalan sehingga beberapa generasi. Kromosom anak yang terhasil akan mengalami persaingan bagi memastikan hanya yang terbaik dikekalkan ke generasi baru. Pemilihan kromosom ibubapa pula hanya akan berlangsung selepas ikatan pasangan berakhir. Curang (infidelity) yang dilaksanakan sekali-sekala boleh menjana kepelbagaian, merebak maklumat genetik di seluruh populasi dan mempercepatkan penumpuan. Kaedah kedua yang akan dibentangkan dikenali sebagai *MopGA Adaptif (AMopGA)*. Adaptasi parameter akan dilakuk-



an secara automatik sepanjang operasi evolusi. Ini akan meningkatkan prestasi. Lebih-lebih lagi, pengguna tidak perlu risau tentang kesesuaian parameter bagi GA lagi. Keputusan ujian menunjukkan bahawa MopGA dan AMopGA memang berkesan berbanding dengan kaedah semasa dari segi kualiti and masa pemprosesan.

# PAIR BONDS IN GENETIC ALGORITHM

## ABSTRACT

This work presents a comprehensive investigation on the concept of pair bonds (monogamous pairs) for the mating phase of genetic algorithms (GAs). GA is a heuristic search technique based on the principles and mechanisms of natural selection. Traditionally, parents are selected at every generation to reproduce offspring through crossover and mutation operations. The process reiterates until some termination conditions are met. However, nature sometimes exhibits the formation of enduring relationships between mating partners. In modern human society, some avian models, fish, rodents, and even lizards, pair bonds are integral aspects of their social behaviour. These species usually share the same mating partners throughout their lifetime - socially monogamous. Taking the cue from nature, this thesis studies the feasibility of pair bonds in GA. Consequently, two methodologies are proposed: Firstly, in the *Monogamous Pairs Genetic Algorithm (MopGA)*, parents are bonded and mated consistently over several predefined generations. Selection of new parents pairs will only take place at the end of pair bond tenure. Meanwhile, competition occurs between siblings to ensure only the best offspring are retained. Occasional infidelity generates variety, spreads genetic information across the population and speeds up convergence. Secondly, to improve the ease-of-use of MopGA, an *adaptive MopGA (AMopGA)* is introduced. Algorithm sensitive-parameters are tuned adaptively throughout the evolutionary process - further improving the performance. Rigorous performance investigation of both methodologies are carried out on different notable benchmark problems. The results reveal that the algorithms are very competitive to existing approaches in terms of solution quality as well as computational effort.

## CHAPTER 1

# INTRODUCTION

### 1.1 Preliminaries

Analogous to the evolutionary processes occurring in living nature, the principles of Genetic Algorithm (GA) were initially conceived by John Holland and his colleagues at the University of Michigan back in the early 1960's (Holland, 1975). GA, designed to simulate the natural evolutionary system, grounded on Charles Darwin's theory of survival of the fittest, is a heuristic and probabilistic search algorithm. GA is well accepted for its simplicity of operation and power of effect (Goldberg, 1989).

Unlike traditional optimization or search methods, GA examines a pool of probable solutions simultaneously, not just one. Moreover, GA often does not require priori knowledge of a problem to be solved. With only random choices, there is tendency to move towards likelihood of improved regions in the search space. The observed behaviour has made GA particularly successful in solving problems in wide variety of fields, including scheduling (Chung and Chan, 2012; Chen et al., 2012), engineering (Wang et al., 2012b), robotic (Köker, 2013) and transportation (Antony Arokia Durai Raj and Rajendran, 2012), to name but a few.

The artificial evolutionary process begins with the construction of candidate solutions (analogous to chromosomes of biological system). They usually take the form of binary strings, though real-valued and other forms are not by all rare. At each generation, a set of chromosomes is created through genetic operators like crossover and mutation. Each chromosome

is associated with a fitness value that is found by evaluating it with an objective or fitness function. The fitness function provides a measurement of quality used by the selection mechanism to guide the search towards regions where average fitness of the population improves. Crossover and mutation operations facilitate diversity while selection ensures quality.

While GA enjoys a wide range of credits as novel optimization, it too suffers from some challenges, including premature convergence (Kramer, 2010), low speed of convergence (Shi et al., 2013), random roaming, and low accuracy of solution (Zhang et al., 2009), especially when dealing with large-scale optimization problems.

In the search for GAs that do not succumb to premature convergence and that efficiently identify good solutions to computationally difficult problems, many organizations of the population and the operations on its members have been proposed. These include the island models (Cheng and Yang, 2010), cellular models (Villagra et al., 2014), hierarchical models (Yu and Ma, 2012), heterogeneous population-based models (Lahoz-Beltra et al., 2009; Akbari et al., 2010) and others. Generally, this work presents another such model, based on the idea of monogamy.

