
EVOLUTIONARY ALGORITHMS FOR FINANCIAL TRADING

by

DOME LOHPETCH



Submitted for the Degree of
Doctor of Philosophy
on completion of research in the
Department of Computing Sciences
School of Mathematical and Computing Sciences
Heriot-Watt University
November 2011

This copy of the thesis has been supplied on the condition that anyone who consults it is understood to recognise that the copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author or the university (as may be appropriate)

Declaration

I hereby declare that the work presented in this thesis was carried out by myself at Heriot-Watt University, except where due acknowledgement is made, and not been submitted for any other degree.

Dome Lohpetch (Candidate)

Professor David Wolfe Corne (Supervisor)

Date

Acknowledgements

First and foremost, I would like to thank my parents and my family who are always on my side. They give me incomparable love, unconditional support and encouragement to pursue my interests and my dream. They always believe in me even if sometimes I doubted myself.

I would like to express my sincere gratitude and deep appreciation to my supervisor, Professor David Wolfe Corne, for his continuous support in my PhD. He taught me how to do research and how to write an academic paper. He has read all my work patiently and given significant contributions to my research. He always gives constructive comments, excellent advice, as well as great supervision. Without his encouragement and constant guidance, I could not have finished this thesis.

An honourable mention goes to my great friends; Walailuck, Chamnan, Kannika, Pitak, Wasit, Warunee as well as Komsan. They make my life in Scotland as warm as staying with my family. The years with them are so meaningful. In addition, I am pleased to thank my office mates Guleng and Muhunnad for sharing the research environment and wonderful activities.

Last but not least, I would like to thank the Department of Mathematics, King Mongkut's University of Technology North Bangkok for allowing me to come to do a PhD in the UK, the Royal Thai Government for funding and the staff at the OEA, Royal Thai embassy (London) for assistance.

Abstract

Genetic programming (GP) is increasingly popular as a research tool for applications in finance and economics. One thread in this area is the use of GP to discover effective technical trading rules. In a seminal article, Allen & Karjalainen (1999) used GP to find rules that were profitable, but were nevertheless outperformed by the simple “buy and hold” trading strategy. Many succeeding attempts have reported similar findings. This represents a clear example of a significant open issue in the field of GP, namely, *generalization in GP* [78]. The issue of generalisation is that GP solutions may not be general enough, resulting in poor performance on unseen data. There are a small handful of cases in which such work has managed to find rules that outperform buy-and-hold, but these have tended to be difficult to replicate. Among previous studies, work by Becker & Seshadri (2003) was the most promising one, which showed outperformance of buy-and-hold. In turn, Becker & Seshadri’s work had made several modifications to Allen & Karjalainen’s work, including the adoption of monthly rather than daily trading. This thesis provides a replicable account of Becker & Seshadri’s study, and also shows how further modifications enabled fairly reliable outperformance of buy-and-hold, including the use of a train/test/validate methodology [41] to evolve trading rules with good properties of generalization, and the use of a dynamic form of GP [109] to improve the performance of the algorithm in dynamic environments like financial markets. In addition, we investigate and compare each of daily, weekly and monthly trading; we find that outperformance of buy-and-hold can be achieved even for daily trading, but as we move from monthly to daily trading the performance of evolved rules becomes increasingly dependent on prevailing market conditions. This has clarified that robust outperformance of B&H depends on, mainly, the adoption of a relatively infrequent trading strategy (e.g. monthly), as well as a range of factors that amount to sound engineering of the GP grammar and the validation strategy. Moreover,

we also add a comprehensive study of multiobjective approaches to this investigation with assumption from that, and find that multiobjective strategies provide even more robustness in outperforming B&H, even in the context of more frequent (e.g. weekly) trading decisions. Last, inspired by a number of beneficial aspects of grammatical evolution (GE) and reports on the successful performance of various kinds of its applications, we introduce new approach for (GE) with a new suite of operators resulting in an improvement on GE search compared with standard GE. An empirical test of this new GE approach on various kind of test problems, including financial trading, is provided in this thesis as well.

Contents

Acknowledgements	iii
Abstract	iv
List of Tables	xii
List of Figures	xix
1. Introduction	1
1.1 Motivation	5
1.2 Technical Analysis versus Random Walk & Market Efficiency Hypotheses	7
1.3 Contribution.....	9
1.3.1 Contributions to Computer Science	10
1.3.2 Contributions to Finance.....	11
1.4 Thesis Scope and Limits.....	12
1.4.1 Scope of the thesis in terms of Fitness Functions:.....	12
1.4.2 Scope of thesis in terms of Technical Indicators:	13
1.4.3 Scope of Methodologies and Techniques to Provide Generalization in Dynamic Environments:	13
1.4.4 Scope of thesis in terms of Parameters and Configuration:.....	13
1.4.5 Scope of thesis in terms of Data Sets:	14
1.5 Thesis Structure	14
1.6 Publications	16
2. Background and Literature Review	17
2.1 Evolutionary Computing (EC)	18

2.1.1 General Scheme of an Evolutionary Algorithm	19
2.1.2 Introduction to the Genetic Algorithms (GA)	21
2.1.3 Introduction to Genetic Programming (GP)	23
2.2 Multi-objective Evolutionary Algorithms (MOEA).....	27
2.2.1 An Overview of Multi-objective Optimization Problems (MOOP)	27
2.2.2 Principles of Multi-objective Optimization	30
2.2.3 EA Approaches to Multi-objective Optimization.....	35
2.2.4 NSGA-II	37
2.3 Technical Analysis	41
2.3.1 Identifying Trend Reversals	42
2.3.2 Technical Indicators.....	45
2.4 EA and MOEA in Computational Finance.....	51
2.4.1 Related works of Evolutionary Algorithms with Single-Objective Approach in Financial Market.....	51
2.4.2 Related works of Evolutionary Algorithms with Multi-Objective Approach in Financial Market.....	56
2.5 Discussion and Conclusion	57
3. Single Objective Algorithms for Evolved Technical Trading Rules	59
3.1 Introduction	59
3.2 Evolving Robust Trading Rules	62
3.2.1 Overview.....	62
3.2.2 Function and Terminal Sets Overview	64
3.2.3 The Fitness Function.....	67

3.2.4 Operators and Initialization	72
3.3 Experiments.....	77
3.3.1 GP Parameters	77
3.3.2 Data splits	78
3.4 Results	84
3.4.1 Monthly Trading.....	85
3.4.2 Weekly Trading	90
3.4.3 Daily Trading.....	92
3.4.4 Performance Comparison	96
3.5 Rule Analysis	102
3.5.1 Rule Analysis for Monthly Trading.....	103
3.5.2 Rule Analysis for Weekly Trading.....	106
3.5.3 Rule Analysis for Daily Trading.....	110
3.6 Discussion and Conclusions.....	114
4. Multiobjective Algorithms for Evolved Technical Trading Rules	119
4.1 Introduction	119
4.2 GP Approach and Multi-objective Characterization.....	120
4.2.1 GP Approach	120
4.2.2 Single and Multi-objective Approaches	121
4.3 Experiments.....	127
4.4 Results	128
4.4.1 Profit Driven Approach Results.....	128

4.4.2 Risk-adjusted Approach Results	143
4.5 Rule Analysis	156
4.5.1 Rule Analysis for Monthly Trading	156
4.5.2 Rule Analysis for Weekly Trading	159
4.5.3 Rule Analysis for Daily Trading	164
4.6 Concluding Discussion	168
5. A New Approach for Grammatical Evolution	170
5.1 Introduction	170
5.2 Test Problems for Comparing GE and GP	172
5.2.1 Symbolic Regression (SR)	172
5.2.2 Symbolic Integration (SI)	174
5.2.3 Santa Fe Ant Trail (ANT)	175
5.3 GE Methodology	177
5.3.1 Overview	177
5.3.2 GE Fitness Calculation and Mapping Process	178
5.3.3 GE Initialization	183
5.4 Repair Strategies	183
5.4.1 Valid Map Method1 (MAP01)	184
5.4.2 Valid Map Method2 (MAP02)	184
5.4.3 Valid Map Method3 (MAP03)	185
5.5 Performance Comparison of Valid Map Methods	187
5.6 Standard GE Genetic Operators	189

5.6.1 One-point Crossover Operator	189
5.6.2 Two-point Crossover Operator	189
5.6.3 Flip Mutation Operator	190
5.7 Typed Genetic Operators.....	190
5.7.1 Two-point Typed Crossover Operator	197
5.7.2 Typed Mutation Operator	202
5.8 Results of GE on various test problems	213
5.9 Results of GE on Real Word Problem – Financial Trading.....	217
5.10 Concluding Discussion.....	221
6. Conclusion	224
6.1 Summary	224
6.2 Contributions.....	225
1.6.1 Contributions to Computer Science.....	225
1.6.2 Contributions to Finance.....	228
6.3 Future Work	230
Appendix A	232
A.1 Samples of Technical Trading Rules for Single-objective Approach	232
A.1.1 Monthly Trading	232
A.1.2 Weekly Trading	233
A.1.3 Daily Trading.....	234
A.2 Samples of Technical Trading Rules for Multi-objective Approach	235
A.2.1 Monthly Trading	235

A.2.2 Weekly Trading	236
A.2.3 Daily Trading.....	237
Bibliography.....	238

List of Tables

Table 2-1: Summary notes of the well-known multi-objective algorithms	37
Table 2-2: Calculations of 5-day moving averages on three days.	47
Table 2-3: Summary on key factors of related works in single-objective approach for both risk unadjusted and risk adjusted methods with (a) blue lines of works on global market indices, (b) yellow lines of works on foreign exchange market and (c) white lines of works on stocks of individual companies.	55
Table 3-1: The function set and terminal set for multi-objective strongly typed GP.	65
Table 3-2: GP Parameters of single-objective approach.	77
Table 3-3: Details for each monthly data split – T: Training Period, V: Validation Period, E: Evaluation Period.	79
Table 3-4: Details for each weekly data split – T: Training Period, V: Validation Period, E: Evaluation Period.	82
Table 3-5: Details for each daily data split – T: Training Period, V: Validation Period, E: Evaluation Period.	83
Table 3-6: Summary of results for M01-M04 data splits of 7 different objectives: MR, 4 PC periods and 2 risk-adjusted approaches.	87
Table 3-7: Summary of results for N_M01-N_M04 data splits of 7 different objectives: MR, 4 PC periods and 2 risk-adjusted approaches.	88
Table 3-8: Summary notes comparing different setups.	90
Table 3-9: Summary of results for W01-W04 data splits of 5 different objectives: MR, 2 PC periods and 2 risk-adjusted approaches.	92
Table 3-10: Summary of results for N_W01-N_W04 data splits of 5 different objectives: MR, 2 PC periods and 2 risk-adjusted approaches.	92
Table 3-11: Summary of results for D01-D04 data splits of 5 different objectives: MR, 2 PC periods and 2 risk-adjusted approaches.	94

Table 3-12: Summary of results for N_D01-N_D04 data splits of 5 different objectives: MR, 2 PC periods and 2 risk-adjusted approaches.	95
Table 3-13: Summary note on performance for each method with regime 2 for monthly, weekly and daily data.	96
Table 3-14: Paired one-tailed T-Test on the hypothesis that the best method is MR for monthly trading, displaying mean and p-value for each configuration, spotting p-value > 0.10 with gray highlight.	97
Table 3-15: Paired one-tailed T-Test on the hypothesis that for weekly data splits of W01, W04, N_W01 and N_W03, the best method is SHARO and for the rest of weekly data splits: W02, W03, N_W01 and N_W04, the best method is MR. Each row displays mean and p-value for each configuration, and if p-value > 0.10, then it is highlighted with gray colour.	99
Table 3-16: Paired one-tailed T-Test on the hypothesis that for five daily data splits of D01, D02, D04, N_D02 and N_D03, the best method is MSTLRO, for the two daily data splits of D03 and N_D04, the best method is SHARO and MR is the best method in the last daily data split of N_D01. Each row displays mean and p-value for each configuration, and if p-value > 0.10, then it is highlighted with gray colour.	100
Table 3-17: Level1 Monthly Table-Profit Driven Approach.	104
Table 3-18: Level2 Monthly Table-Profit Driven Approach.	105
Table 3-19: Level1 Monthly Table- Risk-Adjusted Approach.	106
Table 3-20: Level2 Monthly Table- Risk-Adjusted Approach.	106
Table 3-21: Level1 Weekly Table-Profit Driven Approach.	107
Table 3-22: Level2 Weekly Table-Profit Driven Approach.	108
Table 3-23: Level1 Weekly Table- Risk-Adjusted Approach.	109
Table 3-24: Level2 Weekly Table- Risk-Adjusted Approach.	110
Table 3-25: Level1 Daily Table-Profit Driven Approach.	111

Table 3-26: Level2 Daily Table-Profit Driven Approach.....	111
Table 3-27: Level1 Daily Table- Risk-Adjusted Approach.....	112
Table 3-28: Level2 Daily Table- Risk-Adjusted Approach.....	113
Table 4-1: Three objectives of single-objective approach.	121
Table 4-2: Six objectives of multi-objective approach.	122
Table 4-3: Nine multi-objective configurations of profit driven approach.....	123
Table 4-4: Three risk-adjusted objectives of single-objective approach.	123
Table 4-5: Five objectives used in risk-adjusted multi-objective approaches.	124
Table 4-6: Twenty multi-objective configurations of risk-adjusted approach.....	125
Table 4-7: GP Parameters for multi-objective approach.....	128
Table 4-8: Paired one-tailed T-Test for the comparison between the best single-objective method and the top two best multi-objective methods for monthly trading of profit driven approach, choosing the top two highest performances, displaying means and p-values for the top two best multi-objective methods and highlighting with gray colour if p-values > 0.10.	132
Table 4-9: Paired one-tailed T-Test for the comparison of the top two best multi-objective methods for monthly trading of profit driven approach, displaying means and p-values for the second best multi-objective methods and highlighting with gray colour if p-values > 0.10.	133
Table 4-10: Paired one-tailed T-Test for the comparison between the best single-objective method and the top two best multi-objective methods for weekly trading of profit driven approach, choosing the top two highest performances, displaying means and p-values for the top two best multi-objective methods and highlighting with gray colour if p-values > 0.10.	135
Table 4-11: Paired one-tailed T-Test for the comparison of the top two best multi-objective methods for weekly trading of profit driven approach, displaying means and	

p-values for the second best multi-objective methods and highlighting with gray colour if p-values > 0.10.	137
Table 4-12: Paired one-tailed T-Test for the comparison between the best single-objective method and the top two best multi-objective methods for daily trading of profit driven approach, choosing the top two highest performances, displaying means and p-values for the top two best multi-objective methods and highlighting with gray colour if p-values > 0.10.	139
Table 4-13: Paired one-tailed T-Test for the comparison of the top two best multi-objective methods for daily trading of profit driven approach, displaying means and p-values for the second best multi-objective methods and highlighting with gray colour if p-values > 0.10.	140
Table 4-14: Paired one-tailed T-Test for the comparison between the best single-objective method and the top two multi-objective methods for monthly trading of risk-adjusted approach, choosing the top two highest performances, displaying means and p-values for the top two best multi-objective methods and highlighting with gray colour if p-values > 0.10.	146
Table 4-15: Paired one-tailed T-Test for the comparison of the top two best multi-objective methods for monthly trading of risk-adjusted approach, displaying means and p-values for the second best multi-objective methods and highlighting with gray colour if p-values > 0.10.	147
Table 4-16: Paired one-tailed T-Test for the comparison between the best single-objective method and the top two best multi-objective methods for weekly trading of risk-adjusted approach, choosing the top two highest performances, displaying means and p-values for the top two best multi-objective methods and highlighting with gray colour if p-values > 0.10.	150

Table 4-17: Paired one-tailed T-Test for the comparison of the top two best multi-objective methods for weekly trading of risk-adjusted approach, displaying means and p-values for the second best multi-objective methods and highlighting with gray colour if p-values > 0.10.	151
Table 4-18: Paired one-tailed T-Test for the comparison between the best single-objective method and the top two best multi-objective methods for daily trading of risk-adjusted approach, choosing the top two highest performances, displaying means and p-values for the top two best multi-objective methods and highlighting with gray colour if p-values > 0.10.	153
Table 4-19: Paired one-tailed T-Test for the comparison of the top two best multi-objective methods for daily trading of risk-adjusted approach, displaying means and p-values for the second best multi-objective methods and highlighting with gray colour if p-values > 0.10.	154
Table 4-20: Performance of single-objective methods and MSTLRO-CXP method in comparison with the buy-and-hold strategy for daily trading splits.	155
Table 4-21: Level1 Monthly Table-Profit Driven Approach.	157
Table 4-22: Level2 Monthly Table-Profit Driven Approach.	158
Table 4-23: Level1 Monthly Table- Risk-Adjusted Approach.	158
Table 4-24: Level2 Monthly Table- Risk-Adjusted Approach.	159
Table 4-25: Level1 Weekly Table-Profit Driven Approach.	160
Table 4-26: Level2 Weekly Table-Profit Driven Approach.	161
Table 4-27: Level1 Weekly Table- Risk-Adjusted Approach.	162
Table 4-28: Level2 Weekly Table- Risk-Adjusted Approach.	163
Table 4-29: Level1 Daily Table-Profit Driven Approach.	164
Table 4-30: Level2 Daily Table-Profit Driven Approach.	165
Table 4-31: Level1 Daily Table- Risk-Adjusted Approach.	167