## **1.2 Background**

### **1.2.1 Exploration and Exploitation**

In many literatures, the terms *exploration* and *exploitation* often appear with association to the contribution of selection and genetic operators to search effectiveness. Nevertheless, many have taken for granted the intuitive meaning of the concepts and left the interpretation open for readers. Explicit investigation of these terms is rare and as put by Eiben and Schippers (1998)

"... *there is no generally accepted perception on exploration and exploitation in evolutionary computing.*". Of course, it is not mandatory to explain the working of GAs merely on exploration and exploitation, but we feel that it is appropriate for the clarification of our research intentions.

The common view about how GA works is closely associated with the exploration of search space by genetic operators, and exploitation by selection mechanism. Mutation and crossover are often seen as playing the roles of exploitative operator and explorative operator, respectively. Understandably, mutation is exploitative because it often resulted in small changes and preserves most of the existing genetic materials. Crossover performs recombination of genetic information from parents to form new offspring, thus suited the role of exploring new search region.

However, a different opinion exists. Eiben and Schippers (1998) provide an early discussion on evolutionary exploration and exploitation. Their works question the common belief that selection takes on the role as exploitative operator while crossover and mutation perform exploration. They find no universally accepted agreement on this topic. They also argue that mutation can be seen as explorative operator since it introduces new material in an unbiased manner to the population. Similarly, crossover takes the role of exploitative operator when one sees it as re-using existing old genetic materials for reproduction.

Nonetheless, most authors (Črepinšek et al., 2013), including this one, state the essence of striking a good balance between exploration and exploitation. The two terms are seen as opposing forces that need to be balanced in order to achieve good results. Following this convention, exploitation can be viewed as the good use of current information, while exploration is the discovery of new materials. The rest of the research will be grounded on this concept to improve GA performance.

### 1.2.2 Convergence Behaviour

In evolutionary search, convergence is related to the uniformity of its population. A population is said to have converged when its average population fitness is approximated to the best individuals.

Convergence can be attributed to the presence of stochastic errors such as the genetic drift. GAs have been observed to have converged even in the absence of selection pressure (e.g. using a constant fitness function) or with very low or zero mutation rate. The convergence to (local) optima can be understood as an instance of genetic drift. The process is however non-ergodic in nature. That is, the limit population strongly relies on the initial population (Schmitt, 2001). Consider, if by chance, a gene becomes predominant in the population, then there is a high chance that it will become more predominant in the coming generations. In a finite population, if the predominance is sustained over several successive generations, then the gene may spread to all individuals in the population. Since crossover does not introduce new gene values and mutation is generally low, each gene eventually becomes fixed over time.

As pointed out by Rudolph (1994), convergence to global optima is not an inherent property of canonical GA but rather a result of the algorithmic trick of keeping track of the best solution found over time.

Kramer (2010) states that "*Evolution rewards short term success...*". An evolutionary process is bound to get trapped in local optima. When the search converges towards local points, the phenomenon is known as *premature convergence*. It usually arises when the genes of some high rated individuals dominate the population too quickly; restricting genetic operators from producing any more better offspring than their parents (Fogel, 1994). As a result, the algorithm is halted from searching for other possibly better solutions.

Premature convergence behaviour is often attributed to the randomness nature of canonical GA in the mating process even though selection pressure has the tendency to push the algorithm towards better search regions.

Another cause of premature convergence is related to the decrease in population variance. Hence, the countermeasure proposed by many authors is to maintain higher diversity in the population pool. Diversity is the volumes of dissimilarities between individuals within a population (Artyushenko, 2009). By virtue maintaining higher population diversity would alleviate premature convergence, and obtain better results (Popovic and Murty, 1997).

Increasing the population size is an intuitive solution for maintaining higher population diversity. However, this leads to major increase in solution space and computational time, especially in large-scale problems. Thus, larger population is not a wise choice at time.

Increasing mutation rate may seem promising at first, but it too, alone is not an attractive solution. Mutation by itself is not enough to resume diversity for the entire population, especially when the population is already stuck at a sub-optimal stage. If mutation is too high, then the population becomes very unstable and will not be able to converge. Determining the choice of best mutation rate can be both tedious and very much problem-dependent.

Therefore, the likelihood 'ingredient' to reaching global optima is to strike a balance between exploration and exploitation in GA (Eiben and Smith, 2003).

### **1.2.2(a) Dilemma**

Increasing selective pressure speeds up search with the payoff of losing population diversity fast. On the other hand, maintaining population diversity helps to escape trapping in local optima, but offsets the effect of increasing selective pressure.

As noted, population diversity is closely related to GA convergence. The effect of loss of genetic diversity is twofold. On a negative side, it causes premature convergence. But, on a positive effect, it is required to drive the population towards promising region(s).

### 1.2.3 Parameter Setting

A common critic in GA applications is the difficulties in deciding the proper control parameters. Examples of such parameters include the population size, crossover probability and mutation probability.

Despite the long history of GAs, there appears to be no widely accepted guides for the use of right parameters. It is obviously time consuming to tune each parameter manually. Moreover, parameters appear to be problem-dependent. Different set of parameters may work well on some problems, but not for the others.

The number of published papers on various parameter setting methods has increased dramatically over the years (Eiben and Smit, 2011; Kramer, 2010). This shows how important parameter settings is in GA. While parameter values largely influence an algorithm's performance, there exists no universally agreeable values for all problems; posing huge challenge to the GA communities. Some believes that parameter tuning is a special case of Evolutionary Algorithms (EAs) design and thus, an optimization problem in itself (Kramer, 2010).

Following the parameter setting taxonomy offered by Eiben et al. (2007), parameters can be chosen before or during GA runs. The former, also known as *parameter tuning* is often time consuming, less accurate and problem-specific. The allocated value remains at constant throughout evolution. In the latter case (*parameter control*), an initial value undergoes changes



during GA runs. Included in this category of parameter control are the deterministic, adaptive, and self-adaptive approaches. Due to flexibility and less time consuming, the parameter control approaches are receiving much attention in the late (Elsayed et al., 2011; Rajakumar and George, 2013; Osaba et al., 2014).

### 1.3 Pair Bond

In nature, *monogamy* implies a social organization in which the male and female organisms breed exclusively with each other, even though extrapair copulations or infidelity are commonly found in monogamous species (Young and Wang, 2004; Hatchwell et al., 2000). In monogamous society, mating pairs display selective (though not exclusive) affiliation, copulation, nest sharing and biparental care of offspring. Pair bond is a form of intense social attachments.

The duration of pair bond and age have been suggested to be the two main determinants for reproductive success in long-term monogamous pair bonding. The hypothesis has been proven true by Griggio and Hoi (2011) in their studies of the long-term pair bond duration in socially monogamous bearded reedling, *Panrus biarmicus*<sup>1</sup>. They find that the longer the duration of pair bonds, the more successful they are in breeding attempts, hatching and fledging.

Recent discoveries in molecular, cellular and neurobiology further enable insightful understanding of pair bonding. Pair bond has a biological architecture with molecular and neural mechanisms (Young and Wang, 2004; Young et al., 2005). Their research presented a neurological model of pair-bond formation based on studies in monogamous rodents. They observed that concurrent activation of neuropeptide<sup>2</sup> and dopamine<sup>3</sup> receptors in the reward centers of the brain during mating causes a conditioned partner preference - pair bonds.

---

<sup>1</sup>The bearded reedling is a small orange-brown bird with a long tail, also known as bearded tit or bearded parrotbill.

<sup>2</sup>The ability to recognize individual is contributed by neuropeptides oxytocin and vasopressin.

<sup>3</sup>Mesolimbic dopamine is involved in reinforcement and reward learning.

Also utilizing neural mechanisms, Resendez and Aragona (2013) suggest that monogamous pair bonds are formed and maintained by a balance between mu- and kappa-opioid receptor<sup>4</sup> activation. Endogenous opioid transmission influences the ability of an animal to see positive stimuli as reward, whereas harmful ones as aversion. Hence, when bond is seen as rewarding, they will be maintained properly in the relationship (Resendez and Aragona, 2013).

To better illustrate the process of pair bonding, we refer to Figure 1.1 - a simplified version of striatopallidal regulation of pair bonding (Resendez and Aragona, 2013). Prior to bond, two individuals must meet - *Meeting*. Then pair bond is formed through the presence of neural mechanisms where both parties acknowledge the rewarding neural events - *Formation*. In the next stage, the bond must be maintained - *Maintenance*. That is, the individuals involve show preference to only one mate and reject other potential mates as mediated by the neural mechanism. At the same time, the absence of a partner is likely to induce stress and negative effect on the individual and further promote pair bond maintenance - *Isolation* (Bosch et al., 2008).



Figure 1.1: Striatopallidal regulation of pair bonding - a simplified version (Resendez and Aragona, 2013)

---

<sup>4</sup>Opioid receptors are a group of G protein-coupled receptors with opioids as ligands. They are widely distributed in the brain, spinal cord and digestive tract

## 1.4 Inspiration and Motivation

Fascinated by animal monogamy, this thesis draws inspiration from the fact that apart from modern human society, 90% of bird species are monogamic and usually exhibit biparental care of young (Schuiling, 2003). Some species of fish (Oldfield and Hofmann, 2011), rodents (Young et al., 2011), and lizards (Leu et al., 2011) also display enduring relationships between mating partners.