Table 4-32: Level2 Daily Table- Risk-Adjusted Approach.....	167
Table 5-1: Symbolic regression configuration for GE.....	173
Table 5-2: Symbolic regression configuration for GP	174
Table 5-3: Santa Fe ant configuration for GE.....	176
Table 5-4: Santa Fe ant configuration for GP	176
Table 5-5: Summary table for six testing problems.	177
Table 5-6: The number of available choices for each grammar rule.	181
Table 5-7: Results of standard GP, standard GE and GE with valid map methods on six problems, showing success rate of finding the solution over 100 runs (e.g. 0.87 means that in 87 out of 100 runs it succeeded to find the solution).	187
Table 5-8: Symbolic BNF grammar with Dereference Count (Deref. Count), Reference Count (Ref. Count) and Rule Type for each production rule.....	193
Table 5-9: An example of mapping integer string (15 codons) with symbolic grammar, displaying details for each mapping state (15 mapping states).....	195
Table 5-10: Typed genetic operator testing configuration table.	213
Table 5-11: Results of standard GP, standard GE and GE with valid map methods and typed genetic operators on six problems, displaying success rate of finding the solution over 100 runs for two approaches: Recursive-non-recursive approach and Internal-Leaf approach, and the highest values of ANT, SR1, SR2, SR3, SR4, SR5 are surrounded by red dash-line boxes.....	214
Table 5-12: Confidence interval at 90% of the best methods for each problem compared with standard GP.	215
Table 5-13: Monthly trading results of GE versus GP.....	217
Table 5-14: Paired one-tailed T-Test on the hypothesis that the best method is the method with highest average return for each monthly data splits. Each row displays	

mean and p-value for each configuration, and if p-value > 0.10, then it is highlighted with grey.	218
Table 5-15: Weekly trading results of GE versus GP.	218
Table 5-16: Paired one-tailed T-Test on the hypothesis that the best method is the method with highest average return for each weekly data splits. Each row displays mean and p-value for each configuration, and if p-value > 0.10, then it is highlighted with gray colour.	219
Table 5-17: Daily trading results of GE versus GP.	220
Table 5-18: Paired one-tailed T-Test on the hypothesis that the best method is the method with highest average return for each daily data splits. Each row displays mean and p-value for each configuration, and if p-value > 0.10, then it is highlighted with gray colour.	220

List of Figures

Figure 2-1: Flowchart of the standard Genetic Algorithm (GA).	22
Figure 2-2: Parse tree of arithmetic expression.....	23
Figure 2-3: Parse tree of logical expression.....	24
Figure 2-4: Parse tree of computer code.....	24
Figure 2-5: GP crossover.	27
Figure 2-6: GP mutation.	27
Figure 2-7: A new car-buying decision-making problem with conjectural trade-off solutions.	28
Figure 2-8: An example of two-objective optimization indicating Pareto front, Pareto optimal set, feasible region and infeasible region.....	31
Figure 2-9: a two-objective optimization problem with six different solutions demonstrating concept of Pareto domination.....	33
Figure 2-10: The possible relations of six solutions in objective space.....	34
Figure 2-11: Process of filling the new population of the NSGA-II algorithm.	38
Figure 2-12: A pseudocode of NSGA-II.....	39
Figure 2-13: A pseudocode of crowding distance assignment procedure: Crowding-sort($\mathcal{F}, < C$).	40
Figure 2-14: A pseudocode for crowded tournament selection algorithm to select μ parents into mating pool.....	41
Figure 2-15: An illustration of support line and zone of support.....	43
Figure 2-16: An illustration of resistance line and zone of resistance.	44
Figure 2-17: An illustration of the concepts of support, resistance and a breakout.....	45
Figure 2-18: Weekly price of S&P500 from 2008 to 2010, with 3-week (blue) and 12-week moving average (red).	46

Figure 2-19: Weekly price of S&P500 from 2008 to 2010, with example of four trend lines: two resistances and two supports.	48
Figure 2-20: Weekly price of S&P500 from 2008 to 2010, with 3-week ROC and 12-week ROC.	50
Figure 2-21: Weekly price of S&P500 from 2008 to 2010 showing a strong uptrend being confirmed by a strong increase in volume.	51
Figure 3-1: An example trading rule in the GP population.	63
Figure 3-2: The trading strategy model of the technical trading rule.	64
Figure 3-3: An example of monthly trading (unit = month) of 5 successive periods with total length at 60 months (5 years) for performance consistency (<i>PC</i>) at $K = 12$ months and $X = 5$ periods.	69
Figure 3-4: GP subtree-swap crossover. Two crossover points in both parents are indicated by red circles.	72
Figure 3-5: GP Subtree Mutation.	73
Figure 3-6: GP Point Mutation.	74
Figure 3-7: GP Permutation.	75
Figure 3-8: GP Hoist.	76
Figure 3-9: The S&P500 index over the period 1960-2010, illustrating four of the data splits for the case of monthly trading: M01-M04.	81
Figure 3-10: The S&P500 index over the period 1960-2010, illustrating four of the data splits for the case of monthly trading: N_M01-N_M04.	81
Figure 3-11: Characterizing the buy-and-hold performance for each data split. Monthly splits are on the top, weekly splits in the middle and monthly splits on the bottom. Each bar shows relative proportions of the buy-and-hold performance in the training (lower), validation (middle) and evaluation or test (upper) periods of the data split.	84

Figure 3-12: Rule analysis at Level1 and Level2; there are 6 technical indicators for Level1 (red circles) and 3 subtrees (at height-2) for Level2 (blue dash-line boxes). ...	103
Figure 3-13: Level1 Monthly Proportion-Profit Driven Approach.....	103
Figure 3-14: Level2 Monthly Proportion-Profit Driven Approach.....	104
Figure 3-15: Level1 Monthly Proportion-Risk-Adjusted Approach.....	105
Figure 3-16: Level2 Monthly Proportion- Risk-Adjusted Approach.....	106
Figure 3-17: Level1 Weekly Proportion-Profit Driven Approach.....	107
Figure 3-18: Level2 Weekly Proportion-Profit Driven Approach.....	108
Figure 3-19: Level1 Weekly Proportion-Risk-Adjusted Approach.....	108
Figure 3-20: Level2 Weekly Proportion- Risk-Adjusted Approach.....	109
Figure 3-21: Level1 Daily Proportion-Profit Driven Approach.	111
Figure 3-22: Level2 Daily Proportion-Profit Driven Approach.	111
Figure 3-23: Level1 Daily Proportion-Risk-Adjusted Approach.	112
Figure 3-24: Level2 Daily Proportion- Risk-Adjusted Approach.	113
Figure 3-25: The suggestion of investment by evaluating the component “ $MN-2 < MX-1$ ” in the trading rule.	116
Figure 3-26: Indication of up-trending market when case 1: “ $MN-2 < MX-1$ ” and case 3 “ $MX-2 < MN-1$ ” becoming true, and Indication of down-trending market when negative forms of case 1 (“ $MN-2 \geq MX-1$ ”) and case 2 (“ $MX-2 \geq MN-1$ ”) becoming true.	118
Figure 4-1: Performance of multi-objective methods in comparison with single-objective configurations for monthly trading splits. E.g. 2CMR (MJV) achieves 75%, meaning that: averaged over all 8 monthly data splits, 2CMR (MJV), in 75% of the 20 independent tests, outperformed all of the single-objective methods.	130
Figure 4-2: Performance of multi-objective methods in comparison with the buy-and-hold strategy for monthly trading splits. E.g. most algorithms achieve 100%, meaning	

that: averaged over all 8 monthly data splits, the method out-performed buy and hold in all of the 20 independent tests.....	131
Figure 4-3: Performance of multi-objective methods in comparison with single-objective configurations for weekly trading splits.....	134
Figure 4-4: Performance of multi-objective methods in comparison with the buy-and-hold strategy for weekly trading splits.....	135
Figure 4-5: Performance of multi-objective methods in comparison with single-objective configurations for daily trading splits.....	138
Figure 4-6: Performance of multi-objective methods in comparison with the buy-and-hold strategy for daily trading splits.	138
Figure 4-7: Performance of multi-objective methods (incorporating risk) in comparison with single-objective configurations for monthly trading splits.	144
Figure 4-8: Performance of multi-objective methods (incorporating risk) in comparison with the buy-and-hold strategy for monthly trading splits.....	145
Figure 4-9: Performance of multi-objective methods (incorporating risk) in comparison with single-objective configurations for weekly trading splits.....	148
Figure 4-10: Performance of multi-objective methods (incorporating risk) in comparison with the buy-and-hold strategy for weekly trading splits.....	149
Figure 4-11: Performance of multi-objective methods (incorporating risk) in comparison with single-objective configurations for daily trading splits.....	152
Figure 4-12: Performance of multi-objective methods (incorporating risk) in comparison with the buy-and-hold strategy for daily trading splits.	153
Figure 4-13: Level1 Monthly Proportion-Profit Driven Approach.....	157
Figure 4-14: Level2 Monthly Proportion-Profit Driven Approach.....	157
Figure 4-15: Level1 Monthly Proportion-Risk-Adjusted Approach.....	158
Figure 4-16: Level2 Monthly Proportion-Risk-Adjusted Approach.....	159

Figure 4-17: Level1 Weekly Proportion-Profit Driven Approach.....	160
Figure 4-18: Level2 Weekly Proportion-Profit Driven Approach.....	161
Figure 4-19: Level1 Weekly Proportion-Risk-Adjusted Approach.....	162
Figure 4-20: Level2 Weekly Proportion-Risk-Adjusted Approach.....	163
Figure 4-21: Level1 Daily Proportion-Profit Driven Approach.	164
Figure 4-22: Level2 Daily Proportion-Profit Driven Approach.	165
Figure 4-23: Level1 Daily Proportion-Risk-Adjusted Approach.	166
Figure 4-24: Level2 Daily Proportion-Risk-Adjusted Approach.	167
Figure 5-1: BNF Grammar for both symbolic regression problem and symbolic integration problem.....	173
Figure 5-2: BNF Grammar for Santa Fe ant Trail.....	175
Figure 5-3: A mapping between the grammatical evolution (GE) system and a biological genetic system.	178
Figure 5-4: GE fitness calculation	180
Figure 5-5: An example of codons used to map the rules.....	181
Figure 5-6: An example of codons generated invalid genome.	184
Figure 5-7: Unwrapped codons after reusing codons in Figure 5-6 3 times.....	184
Figure 5-8: Unwrapped codons after fixing with valid map method2	185
Figure 5-9: Unwrapped codons after fixing with valid map method3	185
Figure 5-10: One-point crossover.	189
Figure 5-11: Two-point crossover.....	189
Figure 5-12: Flip mutation.	190
Figure 5-13: Codons used to demonstrate the process of typed genetic operators.	195
Figure 5-14: Derivation tree of codons in Figure 5-13 for initial mapping state, state1, state2 and state6.	196
Figure 5-15: Derivation tree of codons in Figure 5-13 for final mapping state.....	196

Figure 5-16: Derivation tree of the first parent with the first crossover point at position 8 highlighted with bright green colour.....	198
Figure 5-17: Derivation tree of the first parent with valid token starting at position 8 and ending at position 13 highlighted with bright green colour.	199
Figure 5-18: Derivation tree of the second parent with valid token starting at position 2 and ending at position 5 highlighted with bright blue.....	200
Figure 5-19: Derivation tree of the first child with the valid token from the second parent highlighted with bright blue colour.....	201
Figure 5-20: Derivation tree of the second child with the valid token from the first parent highlighted with bright green colour.....	202
Figure 5-21: Subcodon Mutation	204
Figure 5-22: Point Mutation.....	205
Figure 5-23: Hoist mutation.	207
Figure 5-24: Collapse Subcodon Mutation.	209
Figure 5-25: Grow Mutation.	210
Figure 5-26: Expansion Mutation.	212

1. Introduction

Trading in financial markets is a profitable business that attracts many people running into this financial battlefield, as one could turn into a millionaire within one day. However, one could make himself trouble by losing all his money too, if he made the wrong decision when trying to predict the trend in the financial markets. For this reason, investors, also called traders, need trading strategies to guide them in finding when are the right times to buy and sell securities, and basically there are two common strategies: the Buy-and-Sell (B&S) strategy and the Buy-and-Hold (B&H) strategy. In the first approach, the buy and sell strategy, technical analysis is used, which is a financial technique to forecast future price movements, to capture trends and generate the right signals for buying and selling shares to make a profit. This can be done with the help of various tools in technical analysis such as technical plotting and technical indicators, and those tools can be used to form the technical trading rules that trigger such profitable trading signals to the investors for active and frequent trades in financial markets. However, there has been long-standing debate about the use of technical analysis to find technical trading rules for trading in financial markets. Many believe that it has predictive power and can be profitable, and some claim that it has forecasting ability but no profitability, while a number of economists continue to be convinced that it has no predictive power at all and believe in the efficient stock market and the random walk hypotheses [30-33], key theories in economics and finance, which state that there should not be any discernable and exploitable pattern in stock price data, as financial markets are efficient and stock market prices evolve according to a random walk (see section 1.2 for more detail). The outcome for this scholarly debate seems inconclusive. Another alternative, simple and common approach is known as the buy-and-hold strategy, which works in the more passive sense of an investment strategy. As its name suggests, buy-and-hold works by buying the securities on the first day of a particular

period, holding them during the period and selling them on the last day of that period, and normally investors in this approach buy and hold the assets for a long-term investment (holding them for a long period of time), regardless of small price fluctuations. Comparing between investors of the two approaches (buy-and-sell and buy-and-hold), traders in the buy-and-sell strategy believe in their more active investment strategy and assure us that they can yield greater profit than investors using the second strategy.

With the revolution in Artificial intelligence (AI) since 1987, there have been many attempts to use machines and algorithms to increase the chance of acquiring and accumulating wealth. Particularly in the field of Evolutionary Computing (EC), which has been developed rapidly over the past decade, there have been a number of studies applying EC and other biologically inspired algorithms to solve real-life problems. One of the strong interests in real-world problems is that of applying Evolutionary Algorithms (EA) for making a profit in financial markets. Genetic Programming (GP) is the one member in the family of EAs which has been most applied to a diverse range of problems in finance. Among diverse applications, evolving technical trading rules from historical data by using GP with technical analysis is an attractive one on which we have focused, as GP systems can automatically generate and adapt profitable trading rules and devise them to generate appropriate signals for buying/selling securities over particular periods for active trading in the financial markets. Previously there have been a number of attempts to use GP and other evolutionary algorithms with technical analysis for acquiring technical trading rules (see section 2.4 for more review), and all these previous works have encountered same common problems of overfitting, which is the major issue in the field of machine learning and data mining techniques, including GP. Overfitting can occur when a model evolved from the learning algorithms fit the training data set too perfectly, and the performance of that model is increasingly

impressive in fitting known data but becomes worse in predicting unseen data as it is not generalized to the new data. This leads to one of the significant open issues in the field of GP, namely, generalization in GP [78]. It is quite often that the lack of generalization ability is a result of lack of good prior planning by the practitioner. Moreover, it should be noted that the financial markets is a dynamic environment, in which the underlying fitness landscape and the related optimal solutions are continually changing over time [109]. As a result, it is very challenging to make a profit in such a dynamic environment, and to avoid over-fitting by evolving solutions with good properties of generalization [78].

In addition, another problem involved with EAs is about the selection of a proper fitness function. Basically, we can classify the fitness functions for evolving technical trading rules with EAs into two approaches: profit driven approaches and risk-adjusted approaches. In the first approach, the focus is on maximizing on profit only without being concerned with risk factors in its trading, whereas the latter approach includes risk components in the fitness evaluation, considering risk factors as an essential aspect for both human traders and companies. For technical analysis, there is another issue we should take into account, which is the selection of technical indicators to use as there are lots of available technical indicators. If too many indicators are used in EAs, then the size of the search space becomes larger, and this may result in poor performance of trading as EAs cannot find good trading rules in reasonable time due to the huge search space. Moreover, another point that we need to be concerned with is what is the appropriate set of indicators for the given financial data set, which might be a global market index (e.g. Dow Jones, S&P500) or stocks of individual companies. Among previous studies, the study of Becker and Seshadri [6, 8] on S&P500 index data looked promising as they claimed that their results could outperform the buy-and-hold strategy, which is commonly used as a well-known standard benchmark, and many previous

attempts failed to beat this standard strategy (see section 2.4 for more detail). This successful work made a number of changes to Allen and Karjalainen's regime [2], such as the use of monthly data instead of daily data. However, it is unclear that the configuration of Becker et al., which could beat the B&H strategy in case of monthly trading, would get the same performance for the more volatile data of weekly and daily trading. In addition, Becker et al. also introduced the fitness function that measures the number of well-performing periods rather than just total return, and the results of their experiments indicated the robustness of this new fitness method; but again the investigation had not been done on testing this fitness function on weekly and daily trading, or even on more than one monthly data period to offer more extensive evidence on its robustness. The guidelines for generating results that robustly outperform buy-and-hold should be more completely investigated in all cases of monthly, weekly and daily trading. Furthermore, it should be noticed that the work of Becker et al. and many previous researchers, focused on a single-objective fitness function, and this leads to question whether the use of multi-objective methodologies can bring about superior performance over a single-objective approach. This assumption is supported by the studies in [41, 102], which indicated improvement of genetic programming generalization ability by using multi-objective optimization and a three-data-sets methodology respectively, and it should be extensively investigated with a number of multi-objective configurations that are constructed from combinations of diversified fitness functions with both profit driven type and risk-adjusted type. Finally, the last issue that came to our attention is about the recent new member in the family of Evolutionary Algorithms: Grammatical Evolution (GE). GE emerged with many apparently beneficial features, and a number of publications reported impressive performance of GE on various kinds of problems [10, 11, 69, 73, 77]. As a result, it seems worthwhile and attractive to understand the GE algorithm and apply it to solve

real-life problems like trading problems, and this will be more fascinating if a comparison on performance is done between GE and GP.

1.1 Motivation

Making a profit in the stock market with evolutionary algorithms is very challenging as we already know that stock prices are time varying and highly volatile, and there are many factors that influence stock price changes. Evolving in a dynamic environment, financial trading with evolutionary algorithms is one of the most interesting real-life problems. The main key to achieve this elusive, desirable goal is to generate technical trading rules that are robust, which means that they can perform well on new, unseen data, not just only on the training data set, and this main key is also related to the substantial open issue of generalization in [78]. In addition, making good parameter choices (i.e. parameter choices for crossover and mutation) for the evolutionary algorithms is also a challenging key issue when tackling dynamic problems like trading in financial markets, as they impact on diversity of the population and the ability of GP to escape from local optima [109]. Previously many attempts have failed to acquire robust trading rules which could outperform the standard benchmark of the buy-and-hold strategy when evaluated with test data sets, and these results can be seen as supporting two main hypotheses in economics and finance, the “Random Walk Hypothesis” and the “Efficient Market Hypothesis (EMH)”, and implied that we could not make any profit from technical trading rules. As we already mentioned in the above section, however, there were many studies that rejected these hypotheses and demonstrated that technical trading rules could be profitable. In consequence, the forecasting ability of technical analysis working with evolutionary algorithms has not been conclusive. And, this has motivated us to investigate and provide more convincing evidence to bear on the question of whether technical analysis has predictive power or

not. Moreover, inconclusive results from using evolutionary algorithms to evolve profitable technical trading rules in the literatures leads to the following research questions: “Can evolutionary algorithm generate robust technical trading rules?”, and “What is an appropriate setup for tackling dynamic problems using evolutionary approaches for financial trading (i.e. trade frequency, use of validation set and design of grammar)?”.

Due to reports of successful studies in [6, 8], it seems worthwhile that the regime used in these studies with genetic programming should be empirically investigated for applying to the more fluctuating data of weekly trading and daily trading. Moreover, the robustness of the fitness function that measures the number of well-performing periods in those studies should be extensively tested on monthly, weekly and daily trading. Next, it is very interesting to find out whether or not genetic programming working with multi-objective configurations can give rise to robustly better-ranking performance over single-objective configurations on both profit driven and risk-adjusted approaches in all three types of financial data: monthly, weekly and daily data. Building on evidence that the use of multi-objective optimization [102] and a three-data-set methodology [41] can lead to improvement of genetic programming generalization ability, this addresses the research question of whether multi-objective methods can bring about superior performance. All of the above-mentioned issues have provided the motivation for this research, so that we could in the end point toward guidelines for generating robust and successful trading rules, especially when comparing with the standard benchmark of buy-and-hold, with either single-objective or multi-objective genetic programming approaches. A final motivation related to this thesis comes from grammatical evolution (GE). Due to a number of beneficial aspects of GE and reports on the successful performance of various kinds of its applications, these catalysed our interest in the GE genome mapping process. However, experiences with the canonical

GE approach led us to the additional research question of whether we can find a new GE approach to bring about improvements over both standard GE and standard GP in trading and other problems. This research question motivated and led us to develop a new approach consisting of a new suite of operators with the feature that allows us to switch a GE search between GE and GP styles, and also explore styles intermediate between the two, and we believe that this new approach will enhance GE search and deliver satisfactory results on various applications.

1.2 Technical Analysis versus Random Walk & Market Efficiency Hypotheses

In this section, we aim to provide brief detail about the forecasting power of technical analysis. Criticized by two key hypotheses in economics and finance, the Random Walk Hypothesis and the Efficient Market Hypothesis (EMH), the profitability of technical analysis has been an argument in the community of economists and financiers for ages. The Random Walk Hypothesis states that stock market prices evolve on account of a random walk. Therefore, the prices of the stock market cannot be predicted, and this is consistent with the EMH that the market is efficient. According to these two hypotheses, they imply that there should not be any discernable and exploitable pattern in the financial data. On the other hand, this means that traders could not make any profit from technical trading rules generated by technical analysis, or it is impossible to beat the market.

Let begin to consider the three forms of the EMH: weak form, semi-strong form and strong form [31], and the detail of each one are as follows [11, 97].

- *The weak form of market efficiency* theorizes that a share price at any point in time reflects all the information contained in its price history, and it implies that the past pattern of price changes cannot be used to predict future price changes.

- *The semi-strong form of market efficiency* states that a share price at any point in time reflects all readily available information which could affect the share's price. It means that published information cannot be used to predict future price changes, but there is still other information not publicly available and is not fully reflected in the price. This implies that profitable trading, sometimes called insider trading, can be done by using information not yet known to the public.
- *The strong form of market efficiency* indicates that a share price at any point in time reflects all information available, including both public and private (non-public) information, and this implies that there is no information, published or not, which investors and traders can use to predict future price changes.

Since the Random Walk Hypothesis and the EMH emerged, there were a number of studies in the 1960s and 1970s that supported these hypotheses as follows: Alexander in 1964 [1], Fama in 1970 [31], Fama and Blume in 1970 [34], Jensen and Bennington in 1970 [50]. However, there were also a number of studies standing on the other side, rejecting these hypotheses. For instance, Pruitt and White in 1988 [82] developed the CRISMA trading system which showed positive returns over a 10-year period using transaction costs at 2%. Brock et al. in 1992 [13] successfully produced significant excess returns by investigating stock index trading on the S&P 500 using two test trading strategies, moving average and trading range break. Moreover, Bessembinder and Chan in 1995 [9], demonstrated that simple trading rules could be profitable as well, however, regardless of transaction costs. All these three works revealed that positive excess returns, compared with buy-and-hold, can be accomplished using technical trading rules. In terms of the effect on technical analysis caused by these hypotheses, there are different opinions about the profitability of technical analysis. Many believe that technical analysis can be used to predict the stock price and make a profit, while some claim that it has predictive power but not enough to make any profit.

In addition, in the academic and financial world, many academics and fundamentalists don't agree with that and are convinced that it has no forecasting ability at all following EMH. On the other hand, technical analysts do not believe that the market is inefficient in the process of absorbing available information into security prices; they instead agree that prediction of market prices is difficult. As a result, there have been substantial arguments for a very long time among financial theorists; as can be seen, it is inconclusive whether technical analysis is profitable or not. Apart from those arguments, it is nevertheless widely applied in practice, and its apparent profitability can be seen from much evidence, some of which we discuss next.

Let us begin with the study of Taylor and Allen in [100] on behalf of the Bank of England; their study revealed that roughly 90% of financial institutions dealing in foreign exchange in London utilized information derived from technical analysis to some degree. Following the consideration of using technical analysis in stock markets of Brock, Lakonishok and LeBaron in [13], they found evidence that simple technical trading rules had predictive power and concluded that the findings of earlier studies that technical trading rules did not have such power were 'premature'. Further, the studies of Sweeny in [99] and Levich and Thomas in [57] concluded that technical trading strategies may be profitable in the case of foreign-exchange markets [11].

1.3 Contribution

This thesis provides an empirical study of using genetic programming (GP) to evolve robust technical trading rules for monthly, weekly and daily trading with both single-objective and multi-objective methodologies and it also provides fundamental analysis on the technical trading rules generated from multiple experiments with both the single and multi-objective configurations. This first strand of work can be considered as within the interface of evolutionary computation and finance. In addition, the second strand of

work explores an aspect of Grammatical Evolution (GE) through the development of strategies to reduce the number of invalid individuals resulting from incomplete mapping. It introduces new approach for grammatical evolution (GE) with a new suite of operators to bring about an improvement on the GE search process. From these two distinct strands of work, this thesis adopts a multi-disciplinary approach and therefore produces contributions spanning both the computer science (particularly in the field of evolutionary computation) and finance domains. In order to make a clear distinction between the contributions if the thesis to computer science and its contributions to finance, the specific contributions for each domain will be indentified separately as follows.

1.3.1 Contributions to Computer Science

1. Proper practice using training, test and validation sets (three data sets methodology) for model selection to choose the rules, and using varied data spits to test the robustness and sensitivity to the data of the rules. This proper practice, used in the work of Chapters 3 and 4, makes contributions across both computer science in the field of evolutionary computation (EC) and the finance domains.
2. General lessons related to the significant open issue of generalization, which indicate that the unpromising results of previous attempts to evolve profitable trading rules, were due in part to a methodology that led to poor generalization.
3. We demonstrate that a multi-objective methodology, which resists over-fitting by spreading functional complexity of the solutions throughout different expressions of each objective, can be use to increase the level of generalization, supported by convincing evidence from the results of multi-objective optimization in Chapter 4.
4. We provide additional evidence that making appropriate parameter decisions can lead to a successful GP application even in a dynamic problem environment, and

this has been shown by using dynamic forms of mutation (4 different mutation operators) during a single GP run. This evidence is supported by the results of the experiments in Chapter 3 and Chapter 4.

5. We provide a comprehensive empirical study of genome mapping methods for grammatical evolution in Chapter 5, and also develop a new suite of operators that effectively allow us to vary a GE search between GE and GP style, by changing the rates of application of certain operators. This new suite of operators leads to a new GE approach that appears very effective, in comparison to standard GP and standard GE, when tested on a range of standard GE and GP test functions. This new approach also allows us to find GE configurations that are also effective in the trading context (unlike standard GE).

1.3.2 Contributions to Finance

1. A new and thorough evaluation of the capability of genetic programming to evolve profitable technical trading rules that can outperform a buy-and-hold strategy. In previous work using GP for trading, results have been often unpromising, but in Chapter 3 we replicate the more promising work of Becker and Seshadri, and we also build on that work in several ways and test the technique in several different trading environments. This enables us to identify, with more confidence than in the work of previous researchers, the conditions in which GP-evolved technical trading rules may be able to outperform the buy-and-hold strategy.
2. The development and evaluation of several multi-objective approaches to evolving technical trading rules with genetic programming. With comprehensive experiments in Chapter 4, we find a subset of configurations (mainly concerning the choice of objectives) that lead to robust and successful trading rules. We also

find that, unlike the single-objective approach, the use of the multi-objective approach can lead to successful weekly trading.

3. The results of the experiments in Chapter 3 and Chapter 4 also provide additional evidence to support the forecasting ability of technical analysis working with evolutionary algorithms, regardless of Random Walk and Efficient Market Hypotheses.
4. By analysing the technical trading rules that arise from multiple experiments with both the single and multi-objective approaches (in sections 3.5 and 4.5), we are able to contribute some insights into trading strategies that are appropriate for different trading environments.

1.4 Thesis Scope and Limits

The scope and limits of the thesis have been stated implicitly in the introduction, motivation and contributions. However, to make them clearer, this section clearly highlights the main scope and limits in the following list.

1.4.1 Scope of the thesis in terms of Fitness Functions:

The fitness functions to use in this thesis are mainly classified into two main groups: (a) profit driven approaches that reward trading rules on the basis of their returns and (b) risk-adjusted approaches that incorporate penalties based on the chance of loss. For the single objective configurations, in the first group, the fitness functions has been limited to consist of two basic fitness types: Market Return (MR) and Performance Consistency (PC), varying looking up periods for 4 different periods for monthly trading (namely 6, 12, 18 and 24 months) and for 2 different periods in the cases of weekly and daily trading (12 and 24 weeks for weekly trading, and 12 and 24 days for daily trading). In the risk-adjusted group the fitness functions contain the common risk measurements of the Sharpe Ratio (SHARO) and Modified Stirling Ration (MSTLRO). In addition, all

fitness functions in the single-objective methods are modified by the complexity-penalizing factor. Next, in the case of the multi-objective configurations, each configuration comprised two, three or four specific objectives. Each individual objective is essentially from the corresponding single-objective method, but also included are the additional objectives: 2 Separated Market Return (2MR), Modified Drawdown (MMDD) and Complexity Penalizing Factor (CXP) (Please see section 3.2.3 and 4.2.2 for more details).

1.4.2 Scope of thesis in terms of Technical Indicators:

The Technical indicators used in all trading experiment in this thesis are based on Becker and Seshadri's work [6, 8], which used six groups of indicators: Prices, Volumes, Moving Averages, Rate of Change, Price Resistance and Trend Line (Please refer to section 3.2.2 for more details).

1.4.3 Scope of Methodologies and Techniques to Provide Generalization in Dynamic Environments:

A three-data-sets methodology for evolutionary model induction is used to choose the rules to evolve solutions with good properties of generalization for single-objective approaches in Chapter 3, and the use of multi-objective methodology is also included to increase the level of generalization in Chapter 4.

1.4.4 Scope of thesis in terms of Parameters and Configuration:

Fixed rates of crossover rate and mutation rate are used in all experiments of Chapter 3, 4 and 5, and a dynamic form of GP (using four different dynamic forms of mutation in a single run) are used to enhance GP's performance when working in the real-world dynamic environments of financial markets (Refer to Table 3-2 and Table 4-7). In Chapter 5, testing the new GE approach on trading problems is restricted to

configurations within the single-objective methodology, and the repair method is also fixed to only the MAP03 genome mapping method. Moreover, the number of combinations of parameters to vary a GE search between GE and GP styles is limited to 10 configurations (Refer to Table 5-10).

1.4.5 Scope of thesis in terms of Data Sets:

Data sets used to evolve profitable trading rules came from the composite index of Standard and Poors 500 (S&P500), a fixed set of 500 stocks which aggregate to daily price indicators (opening, closing, high, low), in three different data types: monthly, weekly and daily data, in the period from 1960 to 2010, and the risk-free assets used in this thesis were the US Treasury bill data from <http://research.stlouisfed.org>. Furthermore, all data splits of monthly, weekly and daily data that we used in the experiments were pre-defined periods with different lengths in the training set, validation set and evaluation (test) set.

1.5 Thesis Structure

The following list provides the organization for the remaining chapters in this thesis.

- Chapter 2 aims to provide background information on Evolutionary Computing (EC), Evolutionary Algorithms (EAs) including Genetic Algorithm (GA) and Genetic Programming (GP), Multi-objective Optimization (MOO) concepts and Multi-objective Evolutionary Algorithms (MOEA) including the NSGA-II algorithm. It also gives general concepts of technical analysis and details for each technical indicator that is used in the experiments in this thesis, and in the last section it has a literature review of related works using EAs for evolving trading rules with both single-objective and multi-objective approaches.

- In Chapter 3, it focuses on using GP to discover robust trading rules with single-objective configurations on monthly, weekly and daily data, and the configurations in this chapter can be separated into two approaches: profit driven and risk-adjusted approaches. Moreover, the performance comparison between each configuration and the fundamental analysis of basic and common structures of best technical trading rules discovered during the experiments are provided in the chapter as well.
- In Chapter 4, basically the approach used in this chapter is similar to that used to describe the single-objective configurations in previous chapter but with a multi-objective approach. It investigates the use of multi-objective GP to acquire the optimal set of technical trading rules, and presents the results of the experiments on monthly, weekly and daily data, focussing on two characteristics: the relative performance of the multi-objective strategies over single-objective configurations and the relative performance between the multi-objective strategies and the standard benchmark buy-and-hold strategy. In addition, the fundamental analysis on basic structures of technical trading rules evolved with multi-objective configurations is also given.
- Chapter 5 presents a new approach for Grammatical Evolution (GE) including new genome repair strategies and typed genetic operators. And, it also provides the test results of this new approach on various kinds of problems including symbolic regression, symbolic integration, Santa Fe Ant trial and financial trading problem, compared with standard implementations of both GE and GP.
- Finally, Chapter 6 summarizes the conclusions of this thesis and offers a discussion about future work, and it also lists again and comments on the contributions achieved from the work described in this thesis.

1.6 Publications

1. Lohpetch, D., and Corne, D. (2009). "Discovering Effective Technical Trading Rules with Genetic Programming: Towards Robustly Outperforming Buy-and-Hold." *World Congress on Nature and Biologically Inspired Computing (NABIC) 2009*, IEEE Press, 431—437. (The best student paper award).
2. Lohpetch, D., and Corne, D. (2010). "Outperforming Buy-and-Hold with Evolved Technical Trading Rules: Daily, Weekly and Monthly Trading." *Applications of Evolutionary Computation, Pt Ii, Proceedings*, C. DiChio, A. Brabazon, G. A. DiCaro, M. Ebner, M. Farooq, A. Fink, J. Grahl, G. Greenfield, P. Machado, M. O'Neill, E. Tarantino, and N. Urquhart, eds., Springer-Verlag Berlin, Berlin, 171-181.
3. Lohpetch, D., and Corne, D. (2011). "Multiobjective algorithms for financial trading: Multiobjective out-trades single-objective." *Proc., Evolutionary Computation (CEC), 2011 IEEE Congress on*, 192-199.

2. Background and Literature Review

This chapter provides background information and reviews recent and seminal literature that is relevant to the work in this thesis, and an overview of key points for each section is as follows.

The work in Chapter 3 is about using Genetic Programming (GP) with a single-objective methodology for financial trading, and also in Chapter 5 we present a new approach for Grammatical Evolution (GE). Since both GP and GE are members of the evolutionary algorithms family, we start the first section in this chapter with a review of the fundamental concepts of evolutionary computing and the basic elements in evolutionary algorithms, and then go into more details for Genetic Algorithms (GA) and end this section with basic description of GP. Since the work in chapter 4 uses multi-objective GP to acquire near-optimal sets of technical trading rules, the next section therefore explains the concept of multiobjective optimization and a range of Multi-objective Evolutionary Algorithms (MOEA) including the NSGA-II algorithm, which is the main MOEA integrated with our GP system. For GP evolving technical trading rules in both single or multi objective configurations, technical indicators are used as the terminal nodes, so the general concepts of technical analysis and details for each technical indicator that is used in the experiments in this thesis are given in the next section of this review. Subsequently, a section is provided in the literature review detailing some related works using EAs for evolving trading rules with both single-objective and multi-objective approaches. Finally, the last section of the review gives a conclusion indicating key points, including a discussion of how and why the most influential papers of relevance in the area relevant to the thesis were identified.

This chapter is therefore organized as follows:

- Section 2.1 provides a basic description of evolutionary computing (EC) and evolutionary algorithms (EA): Genetic Algorithms (GAs) and Genetic Programming (GP).
- Section 2.2 explains the concept of multiobjective optimization and the use of EAs with multi-objective approaches (MOEA) to solve multi-objective optimization problems (MOOP).
- Section 2.3 presents the basic knowledge of technical analysis for trading in financial markets, and gives details of all technical indicators used in the experiments of this thesis.
- Section 2.4 is a historical review of related work in discovering technical trading rules by using Evolutionary Algorithms (EA) with both single-objective approach and multi-objective approaches, providing brief descriptions for each research work.
- The last section, Section 2.5, provides a conclusion of key points, and a discussion of how and why the most influential papers of relevance in the area relevant to the thesis were selected.