Social monogamy has evolved many times and its occurrence has left plethora of studies in evolutionary biology (Hatchwell et al., 2000; Schuiling, 2003; Hosken et al., 2009; Weiß et al., 2010). Among the widely studied socially monogamous birds, extrapair or infidelity has been commonly observed (Hatchwell et al., 2000).

Even though the understanding of monogamy and infidelity is still at its infancy, numerous hypotheses have been attempted to explain the presence of infidelity in monogamous society. These include the good genes and future partnerships hypotheses (Ramsay et al., 2000). Briefly, the former predicts mating success with bias against maladaptive alleles, while the latter favours potential mates.

With pair bonding, the highly acclaimed benefits, especially in humans include but not limited to longer lifespan compare to their unpaired counterparts, and ensuring physical and psychological well-being of offspring (Schuiling, 2003; Hosken et al., 2009).

To our best knowledge, GAs do not conform to monogamous structure. Nonetheless, if social monogamy survives the evolution of nature, we believe such structure may have strength in itself.

## 1.5 Research Problem

GAs are an important emergent computing methodology and a hot topic for research due to their versatility in solving difficult problems, especially in the optimization fields. The number of published papers on GAs has increased by leaps and bounds over the last few decades.

However, in the midst of glory, there exist the two most frequently cited criticisms of GAs: 1) GAs can easily get stuck at local optima (premature convergence), and 2) they often need careful fine-tuning of parameters (Eiben and Smith, 2003).

In order to obtain good results, striking a balance between exploration and exploitation is essential (Črepinšek et al., 2013). Generally, achieving proper balance between exploration and exploitation has been done through careful control-parameter setting. For instance, a high crossover and mutation rates will lead to more exploration of the search space, but at the expense of exploiting existing solutions and missing the good ones. Conversely, low crossover and mutation rates result in many unexplored search regions.

To make matter worse, there exists interactions among algorithm parameters (De Landgraaf et al., 2007), and hence optimizing them cannot be performed independently. Ideally, all combinations of setting should be experimented. However, in the absence of optimization experience and prior knowledge, it is almost inconceivable to identify a proper setting for GAs without resorting to trial-and-error procedure. The trial-and-error scheme usually requires high computational costs.

Moreover, parameter setting is often problem-dependent, i.e. the control parameter setting that leads to best result for one problem may not produce the best result for another problem. In view of exploration/exploitation, this means that every problem differs in the amount of exploration and exploitation required. For example, multimodal function optimization requires more exploration compared to unimodal function optimization.

To achieve best performance, it is also necessary to apply different strategies coupled with different parameter settings at different stages of the evolution (Qin et al., 2009). In other words, different amounts of exploration and exploitation are required at different evolutionary stages.

This thesis presents a method that incorporates and successfully combines two mechanisms (pair bond and infidelity) for guarding against premature convergence with minimum effort in parameter tuning. The algorithm is inspired by social monogamous mating system in nature.

Moreover, conventional GAs do not conform to monogamous structure - selection of parents for mating takes place at every generation without enduring partnership between parents. Conversely in monogamous system, pair bond is retained over successive generations. Exploration and exploitation of the search space can be effectively controlled by the degree of infidelity and pair bond, respectively.

## **1.6 Research Objectives**

The overall research aim is to incorporate the pair bond and infidelity concepts inspired by social monogamy into GA in order to improve performance with least effort in parameter tuning. The main objectives of this thesis are:

1. To incorporate pair bond and infidelity roles found in monogamy mating system into GA in order to improve performance. Performance can be improved by:
  - (a) maintaining population diversity
  - (b) increasing exploitation of search space through pair bonding
2. To improve ease-of-use by reducing user effort required in appropriate control parameter-setting.

## 1.7 Research Contributions

The key contributions of this thesis to the literature are :

1. Introducing a new variant of GA which employs the concept of pair bond and infidelity. It is henceforth called the Monogamous Pairs Genetic Algorithm (**MopGA**).
  - Two new parameters are introduced, namely 1) pair bond tenure, and 2) infidelity rate. The former controls the duration of pair bond and is associated with the role of exploitation. Meanwhile, the latter determines the frequency of exploring other search regions. Together, the two operations control the exploration and exploitation of the search space.
  - In monogamous state, the number of parents selections is reduced. The longer the pair bond tenure, the lower the number of parents selection because bonded parents remain attached for reproduction without the need for re-selection. Consequently, reduction in computational effort is achieved.
  - Enforce the use of crossover operation at every generation. As a result, one conventional parameter tuning (i.e. crossover rate) is reduced.
2. Introducing an enhanced version of the MopGA, known as Adaptive Monogamous Pairs Genetic Algorithm (**AMopGA**).