2.1 Evolutionary Computing (EC)

Evolutionary Computing (EC) is a class of optimisation algorithms drawing their basic ideas from the process of natural evolution. Given a population of individuals, competition among all individuals occurs in order to gain the limited resources in the environment such as food to continue to live through adversity of lack of resources, and the stronger individuals have more chance to pass their genes to next generation and live longer than the weaker ones (survival of the fittest). After a number of generations have passed, individuals containing favourable combinations in their genetic material become dominant in their population. During the process of evolution, it is possible that random

changes in the genetic material of individuals may happen, and these may result in new improved individuals which become numerous in the population, or it may result in new individuals having unfavourable changes, who get eliminated by natural selection.

2.1.1 General Scheme of an Evolutionary Algorithm

Evolutionary Algorithms (EAs) are computer-based problem solving systems which mimic aspects of the biological process of evolution. Back in the 1960s, Fogel et al. [36, 37] proposed his algorithm as *Evolutionary Programming (EP)*, associated with real-number vector representation of finite state machines, and Holland [21, 46, 47] introduced a method called a *Genetic Algorithm (GA)* associated with binary strings. Both were invented in the USA. At the same time, Rechenberg and Schwefel [83, 88] presented *Evolution Strategies (ES)* associated with representation of real-value vectors, in Germany. It was for about 15 years that these three methods developed independently, until in the early 1990s they have come to be considered as one technology to be known as Evolutionary Computing (EC). Also in the early 1990s, Koza [4, 54, 55] brought in a fourth member of the EC family, *Genetic Programming (GP)* associated syntax tree based representation. Lastly, the recent member of the EC family, *Grammatical Evolution (GE)*, associated with binary strings representation and a sophisticated mapping process through Backus-Naur Form (BNF) grammars, was proposed in 2001 by O'Neill and Ryan [72, 73, 77].

The main significant components in EA that must be considered in order to define a particular EA are listed as follows [27, 45, 53]:

- **Representation:** This is a definition of individuals to be evolved, specifying the connection between the original problem space (phenotype space) and the search space (genotype space), which is linked through a mapping called an *encoding* or *representation*. Indeed, an EA works on the encoding of the problem, not on

the problem itself. A solution – a good phenotype – is obtained by *decoding* the best genotype after termination. In EA terminology, a solution to the problem is called an *individual* or a *chromosome*, which is often made of discrete units called *genes*.

- **Population:** This is the container of individuals, and it is to hold (the representation of) possible solutions. The population is normally randomly initialized.
- **Evaluation Function:** This is also called fitness function: designed to estimate how good a solution is in solving the given problem.
- **Selection:** This is the mechanism by which individuals from the population are selected to survive, and reproduce.
- **Variation Operators:** These operations ensure exchange of genetic material between individuals (*recombination* or *crossover*), as well as the occasional changing of random genes (*mutation*). In crossover, generally two chromosomes, called *parents*, are combined together to form new chromosomes, called *offspring*, and the parents are selected among existing chromosomes in the population with preference towards good fitness (depending on the selection method) so that offspring are expected to inherit good genes, while mutation provides random changes into the characteristics of chromosomes. Mutation is generally applied at the gene level.

Moving to how EAs work, the EA takes the first step by generating the first initial population of individuals, representing random initial candidates for solutions to the problem, and then the process of evolution continues iteratively to improve the quality of individuals in the population over many generations. The way to measure how good an individual is at solving the problem is done through a quality function or fitness function. A higher fitness value of an individual means a better chance for that

individual to be chosen to seed the next generation. Creating the offspring for the next generation is done by applying two (usually) variation operators, recombination and mutation, to the population. Recombination requires two or more individuals, which are selected by a selection operator that selects stronger individuals with a higher chance, as parents in order to produce one or more children (the new individuals), whereas mutation is applied to one individual and results in one new individual. In consequence, repeatedly applying those two operators on the parents forms a set of new individuals or offspring. Subsequently, new individuals have their fitness evaluated and then compete with the current individuals in the population to survive into the next generation. This process will continue until termination condition is satisfied, such as reaching a limited computational time or a maximum number of generations, or finding an individual with sufficient quality.

In this thesis we used Genetic Programming (GP) as the main tool for running the experiments in Chapters 3 and 4, and in Chapter 5 it was also used to make a direct comparison with Grammatical Evolution (GE) on various problems. Therefore, in the following section we aim to provide a basic overview of Genetic Programming. However, to provide an easy way to understand GP, we should start with the basic idea of Genetic Algorithms first, since basically GP is a variant of GAs working with a different genotype structure [27].

2.1.2 Introduction to the Genetic Algorithms (GA)

In the 1960s GAs were developed by Holland [47], deriving inspiration from Darwinian evolution for an optimization task, using fixed length binary strings as chromosomes to encode solutions to the problem. GAs are population-based algorithms, working on a population and iteratively evolving a set of possible solutions in the population rather than searching a single solution one by one in a search space like standard search

algorithms. The flowchart in Figure 2-1 provides key operations of what goes on in the process of the standard Genetic Algorithm step by step [11, 27].

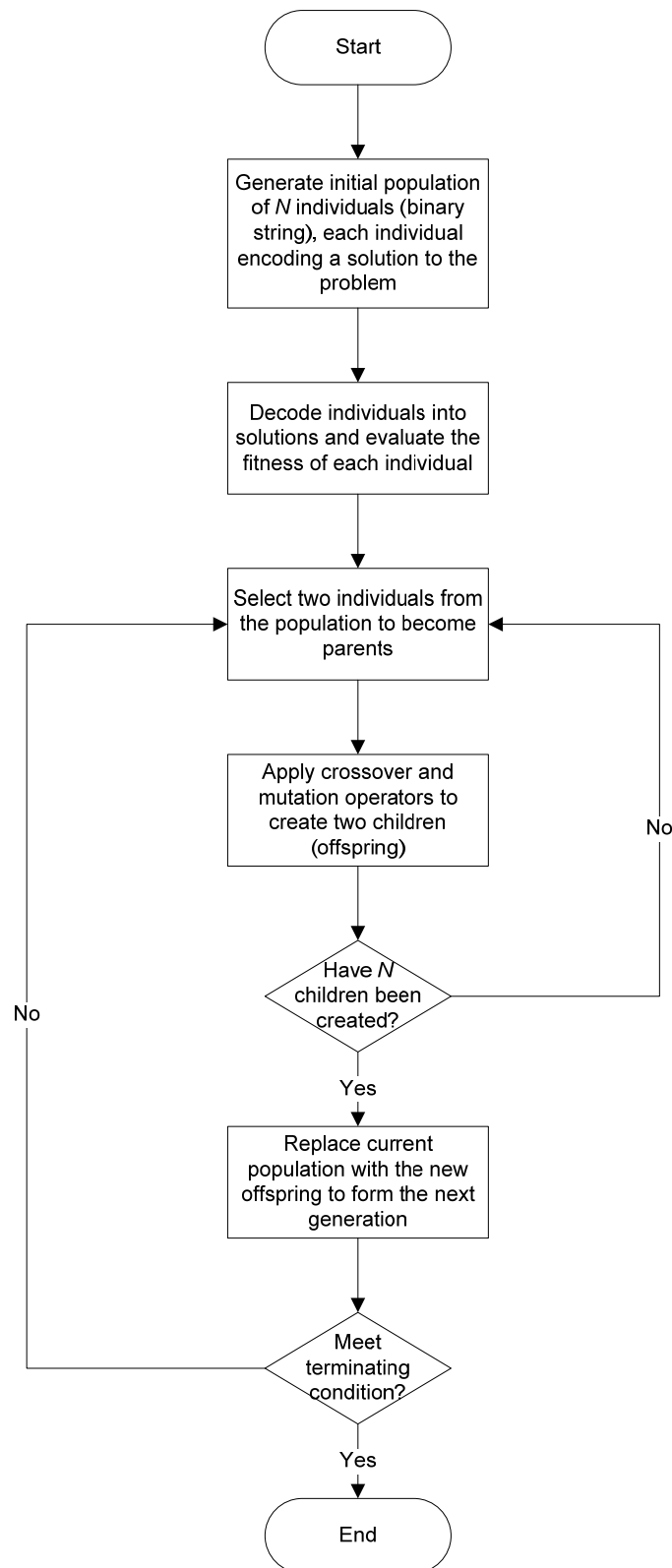


Figure 2-1: Flowchart of the standard Genetic Algorithm (GA).

2.1.3 Introduction to Genetic Programming (GP)

In the previous section we saw an overview of standard Gas; basically GP is simply a variant of GAs working with a different data structure [27], and it marks itself out as different from the GA in that the GP chromosomes are variable-sized representations of parse trees. Designing a GP algorithm requires specification of the following [4, 11, 27, 45]:

- **Representation:** As mentioned above, GP uses a parse tree to represents its solution. Such parse trees are varied depending on the problem we are currently dealing with; they can be parse trees of arithmetic expressions, logical expressions, or computer code as illustrated in Figure 2-2, Figure 2-3 and Figure 2-4 respectively, and normally they have to be interpreted in particular ways to solve the specific problem at hand.

Arithmetic Expression

$$4 \times x - \frac{y}{z + 2} + 12$$

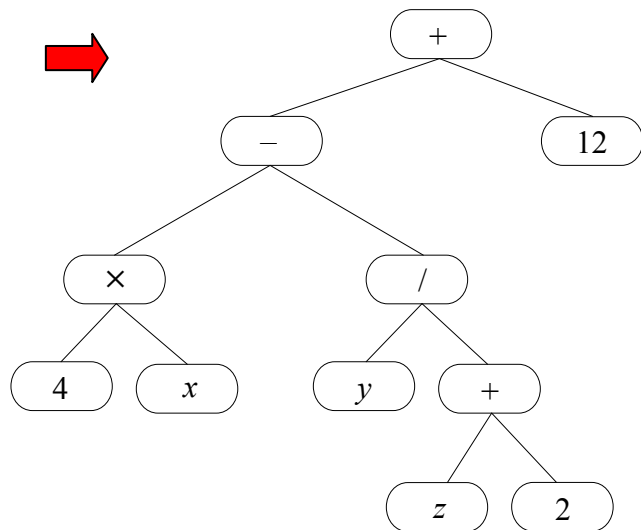


Figure 2-2: Parse tree of arithmetic expression.

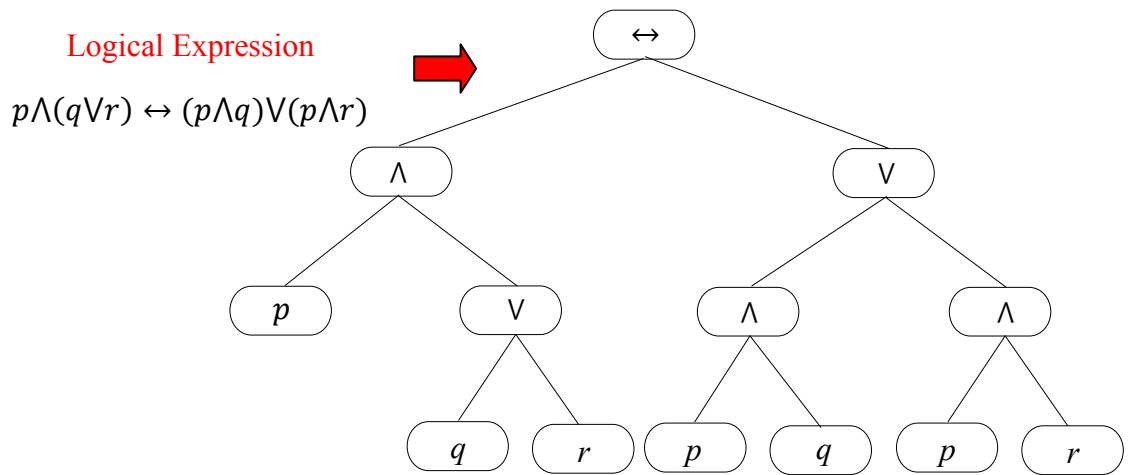


Figure 2-3: Parse tree of logical expression.

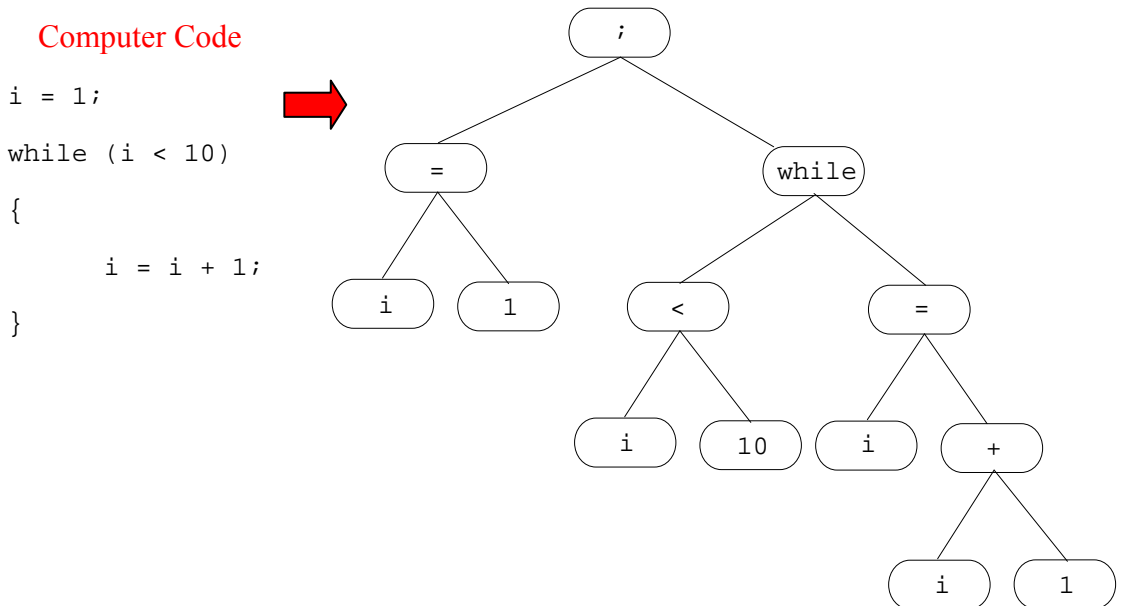


Figure 2-4: Parse tree of computer code.

- Population:** Initially, the population is filled with random parse trees, and these initial random parse trees are generated using elements from two sets: the *function set* and the *terminal set*. Elements of the function set are functions with an arity greater than zero and are allowed as internal nodes, while elements of the terminal set are functions with arity of zero and are allowed as leaf nodes. Please note that the arity of a function means the number of arguments required for that function. Moreover, there are three common ways for generating the

parse trees: *Grow* method, *Full* method and *Ramped-half-and-half* method. Ramped-half-and-half method is the most common method of initialisation in GP since it ensures the diversity of both structure and values of the population, and it combines the Grow and Full initialisation strategies, each of which is used to generate half of the population. In the Full method, parse trees are grown randomly such that all branches reach a predetermined maximum node depth, while in the Grow method trees are grown randomly with no one branch allowed to exceed the maximum node depth.

- **Evaluation Function:** This is an essential part of all evolutionary algorithms including GP, and it plays the main role for quality measurement. Without this component, the evolution process cannot expect progress as it forms the basis for selection, and so it facilitates improvements.
- **Selection:** There are two selection phases occurring in the evolutionary cycle. The first one, *parent selection* or *mating selection*, is to distinguish among individuals based on their quality to allow the better individuals to have more chance to become parents of the next generation, and the role of second one, *survivor selection* or *environmental selection*, is similar to parent selection, but it is used in a different stage of the evolutionary cycle, which is called after generating the offspring from the selected parents. It is a process to select which individuals from both the current population and the offspring will be allowed through to the next generation due to constant population size. There are a number of commonly used selection methods, for examples, *Fitness-Proportionate Selection (FPS)*, *Ranking Selection* and *Tournament Selection*. Above all, *Binary Tournament Selection* is chosen as a main selection method for all experiments in this thesis as it is an operator with the useful property that it does not require any global knowledge of the population and it is very

straightforward to implement, and it is working by randomly selecting two individuals and then the better of the two is selected and returned as a result of selection.

- **Variation Operators:** The main role of variation operators is to create new individuals from old ones. Variation operators in GP are divided into two types based on their arity: Recombination and Mutation.

1. Recombination or Crossover is a binary variation operator and merges information from two parent genotypes into one or two offspring genotypes. The most common implementation is subtree crossover demonstrated in Figure 2-5.
2. Mutation is a unary variation operator and is applied to one genotype and delivers a slightly modified mutant, the child or offspring. The most common implementation works by replacing the subtree starting at a randomly selected node by a randomly generated tree, as displayed by an example in Figure 2-6.

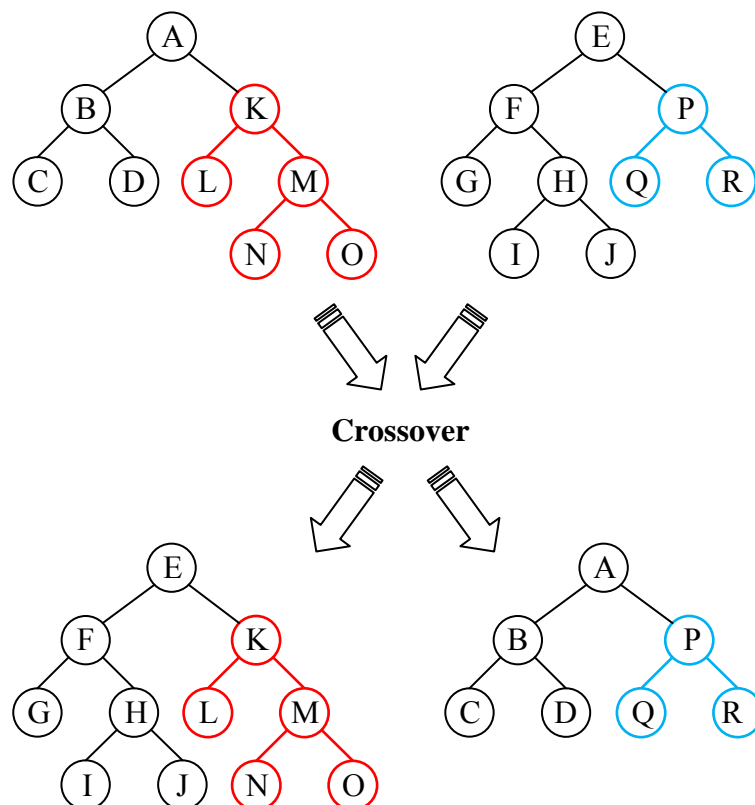


Figure 2-5: GP crossover.