- Remove parameter tuning burden from GA practitioners by:
  - dynamically adapting mutation rate using deterministic rules.
  - dynamically adapting pair bond tenure and infidelity rate using adaptive parameter control approach with feedback from the search as input to the logistic functions.
- Further improvement in performance accuracy is achieved via optional use of:
  - adaptive selection of crossover operators
  - secondary mutation operation

## **1.8 Scope of the Research**

The thesis centres around the basic structure or framework of genetic algorithm with focus on bonds between parents. This directly affects the frequency of parents selection. However, in order to study the effects of bonds, generic operators are used unless on special occasions deem necessary.

The research is limited to optimization in static environment with single-objective functions. Furthermore, the choices of test functions are made based on common use in empirical literature and challenging in a diverse area of applications, though limited to 0/1 knapsack problems and numerical optimization problems.

## **1.9 Organization of Thesis**

The thesis is organized as follows: Chapter 2 presents a quick and superficial overview of a standard genetic algorithm, followed by literature survey on the various developments in GAs, in particular on effort related to this work.

In Chapter 3 two methodologies are proposed and described in details. They are 1) the basic monogamous pairs genetic algorithm (MopGA), followed by 2) an enhanced version, namely the AMopGA, which incorporates adaptive parameters and additional performance improvement features.

Detailed description of the test problems and basic experimental setup can be found in Chapter 4. Chapter 5 introduces the reader to the properties of MopGA using artificial test functions. The effect of changing various parameters are also tested, further illustrating the search and convergence behaviour of the proposed methodology. Preliminary tests on AMopGA are also being carried out using similar test problems in Chapter 6.

Thereafter, applications on widely known NP-hard problems in different domains will be employed to exhibit the robustness, flexibility and potential of the proposed methodologies. Two case studies are being applied in Chapters 7 and 8 for the domain of 0/1 knapsack and real-parameter optimization problems, each separately.

Finally, in Chapter 9 a summary of claimed results and future possibilities are provided. It is hoped that the reader will be challenged to prove or disprove the claims made throughout this thesis, and extend the research in new directions.



## CHAPTER 2

# LITERATURE REVIEW

### 2.1 Preliminaries

Evolution by natural selection has been one of the most studied themes of modern science. Darwin's theory of evolution (Darwin and Beer, 1951) depicts biological system as the product of ongoing process of natural selection. Natural selection favours individuals that are well adapted to the environmental conditions. This phenomenon is also called *survival of the fittest*.

Likewise, there is a form of evolution, known as Evolutionary Algorithm (EA), which takes place in a computer system. EA is a general term used to represent a group of related but dissimilar methodologies that all exploit ideas from natural selection and evolution. Genetic Algorithm (GA) (Holland, 1975), Evolution Strategies (ES) (Rechenberg, 1994; Schwefel, 1993), Evolutionary Programming (EP) (Fogel et al., 1966; Fogel, 2006) and Genetic Programming (GP) (Koza, 1992) are the four prominent approaches in EA.

Briefly, these approaches share a few common characteristics (Yu and Gen, 2010): population-based, fitness-oriented and variation-driven. Firstly, EAs maintain a group of potential solutions, known as *population*, which enables optimization to progress in parallel. Next, every solution in a population is referred to as an *individual*. Each individual is associated to some form of gene representation (encoding) and performance evaluation, called its *fitness* value. Fitter individuals are usually preferred by EA. Finally, individuals undergo a number of variation operations (such as mutation, recombination, crossover), which is fundamental to searching the solution space, until some termination conditions are satisfied.

Initially conceived by Holland (1975), GA belongs to a class of stochastic search methods inspired by natural selection. GA iteratively operates on a population of individuals, each of which represents a potential solution to a given optimization problem. In each iteration step, known as *generation*, individuals in the current population are evaluated and assigned a fitness value. A complete generation cycle comprises three basic genetic operators, namely selection, crossover and mutation. Detailed description of a Standard GA (SGA) can be found in Section 2.2. Theoretical background to better understanding the behaviour of SGA follows in Section 2.3.

GAs' outstanding performance has been validated through various optimization problems. A plethora of algorithm variants have also emerged leading to increase efficiency and effectiveness. These enhancements redefine: 1) the way genes are encoded in chromosome (Király and Abonyi, 2011; Abdul-Rahman et al., 2011), 2) operation of genetic operators (Elsayed et al., 2011; Mudaliar and Modi, 2013), 3) evaluation of fitness function (Kim and Cho, 2001), 4) interaction of individuals in the population (Gong and Fukunaga, 2011; Villagra et al., 2014), and 5) hybridization (Meeran and Morshed, 2012). General overview of these variants will be discussed in Section 2.4.