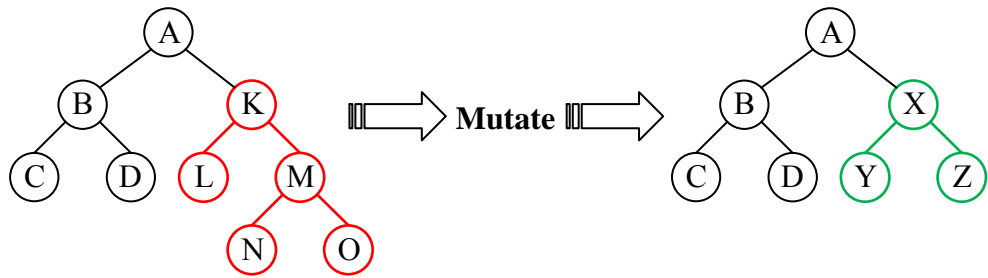


Figure 2-6: GP mutation.

We have decided to choose Genetic Programming (GP) as our main evolutionary algorithm for addressing the financial trading problem, and the main reason is that technical trading rules are naturally represented as trees, which is the standard genotype of GP, by employing internal nodes as operators and leaf nodes as technical indicators. Moreover, representing a technical trading rule as a tree is also easy to understand and interpret by users, which are inventors in financial markets, and this therefore makes it seemingly more preferable to inventors than any other representation. Our GP system and also our GE system were implemented in the C# language, and they were developed by using Microsoft Visual Studio 2008 as the main IDE.

2.2 Multi-objective Evolutionary Algorithms (MOEA)

2.2.1 An Overview of Multi-objective Optimization Problems (MOOP)

A multi-objective optimization problem (MOOP) is a problem involving more than one objective function to be considered in the process of finding one or more optimum solutions. Almost every real-world problem of searching and optimizing solutions by nature involves multiple completing objectives or multiple criteria to be concerned with. Focusing on only the one objective to find the solutions should not be done, when the rest of the objectives are also important; this simply leads to solving the wrong problem. However, when we have many different objectives, it is no longer simple to decide what single solution is the best solution. The set of optimal solutions for a multi-objective

optimization problem actually comprise different solutions produced from trade-offs (conflicting targets) between different objectives, and there is often no single solution which is optimal with respect to all objectives in the set. If we would like to choose a solution that is the best with respect to one objective, then that requires compromising in the performance on other objectives. To give a simple illustration, imagine that we are about to buy a new automobile (this is based on an example of a car-buying decision-making problem from [22]). There are a number of factors that we probably need to take into account such as: price, ranging from four thousand to twenty thousand pounds, and comfort. Let consider two extreme conjectural cars, i.e. one costing about four thousand pounds (solution 1) and another costing about forty thousand pounds (solution 2), as displayed in Figure 2-7. If the cost of the new car is the only concern for all buyers, then the optimal choice in this case is definitely solution 1. As a result, we would not see any expensive cars from car manufacturers on the road. In fact, as we already know, this is not true as this decision-making process in real life is not a single-objective one.

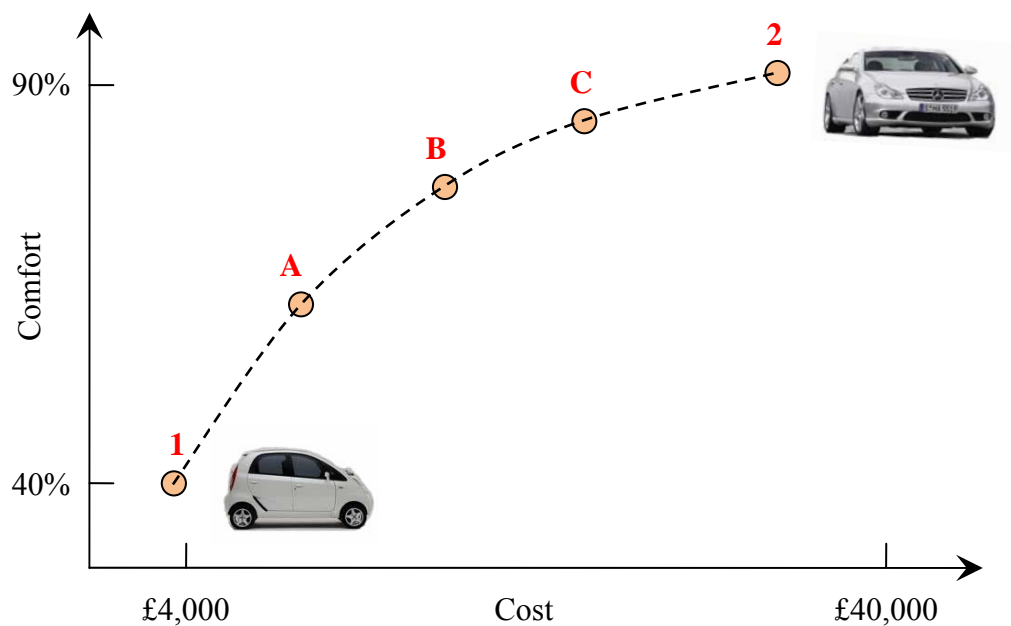


Figure 2-7: A new car-buying decision-making problem with conjectural trade-off solutions.

On the other hand, if comfort is the only concern of rich buyers for making decisions, the answer for them would be solution 2 with a maximum comfort level at 90%, and from the figure it also shows us that an inexpensive car is likely to be less comfortable. For an example, the cheapest car of solution 1 has the lowest comfort level of 40%. Between these two extreme solutions, there exist many other solutions, where a trade-off between cost and comfort can be made, as shown in the figure for solutions A, B and C. These three solutions were produced from raising the quality in one objective with a sacrifice on the other objective.

In the past, solving a multi-objective optimization problem has tended to be done by casting and solving it as a single-objective one due to lack of suitable solution methodologies. To give an example, the *scalarisation* approach works by giving a numerical score to each objective by using a quality function and then weighting and combining these scores into a single fitness scalar value; this approach has been used for many years within research communities (see more details from [19, 22]). However, there are a number of drawbacks to this approach. The rise of Evolutionary Algorithms (EAs) has changed the field of search and optimization over the last few years. An EA is a population-based algorithm as it uses a population to contain the solutions to the problem and evolves them over time, ending with a population of solutions. In the case that an optimization problem has a single solution, it can be expected that, by nature of EA, all elements in an EA population should converge to the optimum solution or a good near-optimal solution. However, if there are multiple optimal solutions of an optimization problem, an EA is still able to capture multiple optimal solutions in its final population. With the ability of an EA that works with a population of solutions and can find multiple optimal solutions in one single simulation run, this makes EAs suitable, beneficial and unique for solving multi-objective optimization problems.

2.2.2 Principles of Multi-objective Optimization

Suppose that we wish to optimize M objectives and we have no clear preference of the objectives relative to each other. All objectives can be the minimization type or maximization type (a minimization type objective can be converted to a maximization type by multiplying by a negative value). The multi-objective optimization problem (MOOP) can be stated in its general form as follows [22].

Definition 2.1: The multi-objective optimization problem (MOOP).

$$\left. \begin{array}{ll} \text{Minimize /Maximize } f_m(x), & m = 1, 2, \dots, M; \\ \text{subject to } g_j(x) \geq 0, & j = 1, 2, \dots, J; \\ & h_k(x) = 0, \quad k = 1, 2, \dots, K; \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \dots, n. \end{array} \right\}$$

A solution x^* is a vector of n decision variables: $x^* = (x_1, x_2, \dots, x_n)^T$ that minimizes/maximizes a given set of M objective functions $z(x^*) = \{z_1(x^*), \dots, z_M(x^*)\}$ in the solution space \mathbf{X} and is restricted by a series of constraints. The last row is the set of constraints called variable bounds, restricting each decision variable x_i to take a value within a lower $x_i^{(L)}$ and an upper $x_i^{(U)}$ bound, and these bounds make up a *decision variable space* D (simply called the decision space). There are J inequality and K equality constraints associated with the problem in rows 2 and row 3, and the terms $g_j(x)$ and $h_k(x)$ are called constraint functions. It is very common for real-life multi-objective optimization problems to have competing objectives conflicting with each other, so it is almost impossible to find a perfect multi-objective solution that gives the best result for each objective function simultaneously as optimizing x with respect to a particular single objective often leads to unacceptable results with respect to the remaining objectives. As a result, a reasonable solution to a multi-objective problem can be found by investigating a set of solutions, where each element in the set satisfies the objectives at an acceptable level without being dominated by any other solution.

Assuming that all objective functions are of the minimization type, a feasible solution vector x is said to *dominate* another feasible solution y ($x \succ y$), if and only if, $z_i(x) \leq z_i(y)$ for $i = 1, \dots, K$ and $z_i(x) < z_i(y)$ for at least one objective function j . Next, if a solution is not dominated by any other solution in the solution space, then that solution is said to be *Pareto optimal*, with the property that it cannot be improved with respect to any objective without declining in quality at least one other objective. And, the *Pareto optimal set* is the set of all feasible non-dominated solutions in \mathbf{X} , and for a given Pareto optimal set, the *Pareto front* is the corresponding objective function values in the objective space. Moreover, it should be noted that the number of Pareto optimal solutions of many real-world problems is tremendous and sometimes infinite [53].

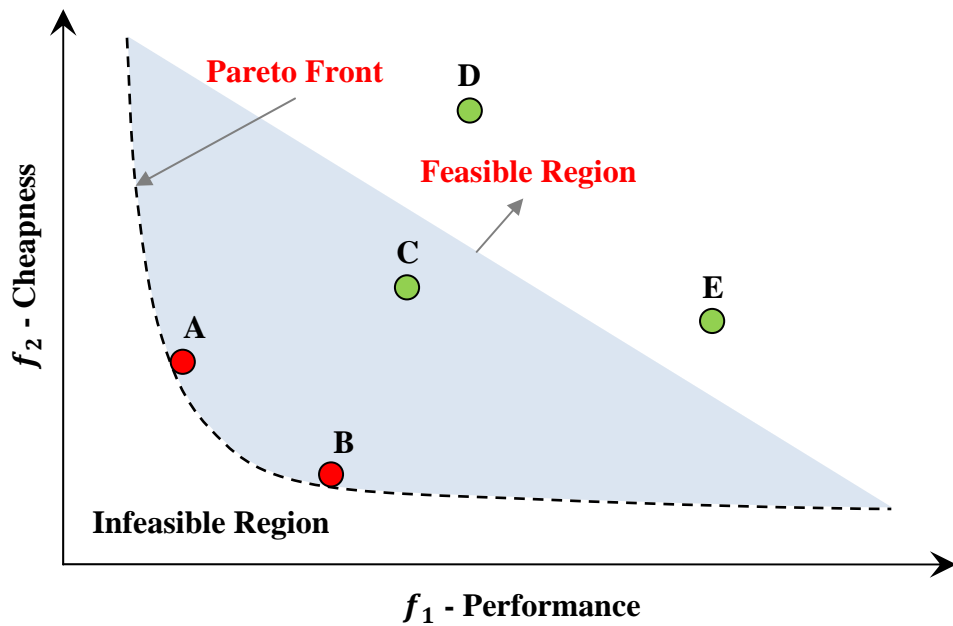


Figure 2-8: An example of two-objective optimization indicating Pareto front, Pareto optimal set, feasible region and infeasible region.

To give a clear picture of the concept of Pareto optimal set and Pareto front, let us consider an example of a two-objective optimization problem of the design of a complex navigation system in Figure 2-8, where the first objective is performance (f_1) and the second is cheapness (f_2) (the inverse of cost), and the goals of this design is to

achieves maximum performance at minimal cost (maximum cheapness). As it can be expected, those two objectives conflict each other; normally desired maximum performance comes with very expensive cost. From the figure, there are five solution points of A, B, C, D and E lying in the feasible region. Among these solutions, the solutions A and B lie on the Pareto front as they are non-dominated by any other solutions in the feasible region of the space.

Next, to give an example for the concept of Pareto domination, Figure 2-9 displays a two-objective optimization problem with six different solutions (A-F). The first objective of f_1 is to be maximized while the second objective of f_2 needs to be minimized, and these two objectives are both equally important. Since both objectives are equal in term of meaning to us, it is usually quite hard or sometimes impossible to find one solution which is best with respect to both objectives at the same time. Having said that, with the help of domination, we can decide which solution is better among any two given solutions in term of both objectives of f_1 and f_2 .

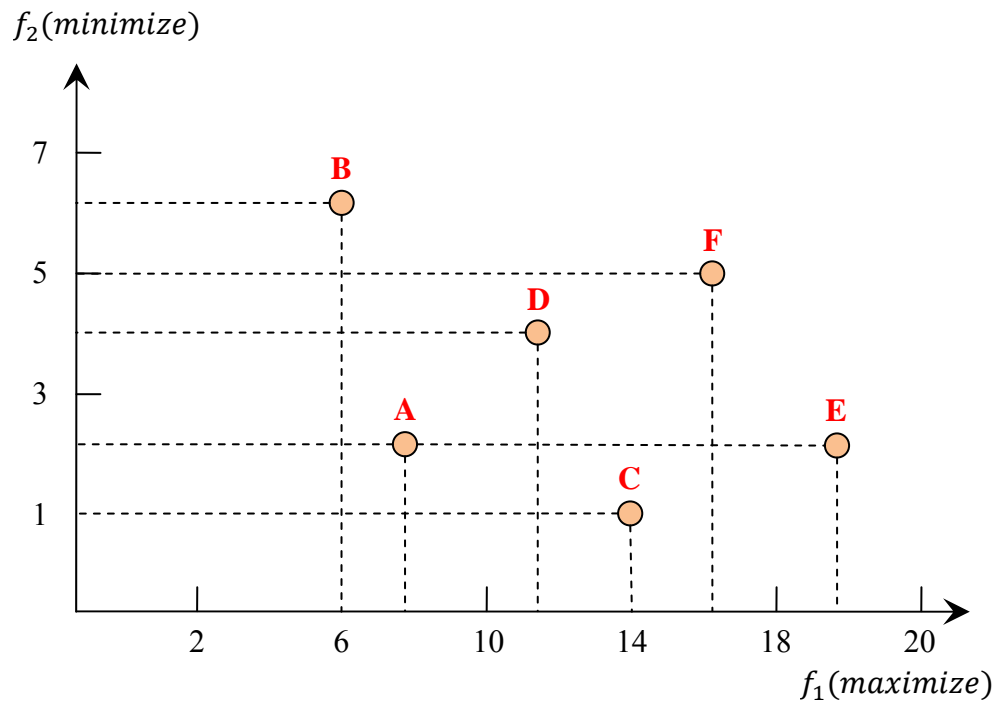


Figure 2-9: a two-objective optimization problem with six different solutions demonstrating concept of Pareto domination.

Let begin with comparing between solutions A and B. As can be seen, solution A is better than solution B in both objective function f_1 and objective function f_2 so that we can claim that solution A dominates solution B. Next, considering solutions A and E, we find that solution E is better than solution A in the first objective f_1 and solution E is no worse than solution A in the second objective f_2 (actually they are equal to each other on f_2). As a result, we can also say that solution E dominates solution A for the same reason as the first claim. For the last instance of comparing between solutions A and D, we observe that solution D is better than solution A in objective function 1, but solution A is better than solution D in objective function 2. For this reason, solutions A and D do not dominate each other. Moving to consider the relations between solution D and the others in Figure 2-10, the blue rectangle represents the region in objective space that is dominated by solution D, while the green rectangle encapsulates the solutions that dominate solution D. Last, all solutions that are lying outside those rectangles are incomparable to or indifferent to solution D.

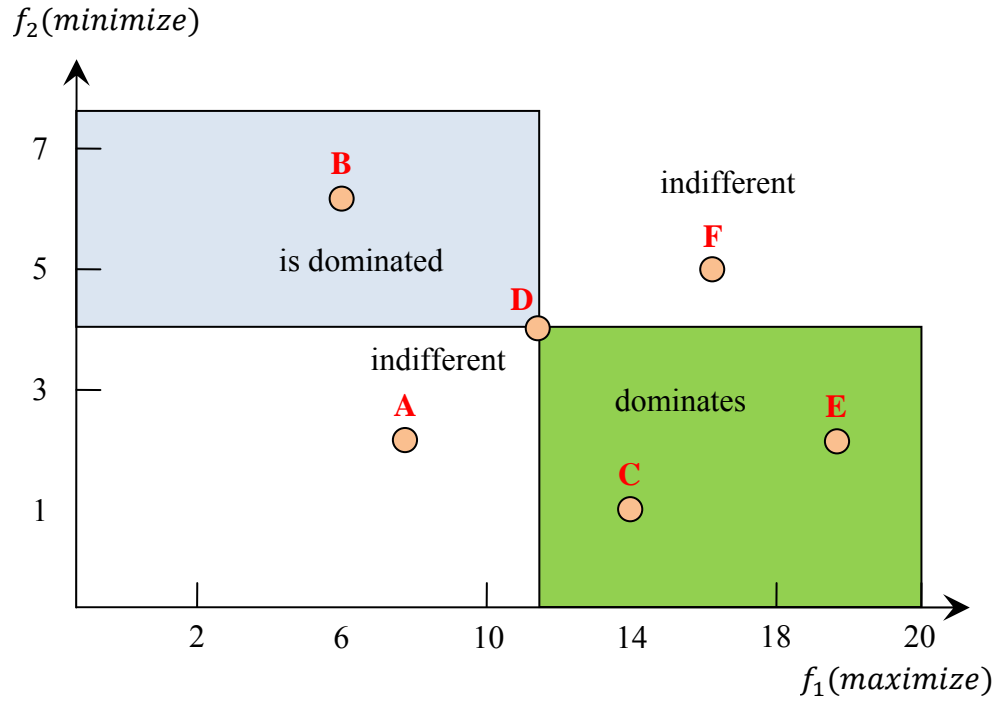


Figure 2-10: The possible relations of six solutions in objective space.

The main task of a multi-objective optimization algorithm is to identify solutions in the Pareto optimal set, but sometimes in many multi-objective problems, it is nearly impossible to identify the entire Pareto optimal set as its size is too large or unbound (infinite set). Furthermore, in case that proof of solution optimality is computationally impracticable, for instance, combinatorial optimization problems, a practical approach to identify the solutions of such a multi-objective problems is done by investigating a set of solutions (also known as the *best-known Pareto set*) that represent the Pareto optimal set as close as possible. In summary, a multi-objective optimization approach should achieve the three conflicting goals as follows [53, 111]:

1. The best-known Pareto front should be as close as possible to the true Pareto front. Ideally, the best-known Pareto set should be a subset of the Pareto optimal set. This can be done by intensifying the search on a particular region of the Pareto front.

2. Solutions in the best-known Pareto set should be uniformly distributed and diverse over of the Pareto front in order to provide the decision-marker a true picture of trade-offs. This demands the search effort to be uniformly distributed over the Pareto front.
3. The best-know Pareto front should captivate the whole scope of the Pareto front. This requires investigating solutions at the extreme ends of the objective function space and extending the Pareto front at both ends to explore new extreme solutions.

2.2.3 EA Approaches to Multi-objective Optimization

There have been many approaches for using Evolutionary Algorithms to solve multi-objective optimization problems. The first attempt in 1984 of using a genetic algorithm (GA) for multi-objective optimization was Schaffer's modification to a single-objective normal GA, the Vector-Evaluated Genetic algorithm (VEGA) [87]. However, after long runs the population of VEGA in the last generation tends to converge to a single optimal solution. After this study, Goldberg suggested the use of the concept of domination to formulate the fitness function using in multi-objective evolutionary algorithm (MOEA) in his book in 1989 [43]. His suggestion provided the inspiration for a lot of researchers to develop a number of MOEAs. Basically we can classify MOEA into two approaches: Non-elitist and Elitist Approaches. Some of the well-known algorithms of both approaches are given brief descriptions below.

Let us begin with non-elitist approaches. The first algorithm, Fonseca and Fleming's multi-objective GA (MOGA) [38] introduced in 1993, works by assigning a rank to each solution equal to the number of individuals of the current population that it dominates, plus one, and the non-dominated individuals are given the lowest rank of 1. After that, all individuals in the current population are sorted based on their ranks, and

the fitness of each individual is calculated by interpolating from the best rank to the worst one. Next, Srinivas and Deb's non-dominated sorting GA (NSGA) [92] presented in 1994 works by ranking all individuals in the current population using Pareto dominance, and the fitness assignment is computed based on dividing the population into a number of "fronts" of equal domination. For the last algorithm of non-elitist approaches, Horn et al. proposed NPGA [48] in 1994. This algorithm uses a modified version of tournament selection based on Pareto dominance, which operates by comparing two solutions on two conditions: the relation of domination between each other and the number of similar individuals already created in the new population.

With a drawback that non elitist MOEAs can potentially lose good solutions, the elitist strategy was applied to MOEA to prevent this problem. The MOEAs that mainly use the elitist concept are the Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) [23], the strength Pareto evolutionary algorithm (SPEA-2) [112] and the Pareto archived evolutionary strategy (PAES) [52]. The first one, NSGA-II, was introduced in 2001, and it still uses the same idea of non-dominated fronts as in NSGA with a considerable improvement in computational complexity (see section 2.2.4). The SPEA-2 algorithm has several significant changes to improve the performance of its prior version of SPEA [113] such as: fixed size archive and fitness assignment considering both dominated and dominating information of individuals. The PAES algorithm uses a (1+1) evolutionary strategy and works with only mutation operators to create a single child from a single parent, which is compared to its parent to update the archive. Both SPEA2 and PAES algorithms use an archive with a predefined fixed size, and this archive is used for containing non-dominated solutions discovered during the search process. Table 2-1 provides summary notes from [53] of the well-known multi-objective algorithms described above with their advantages and disadvantages.

Algorithm	Fitness assignment	Diversity mechanism	Elitism	External population	Advantages	Disadvantages
VEGA [87]	Each subpopulation is evaluated with respect to a different objective	No	No	No	-First MOGA -Straightforward implementation	-Tend converge to the extreme of each objective
MOGA [38]	Pareto ranking	Fitness sharing by niching	No	No	-Simple extension of single objective GA	-Usually slow convergence -Problem related to niche size parameter
NSGA [92]	Ranking based on non-domination sorting	Fitness sharing by niching	No	No	-Fast convergence	-Problem related to niche size parameter
NPGA [48]	No fitness assignment, tournament selection	Niche count as tie breaker in tournament selection	No	No	-Very simple selection process with tournament selection	-Problem related to niche size parameter -Extra parameter for tournament selection
NSGA-II [23]	Ranking based on non-domination sorting	Crowding distance	Yes	No	-Single parameter (N) -Well tested -Efficient	-Crowding distance works in objective space only
SPEA [113]	Ranking based on the external archive of non-dominated solutions	Clustering to truncate external population	Yes	Yes	-Well tested -No parameter for clustering	-Complex clustering algorithm
SPEA-2 [71]	Strength of dominators	Density based on the k -th nearest neighbour	Yes	Yes	-Improved SPEA -Make sure extreme points are preserved	-Computationally expensive fitness and density calculation
PAES [52]	Pareto dominance is used to replace a parent if child dominates	Call-based density as tie breaker between child and parent	Yes	Yes	-Random mutation hill-climbing strategy -Easy to implement -Computationally efficient	-Not a population based approach -Performance depends on cell sizes

Table 2-1: Summary notes of the well-known multi-objective algorithms

Due to the availability of open source code and adaptability to combine with our Genetic Programming (GP) system, the concept of Elitist Non-Dominated Sorting Algorithm (NSGA-II) was applied to our GP algorithm to run all experiments in Chapter 4. As a result, in next section we will have a look at the NSGA-II algorithm in detail.

2.2.4 NSGA-II

2.2.4.1 An Overview

NSGA-II begins by combining between Q_t and parent population P_t together to form set of solutions R_t of size $2N$, and then the entire population R_t is sorting by a non-dominated sorting method. This ensures that a global non-domination check is done among the offspring and parent solutions. After the non-dominated sorting is finished, the new population is populated one by one by solutions of different non-dominated fronts, beginning with the best (first) non-dominated front, following by the second non-dominate front and continuing with the third non-dominated front and so on. Due to the limitation of size of the new population at N , often not all fronts from R_t of size $2N$ can be used, and all remaining unused fronts are simply discarded. In the process of filling in the new population with the last allowed front, it is possible that the number of solutions in the last front is more than the remaining slots in the new population as illustrated in Figure 2-11. This problem can be solved wisely by using a niching strategy to select the members of the last front, which reside in the least crowded regions in that front, rather than randomly disposing some members from the last front [22]. Pseudocode of the NSGA-II algorithm can be found in the next section.

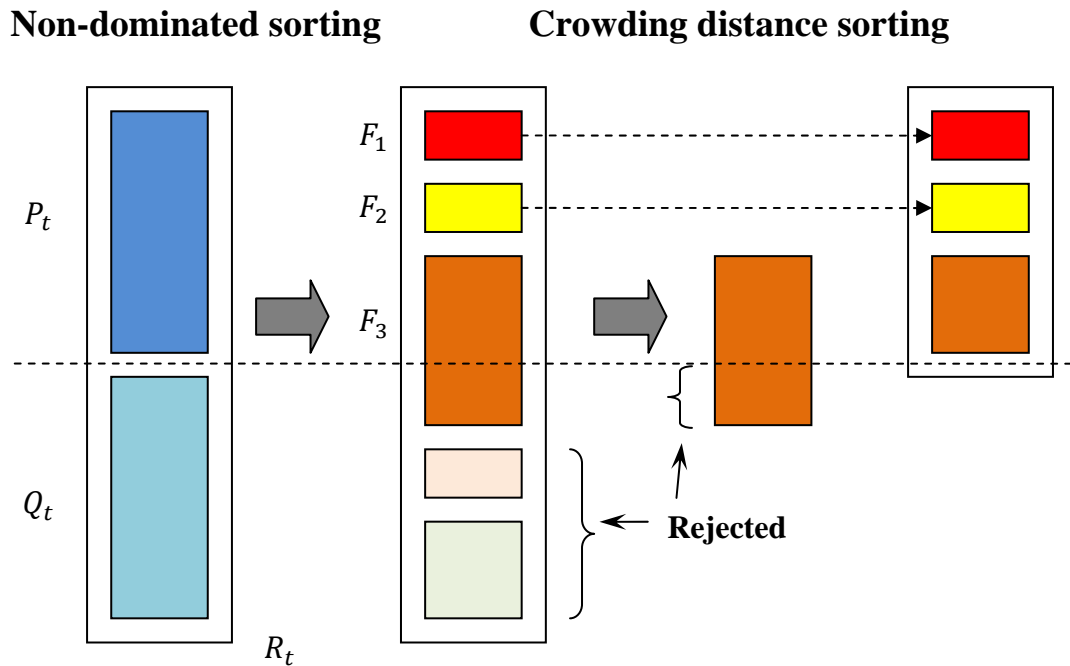


Figure 2-11: Process of filling the new population of the NSGA-II algorithm.

2.2.4.2 NSGA-II pseudocode

To begin with, an initial population P_0 is randomly created, and then it is sorted into different non-domination levels. After that, the non-domination level of each solution is assigned as its fitness, so the fitness of solutions is to be minimized as 1 is the best level. Following that, a binary crowded tournament selection operator, crossover operator and mutation operator are used to create an offspring population Q_0 of size N . Pseudocode of the NSGA-II procedure from [22] is given in Figure 2-12 with supplementary algorithms in Figure 2-13 and Figure 2-14.

Step1: Combine parent and offspring populations and create $R_t = P_t \cup Q_t$.

Perform a non-dominated sorting to R_t and identify different fronts: $\mathcal{F}_i, i = 1, 2, \dots, etc.$

Step2: Set new population $P_{t+1} = \emptyset$.

Set a counter $i = 1$.

Until $|P_{t+1}| + |\mathcal{F}_i| < N$, perform $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$ and $i = i + 1$.

Step3: Perform the Crowding-sort($\mathcal{F}_i, <_c$) procedure (described in section 2.2.4.3) and include the most widely spread $(N - |P_{t+1}|)$ solutions by using the crowding distance values in the sorted \mathcal{F}_i to P_{t+1} .

Step4: Create offspring population Q_{t+1} from P_{t+1} by using the crowded tournament selection (see section 2.2.4.4), crossover and mutation operators.

Figure 2-12: A pseudocode of NSGA-II.

2.2.4.3 Crowding Distance Assignment Procedure

Step1: Call the number of solutions in \mathcal{F} as $l = |\mathcal{F}|$.

For each i in the set, first assign $d_i = 0$.

Step2: For each objective function $m = 1, 2, \dots, M$, sort the set in worse order of f_m or, find the sorted indices vector: $I_m = \text{sort}(f_m, >)$.

Step3: For $m = 1, 2, \dots, M$, assign a large distance to the boundary solutions, or $d_{I_1^m} = d_{I_l^m} = \infty$, and for all other solutions $j = 2$ to $(l - 1)$, assign:

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{\max} - f_m^{\min}}$$

Figure 2-13: A pseudocode of crowding distance assignment procedure:

Crowding-sort($\mathcal{F}, <_c$).

2.2.4.4 Crowding Tournament Selection Algorithm

BEGIN

Set $\text{current_member} = 1$;

WHILE ($\text{current_member} \leq \mu$) **DO**

BEGIN

1. Pick k individuals randomly, with or without replacement;
2. Select the best of these k using crowded tournament selection operator (described in section 2.2.4.5);
3. Denote this individual as i ;
4. set $\text{mating_pool}[\text{current_member}] = i$;
5. set $\text{current_member} = \text{current_member} + 1$;

END

END

Figure 2-14: A pseudocode for crowded tournament selection algorithm to
select μ parents into mating pool.

2.2.4.5 Crowding Tournament Selection Operator

The crowded comparison operator ($<_c$) requires that every solution i has two attributes: 1) a non-domination rank r_i in the population and 2) a local crowding distance (d_i) in the population, and it works by comparing two solutions and returning the winner of the tournament. The crowding distance d_i of a solution i is the average distance of two solutions on either side of solution i along each of the objectives, which is calculated by the algorithm in section 2.2.4.3. The definition of the crowded tournament selection operator is as follows.

Definition 2.2 Crowded Tournament Selection Operator: A solution i wins a tournament with another solution j if any of the following conditions are true:

1. If solution i has a better rank, that is, $r_i < r_j$.
2. If they have the same rank but solution i has a better crowding distance than solution j , that is, $r_i = r_j$ and $d_i > d_j$.

2.3 Technical Analysis

To trade in the financial markets, what we would like to know is when is the right time to buy securities and when is the right time to sell them to make profit to us. To forecast futures prices, technical analysis is a financial markets technique that claims the ability to forecast the future direction of security prices by studying historical market information such as: closing price and volume, and it relies on the belief that all news, fundamental factors and market psychology are reflected in the security price value. The main keys of technical analysis are about studying price movement and trends in financial markets and how to utilize that information in order to predict future prices. A

large number of different methods and technical trading tools, of which details are available over the internet accompanied by free online tutorials, are utilized in technical analysis. Nonetheless, all methods and tools share the common belief that price patterns and trends exist in markets and they can be identified and exploited. “It’s all about the trend” [89]. Please see section 1.2 for a review about the forecasting ability of technical analysis.

2.3.1 Identifying Trend Reversals

The core concept in technical analysis focuses on a *trend* based on the assumption that future prices will more probably continue to go in the direction of the current apparent trend than go in the reverse direction. *The objective of trading systems developed using technical indicators is to identify the current pervasive trend and to trade in that direction, until a trend reversal is anticipated* [67]. In the financial markets, security prices are driven by over-supply (price going down) and demand (prices going up). If demand increases, security prices are promoted and prices fall as supply increases. And, security prices move sideways when supply and demand are equal. Suppose that we are trading in a stock market in an upward trending environment, and we have bought stock at some price, and the stock price continues to increase. Obviously, the identification of possible trend reversals to downward trends is clearly important so that we can sell our current holding stock at the right time and make a profit before the stock price goes down. Concepts which are relevant to the task of identifying trend reversal include those of support and resistance. Support and resistance occur when the forces of supply and demand encounter each other [11, 95]. Identifying support and resistance levels can be done by using trend lines or using pivot point calculations (see [105] for more information about pivot point calculations).

A zone of support (or a floor) is a price level at which demand is thought to be strong enough to prevent the price from falling below, and the price is more likely to reverse at this level rather than break through it. As a result, a zone of support originates at prices where there is a concentration of demand. The following is the logic behind the scene for a zone of support. When the price falls towards support and turns cheaper, buyers become more likely to buy and sellers become less likely to sell. By the time the price arrives at the support level, it is believed that demand will overcome supply and prevent the price from falling below the support level as illustrated in Figure 2-15. The green line in Figure 2-15 is called the support line, and the area below this line is a zone of support.

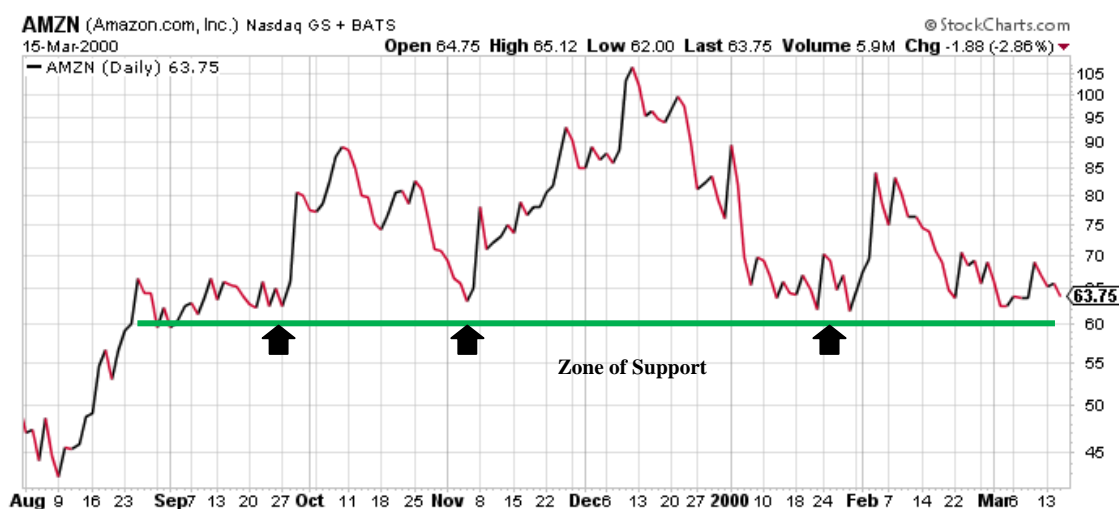


Figure 2-15: An illustration of support line and zone of support.

A zone of resistance (or the wall) is a price level at which supply is thought to be strong enough to prevent the price from rising above it, and the price is more likely to reverse at this level rather than break through it. For this reason, a zone of resistance arises when there is a concentration of supply. And, the logic for a zone of resistance is that as the price rises towards resistance, sellers get more likely to sell and buyers become less likely to buy. By the time the price get to the resistance level, it is believed that supply will overcome demand and prevent the price from rising above resistance as

illustrated in Figure 2-16. The red line in Figure 2-16 is called the resistance line, and the area above this line is a zone of resistance [95, 106].