Furthermore considering the theme of this thesis, special focus will be made on two main aspects: GAs with multiple offspring (Section 2.4.1) and parameter setting in GA (Section 2.4.2). Although far from exhaustive, it is with great hope that one will be able to better appreciate the significant of the proposed methodologies after following through some previous work. Finally, we end this chapter with Section 2.5.

## 2.2 An Overview of the Standard Genetic Algorithm

This section presents a brief outline of a standard genetic algorithm (SGA). A more detailed exposition can be found elsewhere (Eiben and Smith, 2003; Michalewicz, 1998). In what follows, a concise description of the algorithm is provided in Algorithm 2.1, while postponing most details to the ensuing subsections.

---

### Algorithm 2.1 Pseudo code of a typical SGA

---

**Require:** Fitness function ( $f$ ), GA parameters: population size ( $N$ ), maximum iteration ( $G$ ), crossover rate ( $P_c$ ), mutation rate ( $P_m$ )

**Ensure:** Best solution

```

1:  $t = 0$ 
2:  $init(\mathfrak{P}_p^{(0)} = \{\rho_k^{(0)}, \quad k = 1, 2, \dots, N\})$ 
3: while Termination criteria not met do
4:   for  $j = 0$  to  $\frac{N}{2}$  do
5:      $\sigma_j = selection(\mathfrak{P}_p^{(t)})$ 
6:      $\tau_j = crossover(\sigma_j, P_c)$ 
7:      $\tau_j = mutation(\tau_j, P_m)$ 
8:      $f_j = evaluation(\tau_j)$ 
9:      $\mathfrak{P}_o^{(t)} = \mathfrak{P}_p^{(t)} \cup \tau_j$ 
10:   end for
11:    $\mathfrak{P}_p^{(t+1)} = \mathfrak{P}_o^{(t)}$ 
12:    $t = t + 1$ 
13: end while

```

---

At generation  $t = 0$ , the parental population  $\mathfrak{P}_p^{(0)} = (\rho_1, \rho_2, \dots, \rho_N)$  is initialized randomly and their fitness values calculated in line 2. After initialization, the repeat-until-loop is entered (lines 3-13). From the parental population  $\mathfrak{P}_p^{(t)}$  at generation  $t$ , a new offspring population  $\mathfrak{P}_o^{(t)}$  is produced via  $N/2$  number of runs through lines 4-10. Each cycle generates two offspring: Firstly, in selection step (line 5), two parents  $\sigma$  are chosen from  $\mathfrak{P}_p^{(t)}$ . The selection process is dependent upon the parental fitness values  $f$ . This is followed by recombination operation (crossover) in line 6, where the two parents swap segments (at selective loci), and produce two

new offspring  $\tau$ . The probability of crossover is controlled by a crossover rate  $P_c$  parameter. Then mutation is applied at mutation rate  $P_m$  on the resulting offspring in line 7, and their fitness are evaluated in line 8. Thereafter, the offspring are added into the intermediate offspring pool  $\mathfrak{P}_o^{(t)}$  in line 9.

In line 11, the offspring population  $\mathfrak{P}_o^{(t)}$  replaces the old parent population  $\mathfrak{P}_p^{(t)}$  to become the new parent population  $\mathfrak{P}_p^{(t+1)}$  of the next generation. Finally, the termination condition is verified. The evolutionary process ceases when the conditions are satisfied. In the followings, a thorough discussion of each SGA components is provided.

### 2.2.1 Encoding

In any real world applications, the search space is defined by a set of objects with different parameters each. The goal of optimization problem working on these parameters is to optimize them.

Likewise in GA, a chromosome or an individual is a set of parameters that define a candidate solution to the optimization problem. The smallest unit in a chromosome is known as *gene*. A gene consists of two properties, namely its value (also known as *allele*) and its location (called *locus*) (Yu and Gen, 2010). An allele is usually represented by a single symbol. For instance, in a binary representation, an allele takes on the value of 0 or 1.

GAs work alternatively between the coding space and the solution space. Analogous to population genetics (Nagylaki, 2012), coding space is also known as the *genotypic* space, while solution space represents the *phenotypic* space (see Figure 2.1). Genotype depicts all the information stored in a chromosome. Meanwhile, phenotype is an overall outward description

of an individual. The construction (mapping) of phenotype from genotypic information forms an essential part of GAs. While genetic operators such as crossover and mutation operate on the level of the genotype, evaluation of individual is performed on the level of the phenotype (Rothlauf, 2006).

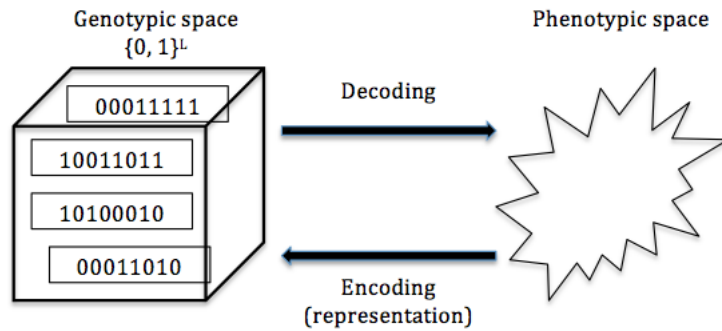


Figure 2.1: An example of mapping between binary genotypic space and phenotypic space.  $L$  represents the chromosome length.

Representation or encoding assigns phenotypes to the corresponding genotypes. It has been shown that the performance of GA may be affected by the underlying encoding scheme used (Abdul-Rahman et al., 2011). There exists a variety of encoding schemes within the literature, with binary and real (floating point) encodings being the most popular.

Binary representation is one of the earliest encoding scheme found in GAs (Whitley, 1991; Mitchell et al., 1992; Watson and Pollack, 2000). The search space is defined by  $\{0, 1\}^L$ , where  $L$  is the length of a binary vector  $\mathbf{x} = (x_1, x_2, \dots, x_L) \in \{0, 1\}^L$  (Goldberg, 1989). This makes the search space size  $2^L$ . The genotype-phenotype mapping is dependent upon the optimization problem undertaken by GA. In general, a majority of the other representations are convertible to binary scheme, with the exception of permutation problem.

On the other hand, in real-valued encoding each gene takes on the form of floating-point numbers (Picek et al., 2013; Elsayed et al., 2013a). The search space is presented by  $\mathbb{R}^L$ , where  $L$  represents the real-valued chromosome length. In this case, the genotype space is identical to the phenotype. The real-valued encoding is conceptually closest to real design space.

In combinatorial optimization problems, the commonly employed encoding scheme is based on integer or literal permutation (Liu and Zeng, 2009; Chang et al., 2010; Noraini and Geraghty, 2011). The size of the search space is  $\Omega^L$ , where  $\Omega$  is the  $\Omega$ -ary alphabets used for representation, e.g. ternary, quaternary, quinary, etc.(Rothlauf, 2006).

### **2.2.2 Fitness Function**

Each individual in a GA population comprises two properties, namely its encoding structure (recall Section 2.2.1) and its quality. These two components are usually problem dependent (Whitley, 1994).

The quality of an individual is determined by a *fitness value*. That is, fitness value is used to evaluate how much an individual fits the problem undertaken. The function that is used to calculate this fitness value is known as *fitness function*. Meanwhile, *evaluation* refers to the process of obtaining a fitness value from an individual (Yu and Gen, 2010).

### **2.2.3 Selection**

According to Blickle and Thiele (1995), the central tenet of selection operator is to select a set of individuals in a way that improves the average quality of the population. To achieve this goal, the selection operator gives higher quality individuals a higher chance of being used for generating the next generation population. The quality of an individual, in turn is pertained to a fitness function.

There exists many different selection schemes for GAs, each with varying characteristics. The most prominent are proportionate selection (Holland, 1975) and tournament selection (Goldberg, 1989). In the former, individuals are selected for reproduction in proportion to their fitness relative to the other individuals in the population (also referred to as roulette wheel selection). The probability of an individual  $x_i$  being selected is calculated as:

$$\frac{f(x_i)}{\sum_{j=1}^N f(x_j)}$$

where  $N$  symbolizes the population size. The higher the individual's fitness, the higher its chances of being selected for reproduction.

On the other hand, in tournament selection,  $s$  number of randomly chosen individuals compete for a place in the mating pool, with the high fitness individuals winning the tournament. The value  $s$  is referred to as the *tournament size*. It controls the selection pressure <sup>5</sup>.

#### 2.2.4 Crossover

Crossover is one of the primary variation operators in GAs. It is a reproduction technique that usually employs two parents to produce two offspring by exchanging portions of the parents' internal representation. These offspring carry partial information from both parents. The frequency of crossover is determined by a crossover rate  $P_c$  that is recommended between 0.5 and 0.9 (Garai and Chaudhuri, 2002).

---

<sup>5</sup>The selection pressure is the degree of favouring better individuals - the higher the selection pressure, the more better fitness individuals are selected (Miller and Goldberg, 1995)

Various crossover procedures exist in GA literature. Perhaps, the simplest form is the single-point (also known as 1-point) crossover. Here, a point of exchange is set at any random locations in the two individuals' genomes. The two individuals then exchange genomes at the 'breaking point'. Whereas in 2-point crossover, as the name implied, two crossover locations are randomly selected and the genomes between them are exchanged. The two types of crossover operators described are visualized in Figure 2.2.