Figure 2-16: An illustration of resistance line and zone of resistance.

To sum up, down-trends in price series tend to bounce off support zones, while up-trends tend to bounce off resistance zones. For this reason, a trading strategy using the resistance and support concept is that *stocks should be sold when their price reaches their resistance level, and stocks should be bought if their price bounces off their support level.*

It should be noted that in the case of a price breaking through a support level, that support level often turns into a new resistance level, and the opposite is also true. If a price breaks through a resistance level, then it will often become a support level at the same line in the future. In addition, when the zone of support or resistance is broken, resulting in a breakout happening, this indicates that something unusual has occurred such as important information flowing into the financial market and this has affected recent stock prices. As soon as the break-out occurs, technical analysts consider that the trend in price movement will accelerate and new support and resistance levels will be formed. Therefore, stocks increasing through a resistance level should be bought, while

those falling through a support level should be sold. Figure 2-17 provides an illustration of the concepts of support, resistance and breakout [11].

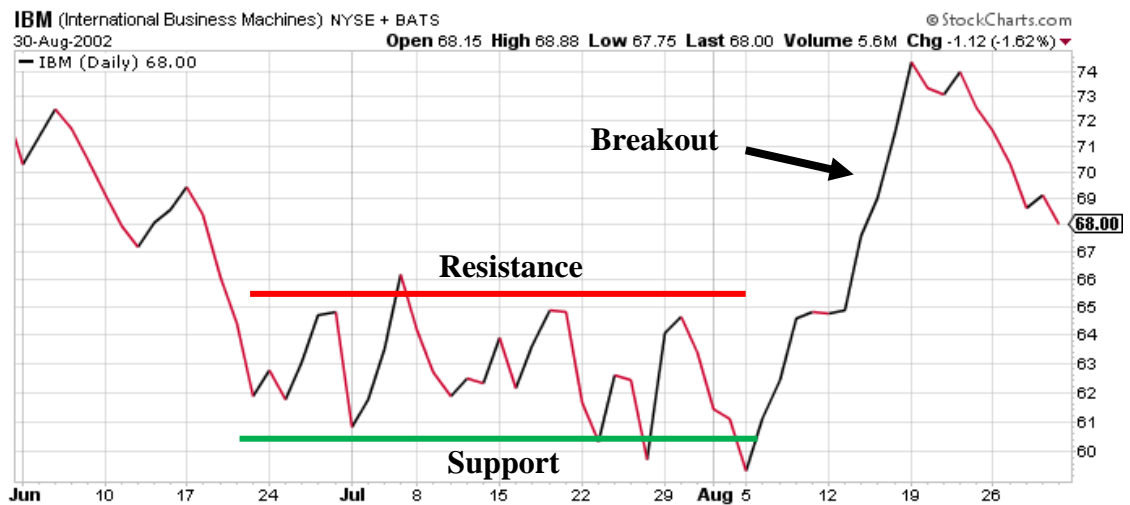


Figure 2-17: An illustration of the concepts of support, resistance and a breakout.

2.3.2 Technical Indicators

Technical indicators are essential components of technical analysis; they are typically mathematical transformations of price or volume, and are used to find out whether a security price is trending or not and also to indicate its price direction. Some indicators may use only the closing prices, while others may integrate volume or other financial data into their formulas. For example, moving averages (MA) are derived from simple calculations and are relatively easy to understand, whereas stochastic oscillators are more complicated and demand an extra effort to fully understand and appreciate. Regardless of the complexity of the formula, technical indicators offer a unique and different perspective on the strength and direction of the underlying price action. Details of technical indicators used in the experiments of this thesis are given as follows.

2.3.2.1 Moving Averages (MA)

The Moving Average (MA) is one of the most popular tools as it is easy to compute, use and understand, and it works by smoothing the price data making it easier to spot trends, and this is especially helpful in volatile markets where prices are much fluctuating. On the one hand, a moving average is a *low-pass filter* because it channels low-frequency signals (trends), but tones down high-frequency price fluctuations. To illustrate, Figure 2-18 displays the weekly price of S&P500 for the period of three years from 2008 to 2010 with its simple moving averages for 3 weeks and 12 weeks. It is noticeable that the longer moving average is much less volatile than either the shorter moving average or the actual weekly value [11, 93].



Figure 2-18: Weekly price of S&P500 from 2008 to 2010, with 3-week (blue) and 12-week moving average (red).

A simple version of moving average is computed by calculating the average price of a stock over a specified number of periods. Though we can calculate moving average of opening, high or low price, most moving averages are based on closing prices. To give an example of how to calculate moving average, Table 2-2 displays calculations of 5-day moving averages on three days. The first day's 5-day moving average is calculated by summing the closing prices for the last 5 days and then dividing the sum by 5 - the number of days.

Daily Closing Price	10, 11, 12, 13, 14, 15, 16
First day of 5-day MA	$(10 + 11 + 12 + 13 + 14) / 5 = 12$
Second day of 5-day MA	$(11 + 12 + 13 + 14 + 15) / 5 = 13$
Third day of 5-day MA	$(12 + 13 + 14 + 15 + 16) / 5 = 14$

Table 2-2: Calculations of 5-day moving averages on three days.

Continuing to calculate the 5-day moving average for the second day, if the next closing price is 15, then the second day's 5-day moving average drops the first data point (10) and adds the new data point (15). The third day's 5-day moving average continues by discarding the first data point (11) and adding the new data point (16). The calculation is repeated and so on in the data series for each day, and then the averages are joined together to form a smooth curving line – the moving average line.

2.3.2.2 Trend Lines

Trend lines are one of the most commonly used and important tools for the technical analyst, and they are constructed by drawing a straight line between two or more price points and then extending into the future to form a resistance or support line. There are basically two types of trend lines: the Upper Resistance Trend Line (URTL) and the Lower Resistance Trend Line (LTRL). The URTL is formed by connecting two local maxima by a straight line, while the LTRL is built by joining two local minima. Those result in resistance levels and support levels respectively. For instance, four trend lines, two resistance lines (red lines) and two support lines (green lines), have been identified in Figure 2-19 for the weekly price of S&P500 from 2008 to 2010. Moreover, as can be seen from the figure, index prices tend to oscillate in the channel formed by corresponding resistance and support lines [96, 107].



Figure 2-19: Weekly price of S&P500 from 2008 to 2010, with example of four trend lines: two resistances and two supports.

2.3.2.3 Rate of Change

The Rate of Change (ROC), also named as simply Momentum, is a centred oscillator that fluctuates above and below zero, and it simply measures the percent change in price from one period to the next. Basically, it shows the different between today's closing price and the closing price N period ago. The formula for ROC is given below.

$$ROC = \frac{\text{Today's close} - \text{Close } N \text{ periods ago}}{\text{Close } N \text{ periods ago}} \times 100$$

eq. 2-1

The ROC plot normally forms a graph that oscillates around the zero line fluctuating from positive to negative values, and it is used to indicate overbought or oversold extremes in the financial market. In general, prices are going up when the ROC stays positive, and conversely prices are decreasing when the ROC is negative. Normally most technical analysts use two ROC indicators: one for short-term ROC and another for long-term ROC. For example, Figure 2-20 presents the weekly price of S&P500 from 2008 to 2010 with its ROC of 3 weeks and ROC of 12 weeks, and the downtrend of price from June 2008 to March 2009 is indicated by the 12-week ROC mostly staying in negative territory [94, 104].

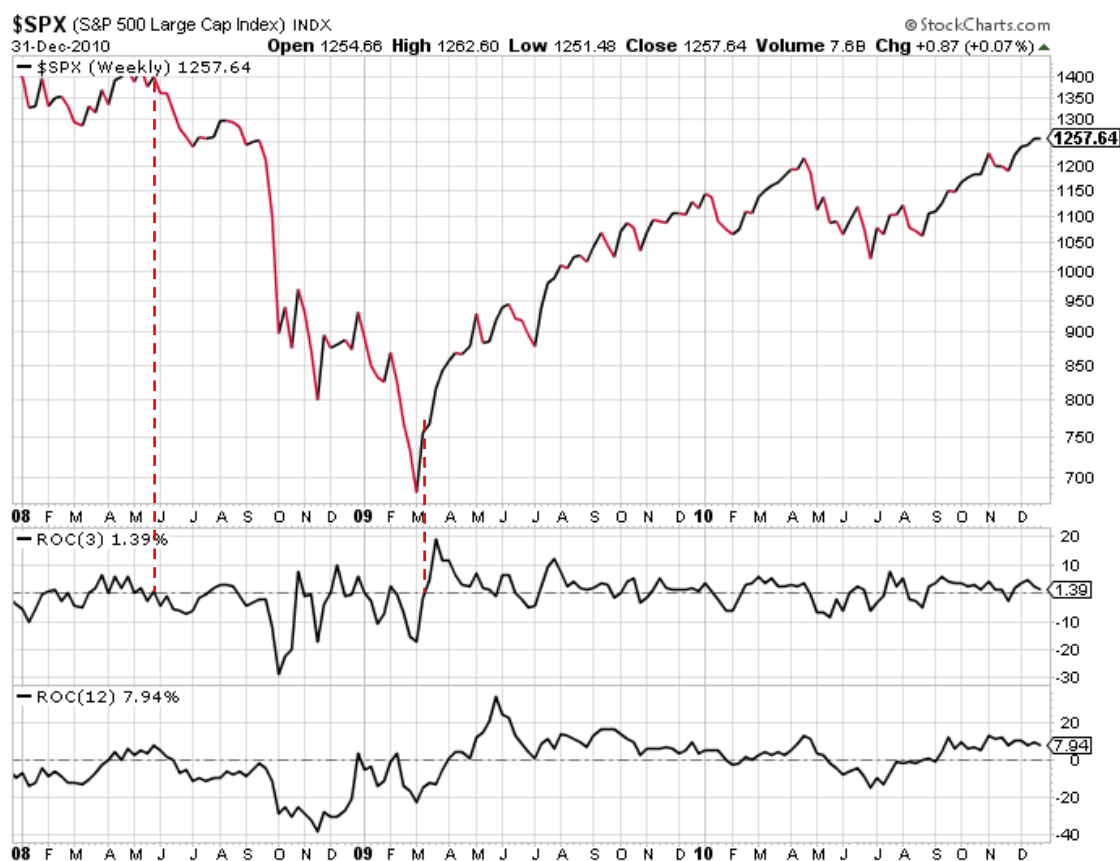


Figure 2-20: Weekly price of S&P500 from 2008 to 2010, with 3-week ROC and 12-week ROC.

2.3.2.4 Volume

Volume is simply the number of shares of a stock that have been traded over a given period of time, and it is used to confirm trends and measure the strength of a trend. It should be expressly stated that “*The higher the volume of the share, the more active the trade in that share.*” Fundamentally, both price and volume information should be combined together to validate the market strength. Technical analysts consider market state being strong if both price and volume are rising. Moreover, volume should be considered along with the trend, as volume should increase when prices are moving in an upward trend (and vice versa), and in the case that a period of low volume is happening, this may indicate uncertainty and possibly indicating trend reversal. Presenting an example for volume analysis in Figure 2-21, we can notice that the strong

increase in volume during the start of 2009 indicates that prices began making new highs as the price of the index increased, with more and more buyers (buying pressure) jumped into the market.

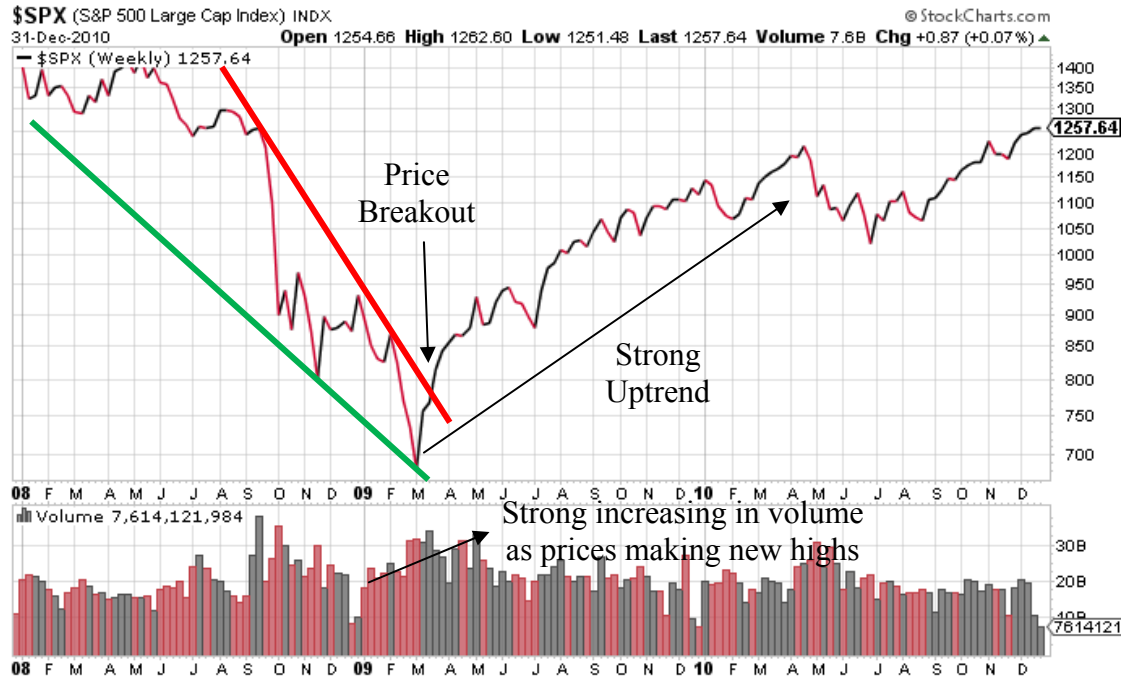


Figure 2-21: Weekly price of S&P500 from 2008 to 2010 showing a strong uptrend being confirmed by a strong increase in volume.

2.4 EA and MOEA in Computational Finance

2.4.1 Related works of Evolutionary Algorithms with Single-Objective Approach in Financial Market

There were a number of previous attempts using Evolutionary Algorithms (EA), including GP, for discovering technical trading rules, and some of them are chosen to be reviewed briefly in this section. They have almost exclusively used a single objective approach and can be classified into two main groups: the profit maximizing approach, and the risk adjustment approach. The profit maximizing approach focuses only on return, without concerning the associated risk [2, 6-8, 28, 35, 63, 79], whereas the risk adjustment approach includes a risk-adjustment in the fitness evaluation process [10, 17,

29, 40, 64, 65, 68, 69]. Basically these works generated trading rules for either (a) global market indices (e.g. S&P500, Dow Jones), (b) foreign exchange markets or (c) stock(s) of individual companies. In brief, as we have indicated, the first significant attempt to discover trading rules was by Allen and Karjalainen [2], which we will refer to hereafter as AK. In this study, GP was used with a single objective, which was to maximize excess return over B&H, to acquire technical trading rules for daily trading on the S&P 500 index from 1928 to 1995. Although Allen and Karjalainen's study did not show consistent excess return compared with simple B&H, it provided inspiration for applying EAs in financial markets. Allen and Karjalainen's fitness function did not consider any risk adjustment, so Neely [68] modified Allen and Karjalainen's work by including four risk-adjustment methods, such as the Sharpe ratio and the X^* measure in his single objective fitness function. However, this modification still was not enough to outperform B&H. After the previous two studies, Becker and Seshadri [6-8] succeeded in finding GP-evolved technical trading rules, which could outperform B&H. Becker and Seshadri made plenty of changes to the method used by Allen and Karjalainen, including the use of monthly data instead of daily data, reducing the GP operator set, and increasing the number of derived technical indicators among the GP operands. In other work that has attempted to look specifically at outperforming B&H, Potvin et al [79] showed that GP trading rules can be generally beneficial in falling or stable markets – but this is not particularly impressive, since B&H is naturally poor in such markets. In the same year, Fransworth et al. [35] also demonstrated that GP can be used to identify predictable patterns in financial asset prices with transaction penalty included during the training process only, but again their results did not outperform the buy-and-hold strategy. After that, Mallick et al. [63] used GP to generate trading rules on the thirty component stocks of the Dow Jones Industrial Average (DJIA), and they evaluated their GP performance against the simple buy and hold approach and the

popularly used MACD technical indicator. Their statistical results revealed that the GP based trading rules generated positive excess returns under all market conditions (whether rising or falling). It was also found that, in general, GP based trading rules offered greater returns over the simple buy and hold approach than the MACD trading signal. Following this, in a study of Esfahanipour et al. [28], GP was applied to automatically generate technical trading rules on individual stocks of 9 Iranian companies listed in different activity sectors of the Tehran Stock Exchange (TSE), and their results showed that their model could generate profitable trading rules in comparison with the buy and hold strategy, especially for companies having frequent trading volume in the market.