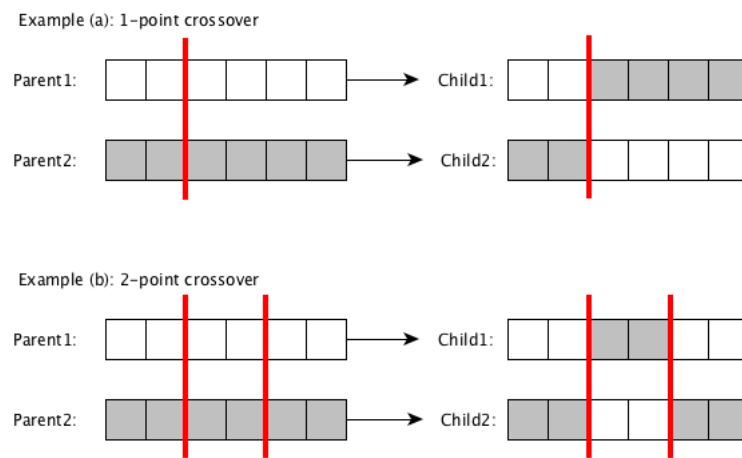


Figure 2.2: Examples of 1-point and 2-point crossover operators

In general, there can be  $n$  number of crossover points selected at random. The  $n$ -point crossover (Eshelman et al., 1989) and uniform crossover (Syswerda, 1989) are the other two commonly used crossover operators in binary encoded GAs. In a uniform crossover, alleles at each parents loci will be swapped with probability 0.5.

De Jong and Spears (1991) show the relationship between crossover and population size. They conclude that with 2-point crossover, performance improves with large population size. On the other hand, uniform crossover gives better results for small population size. Eshelman (1991) also make similar arguments, enlisting the advantages of disruptive operators <sup>6</sup>.

<sup>6</sup>An operator is disruptive if it destroys schemata that are common to both parents (Spears et al., 1992).



As Eshelman et al. (1989) analyse crossover operators in terms of "positional" and "distributional" biases, their empirical studies suggest that no  $n$ -point, shuffle, or uniform crossover is universally better than the others.

More recently, Picek et al. (2013) compares 16 real-coded crossover operators on 24 benchmark functions to show that there are significant differences in efficiency of different crossover operators as affected by the distinctive properties of each fitness function. The researchers conclude that combining crossover operators yields the best results.

#### **2.2.4(a) Roles and Characteristics of Crossover in GAs**

Generally, crossover serves two roles, namely disruption and construction. Additionally, it exhibits the characteristics of exploratory power. The following sub-sections discuss each element in further details.

**Disruption role.** Following the formal schema analysis of GAs by Holland (1975), investigations on the disruption rate of crossover have gained momentum (De Jong and Spears, 1992). They study the likelihood that a sampling of the  $k$ -th hyperplane  $H_k$  will be disrupted by a particular crossover. For instance, a string 11110 with schema  $H = 1**10$  has an order schema  $o(H) = 3$  such that  $k = 3$  and  $H_3$  represents the 3rd order hyperplane (more description about schema can be found in Section 2.3). In other words, a crossover operator that produces offspring that are not members of the same subspace as their parents is *more disruptive* than the one that produces offspring that are in the same subspace as (or more similar to) their parents.

Even though disruption is often associated with exploration, that does not necessarily mean *useful* exploration. Rather, it implies that a hyperplane sample has been modified and it is now different from the original.

According to Syswerda (1989), the uniform crossover is more disruptive than 1- and 2-point crossovers. Whereas, the 2-point crossover is best at minimizing disruption. De Jong and Spears (1992) further show that the disruption rate of uniform crossover is independent of hyperplane defining length. The disruption ability of the crossover operator depends on:

1. position of crossover point(s)
2. alleles common to both parents of the  $k$ -th hyperplane

**Construction role.** In contrast to disruption, when an operator can preserve the similarity exhibited by parents, it takes on the constructive role. This idea stems from Syswerda (1989)'s hypothesis that a more positive theory of crossover is constructive in nature (rather than disruptive). De Jong and Spears (1992) calls this the recombination potential.

In construction theory, the probability that an offspring within a hyperplane can be constructed from existing parents are calculated, instead. De Jong and Spears (1992) shows that highly disruptive crossover operators are also highly constructive. For instance, the uniform crossover (which is also highly disruptive) has higher recombination potential compared to the 1- and 2-point crossover (De Jong and Spears, 1992; Syswerda, 1989).

In a diverse population, especially, crossover displays considerably higher constructive level than mutation (Spears et al., 1992). The recombination ability of crossover is a specialized form of *survival* (Syswerda, 1989); ensuring any lower order schema will appear on the same string and leading to the construction of higher order schema.