In another line of work, risk metrics such as the Sharpe ratio [90] have been included in rules (or in their evaluation). The Sharpe ratio modifies the return of a rule by considering its variation in return over the period. So a rule that achieves a very high return, but with much variation over the period, may be evaluated as the same quality as a rule that has a low return, but is very consistent over the period (e.g. producing a similar small positive return every month for a year). Such metrics typically reduce the fitness of rules that promote trading in volatile conditions, and therefore lead to rules more likely to be applied by investors. For instance, building on Fyfe et al. [40] (whose results were generally outperformed naïve B&H), Marney et al. [64] made headway by including risk metrics, however these attempts did not actually produce usable rules that compared well in comparison to B&H. Also, Cheng and Khai [17] used the modified Stirling ratio for evaluating the fitness of generated rules in foreign exchange between AUS and USD, and their trading rules did not lead to a reasonable rate of investment profit after the transaction cost. O'Neill, Brabazon, Ryan and Collins [69] examined the potential of Grammatical Evolution (GE) on financial trading to produce technical trading rules for the UK FTSE 100 stock index. The single-objective fitness function

used in this study incorporated the risk factor with the profit by subtracting it with the maximum cumulative loss (maximum drawdown) during the training period. The result of this study showed that GE was able to evolve technical trading rules which could beat the buy and hold strategy in four periods out of five. Another work of Brabazon and O'Neill [10] was an attempt to indicate useful technical trading rules for spot foreign-exchange markets. They used GE for trading on daily closing exchange rate data drawn from the London Market for the period 23/10/92 to 13/10/97, and the maximum drawdown was incorporated into their single-objective fitness function by subtracting the maximum drawdown during the training period from the return generated during that period. The result of this work indicated that GE could evolve trading rules which dominated the buy-and-hold benchmark in five of the six periods of US-DM, US-STG and US-Yen data set. Please note that in both studies above, they used fuzzy logic operators, which are f_and , f_or and f_not , along with standard arithmetic operators in their GE grammar, and the trading signals generated from their system were post-processed using fuzzy logic also. In the recent work of Esfahanipour and Mousavi [29], Conditional Sharpe Ratio was used as an risk measurement for trading on 10 individual stocks of the Tehran Stock Exchange (TSE), and the results of this study indicated superior performance over the buy-and-hold strategy especially in the case of a risk adjusted basis. A summary of key factors of all the works reviewed in this section is given in Table 2-3 with (a) blue highlighting for works on global market indices, (b) yellow highlighting for works on the foreign exchange market and (c) unhighlighted lines for works on stocks of individual companies. It is noticeable that the promising approaches seems to be to centred on considering stocks provided by individual companies.

Authors	Year	Outperformed B&H	Transaction Cost	Risk Measurement	Financial Market
Allen & Karjalainen	1999		✓	-	S&P500 Index
Neely	2001		✓	Sharpe Ratio, X^* , etc.	S&P500 Index
O'Neill et al.	2001	✓	✓	Maximum drawdown	UK FTSE 100 Stock Index
Becker & Seshadri	2003	✓	✓	-	S&P500 Index
Fransworth et al.	2004		Partly	-	S&P500 Index
Fyfe et al.	2005		✓	Sharpe Ratio	S&P Indices
Chang & Khai	2002		✓	Modified Stirling Ratio	AUS/USD Exchange Rate
Brabazon & O'Neill	2004	✓	✓	Maximum drawdown	London Market Exchange rate
Fyfe et al.	1999	✓	✓	-	UK Company-Land Securities
Marney et al.	2001		✓	Betas, Sharpe Ratio, X^* etc.	UK Company-Land Securities
Potvin et al.	2004			-	14 Canadian Companies
Mallick et al.	2008	✓	✓	-	30 DJIA Companies
Esfahanipour et al.	2009	✓	✓	-	9 Iranian Companies in Tehran Stock Exchange (TSE)
Esfahanipour & Mousavi	2011	✓	✓	Conditional Sharpe Ratio	10 Iranian Companies in Tehran Stock Exchange (TSE)

Table 2-3: Summary on key factors of related works in single-objective approach for both risk unadjusted and risk adjusted methods with (a) blue lines of works on global market indices, (b) yellow lines of works on foreign exchange market and (c) white lines of works on stocks of individual companies.

Considering such measures of risk, it is worth pointing out that the improvements made by Becker and Seshadri over the Allen and Karjalainen approach include one which effectively transforms the fitness function into a measure of risk – this is the performance consistency measure, which we discuss later. Overall, the incorporation of risk measures is clearly a promising thread of work in the automated trading context.

2.4.2 Related works of Evolutionary Algorithms with Multi-Objective Approach in Financial Market

With the rising use of EAs for multi-objective optimization (MOEAs), there have recently been several applications in the area of finance. Most of these have concentrated on investment portfolio optimization problems [14, 18, 20, 23, 25, 26, 58, 91, 98, 103], which represents the most popular application of MOEAs in this area [20] – this is essentially a problem of finding an ideal collection of companies/equities to constitute a trading portfolio. However there are three recent studies which used an MOEA to generate technical trading rules. In the first of these, Chiam, et al. [18] used an MOEA to generate technical trading rules with two objectives: to maximize total return and to minimize semi-variance as a risk factor, and they concluded that the higher return in the training data tended to correspond to larger volatility in the returns generated on the test data. In the second of these studies, Briza, et al. [12] used Multi-Objective Particle Swarm Optimization method to yield technical trading rules with two objectives: percent profit and Sharpe ratio [62] in 3 contiguous training and testing periods. Both the best and average performances of this study were able to outperform all 5 technical indicators in the training periods, and they also beat the buy-and-hold (B&H) strategy in two periods out of three. On test data, the performance of the best points among 30 Pareto fronts beat all indicators in both objective functions, but failed to beat the buy-and-hold strategy. For the last study, Wilson and Banzhaf [108] investigated the use of a linear genetic programming (LGP) system for automated foreign exchange trading from August 05/08/2008 to 05/08/2009 of four major currency pairs: CAD/USD, EUR/USD, GBP/USD and JPY/USD. Two single-objective fitness functions were tried out: the first one using the unmodified value of assets held subtracted by maximum drawdown, labelled as ‘*Moderate*’ version, and the other using the unmodified value simply divided by maximum drawdown, denoted as ‘*Conservative*’ version. Wilson and Banzhaf’s result showed that the conservative

fitness function generated higher profits than buy and hold or moderate version of fitness for most trend types, while moderate fitness seemed to perform better on the trend with large value changes with high volatility.

In the few studies so far using MOEA for technical trading rules, the results seem inconclusive, however it certainly seems worth more investigation. In Chapter 4, we therefore investigate a number of alternative ways of characterizing technical trading as a multi-objective task, engineering in the context of the GP method in [60, 61] that has its roots in [2, 6-8].

2.5 Discussion and Conclusion

In this chapter we have reviewed the fundamental concepts of evolutionary computing and evolutionary algorithms, particularly Genetic Programming (GP), which is the main tool explored in Chapters 3 and 4. We have seen evidence of many attempts by GP practitioners (in the related work section), in which GP has been used to make a profit in financial markets using evolutionary algorithms with both single and multi objective methodologies, with different degrees of success. This leads us to the question of whether we can use evolutionary algorithms to generate technical trading rules that are robust enough to generalize well to unseen data, with either single or multi objective approaches, and the related question of what are the principal factors (e.g. in configuration or experimental design) towards achieving such good generalization properties. It should be noted that the latter question is very challenging and related to one of the significant open issues in the GP world [78], about generalization in GP. Moreover, we have reviewed how technical analyses use technical indicators to capture the current apparent trends in order to make the right decisions about buying or selling securities in financial markets, and these technical indicators are utilized to form

technical trading rules. To conclude, the research explored in this thesis basically constitutes our attempts to address the research questions identified above.

Moving to a discussion about the most influential work relevant to this thesis, there are four key papers. The first one is the study of Allen and Karjalainen [2] in 1999, published in the *Journal of Financial Economics*, and it was the first and pioneering work in the field for evolving technical trading rules with GP. Although this study did not show consistent excess return compared with the standard benchmark of buy-and-hold, it provided inspiration for many later works in applying EAs for trading in financial markets. Following that, in 2001 Neely [68] modified Allen and Karjalainen's work by integrating risk measurement, such as the Sharpe ratio, in his single objective fitness function. However, this modification still was not enough to outperform B&H. These two main papers gave us an idea of how to use GP to evolve technical trading rules with fitness functions of both profit-driven and risk-adjusted measurement types. After the previous two studies, Becker and Seshadri's works [6, 8], the last two of the four most influential papers that motivated this thesis, claimed that they succeeded in finding GP-evolved technical trading rules, which could outperform B&H, with several changes to the method used by Allen and Karjalainen. After reviewing these works, we could see some room to improve and extend them, such as the use of a three-data-set methodology in model selection to improve the generalization level of the trading rules, and the use of multi-objective approaches instead of single objective configurations. Thereafter, we started by replicating this work and investigating the claims, and then we built on this baseline work by extensive investigation in various different trading environments.

3. Single Objective Algorithms for Evolved Technical Trading Rules

3.1 Introduction

There are many opportunities in the areas of financial markets for machine learning and optimization methods [11], and applications of Genetic Programming (GP) [3, 4, 54], as well as other areas of evolutionary computation, are now common in this area [2, 15-17, 28, 29, 35, 40, 63, 64, 79]. GP is particularly popular in this field, with many studies reporting a variety of different GP applications in finance [2, 15-17, 28, 29, 35, 40, 63, 64, 79]. Our focus is the area known commonly as ‘financial trading’, but more strictly termed as technical analysis [59, 67, 81, 85]. Technical analysis is the name given to the general area of study in which one attempts to forecast the future direction of equity prices, guided either largely or entirely by the patterns that are revealed (or perceived) from historical equity price data. Technical analysis therefore relies deeply on the notion that discernible or discoverable patterns and trends exist in equity price fluctuations over time, and that these can be exploited to predict price movements in the near future. Please see section 2.3 for more details of technical analysis.

Simple, commonly used, and reasonably successful approaches to technical analysis use measures such as moving averages (the mean price for a given stock or index over a given recent period), relative strength indicators (a function of the ratio of recent upward movements to recent downward movements), and several others. A standard approach to engineering GP applications in this area is for GP trees to express trading rules, which combine technical indicator ‘primitives’ with other mathematical operations. Such a rule constitutes a *signal*, which may be interpreted, for example, as a recommendation to buy if the signal is above a threshold. The seminal attempts to use GP in this way were by Chen and Yeh [16] and by Allen and Karjalainen [2]. These

initial studies, and many others that followed, tend to report that GP is able to find rules that are indeed profitable on unseen future data. But, rarely in such work is it also shown that such rules provide greater returns than a standard “*Buy-and-Hold*” (B&H) trading approach. The B&H strategy is: for a given trading period, buy the stock at the beginning of the period, and sell it at the end – hence, always a good strategy in an upwardly moving market, and far simpler than using technical indicators. Similarly, where rules are evaluated in a downward moving market, it is not generally clear if the rules developed by GP can outperform the simple strategy of investing in Treasury bills or the equivalent (i.e. a safe investment with a small but guaranteed return).

Nevertheless, a small amount of research in this area seems able to find rules that outperform buy-and-hold, especially when incorporating risk factors into the fitness model of the rules. In particular, GP-evolved technical trading rules with such success have been reported in Becker and Seshadri [6-8], who adopted the overall approach of Allen and Karjalainen [2] (who did not outperform buy-and-hold), and made several alterations. One of Becker and Seshadri’s alterations was to adopt monthly trading rather than (as in Allen and Karjalainen) daily trading. That is, in [6], rules assume that trades will only be made (if at all) on the first day of the month, and hence deal with a less volatile view of the market. Other alternatives involved modifying the fitness function to reward consistency in performance, as well as simplicity of rules (and hence avoid overfitting). It is intuitively reasonable to suggest that this was an important feature of Becker and Seshadri’s work, in the sense that outperformance of buy-and-hold may not have been achieved without this modification. However, that hypothesis has not yet been tested. In this chapter we test this hypothesis by performing tests on a modified version of Becker and Seshadri’s approach, and explore each of monthly weekly and daily trading. Also, we provide full details to enable replication of [6] as well as showing that a modified experimental setup leads to more robust outcomes.

Basically this chapter is based on the works from [60] and [61] with more extensions and analysis. In common with [60, 61] and the other works cited that lead up to those, all of the approaches this chapter are single-objective approaches.

The rest of this chapter is organized as follows:

- Section 3.2 gives information on the overall GP approach (genome structure, function and terminal sets) and the trading strategy model, and explains how to calculate the fitness objectives. It also gives details of GP operators and the initialization phase of the GP algorithm.
- Section 3.3 provides details of GP parameters used in the experiments, such as population size, mutation and crossover rates. Details for each data split of monthly, weekly and daily data are given in this section as well.
- Section 3.4 reports the results of the experiments on monthly, weekly and daily data for two types of rule-selection strategy, and according to two evaluation metrics: the mean excess return and how it compares to the buy and hold return during the (unseen) evaluation period, and the percentage by which the result outperformed buy and hold.
- Section 3.5 provides a basic analysis of the common structures found within the technical trading rules obtained from the experiments in section 3.4.
- The last section, Section 3.6, gives some conclusions on the comparison between the two rule-selection strategies, on our basic trading rule analysis, and concerning the best configuration for each of monthly, weekly and daily trading.

3.2 Evolving Robust Trading Rules

3.2.1 Overview

The approach we use is based on Becker and Seshadri's work [6, 8] which in turn was a modification of Allen and Karjalainen's work [2]. This approach uses standard GP, with a function set comprising arithmetic, Boolean and relational operators, while the terminal set comprises a collection of basic financial technical indicators, along with real and Boolean constants, and real-valued variables (such as *equity price*). An example of a rule specified by a GP tree (in fact a rule found in [6]) is in Figure 3-1. The details of technical indicators referred to in Figure 3-1 can be found in section 3.2.2.

If we are doing weekly trading (for example) then a rule such as that in Figure 3-1 is to be interpreted as follows. The rule essentially makes an assertion: "The 3-week moving average (MA-3) is less than the lower trend line (LRTL) and the 2-week moving average (MA-2) is less than the 10-week moving average (MA-10) and the lower trend line (LRTL) is greater than the second previous 3-week moving average maxima (MX-2)." In the context of a particular trading intervention, we evaluate this assertion and find that it is either true or false. If the assertion is true, then we generate a *buy signal*; if the assertion is false, then we generate a *sell signal*.

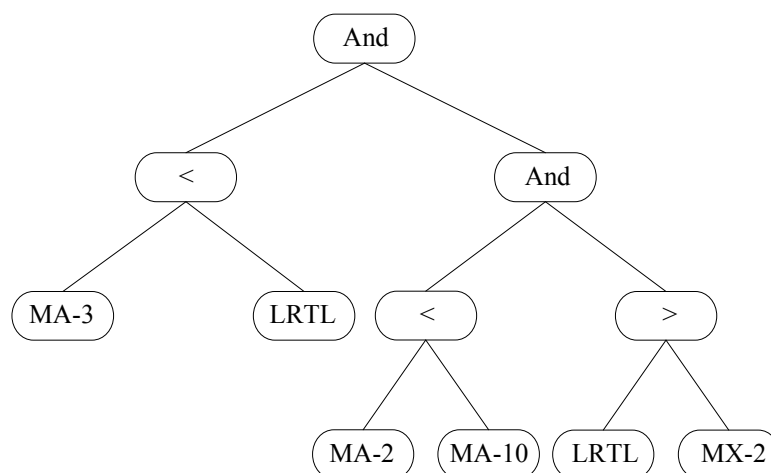


Figure 3-1: An example trading rule in the GP population.

Given a technical trading rule as in Figure 3-1, and given the signal generated by that rule, we then apply that signal in the financial market; the trading strategy for applying the signal is modelled as a simple automaton in Figure 3-2. There are two states in the model: ‘IN’ indicates the state *in the market* (earning the market rate of return) and ‘OUT’ indicates the state of being *out of the market* (earning the risk-free rate of return). The model specifies the position to be taken the following day, given the current market position and the current trading rule signal. This then corresponds to a trading action as follows. If we are currently in the ‘OUT’ state and the trading rule generates a buy signal, then we switch to the ‘IN’ state and make a *buy* decision; if we are currently in the ‘IN’ state and the trading rule generates a sell signal, then we switch to the ‘OUT’ state and make a *sell* decision; otherwise we preserve the current state and do nothing. Please note that in the case that the current day is the last day of the trading period, then the trading rule is forced to generate a sell signal. The overall procedure assumes a fixed amount is to be invested (e.g. \$1,000) whenever the decision is *buy*.

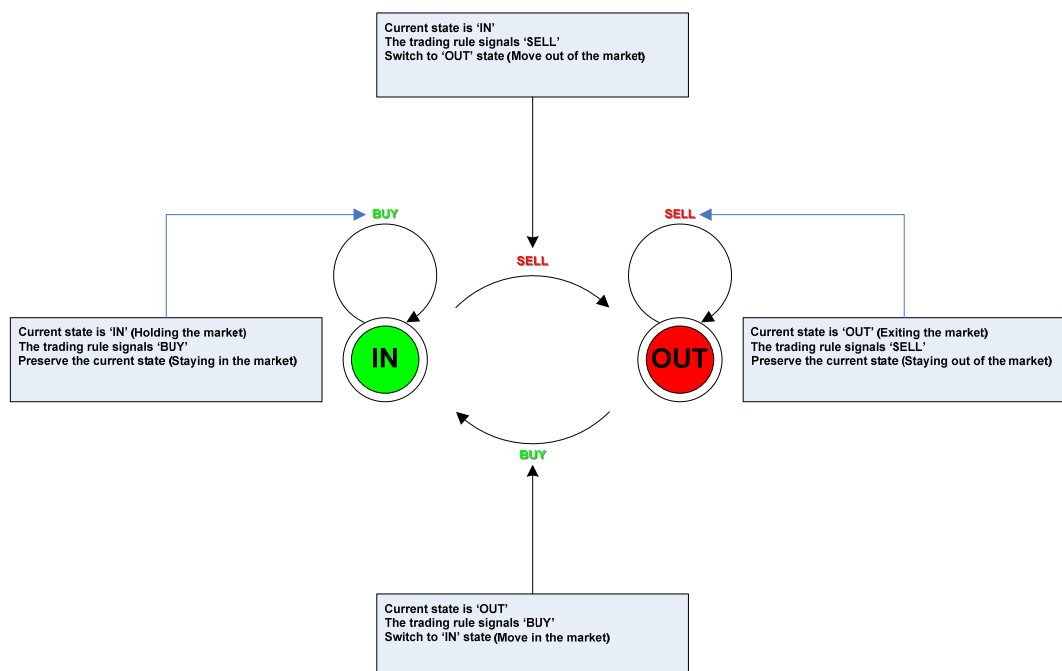


Figure 3-2: The trading strategy model of the technical trading rule.

The remaining subsections explain the approach in further detail, paying particular attention to the fitness function; this description includes several key details that were not explicit in [2, 6, 8], but which we have discovered essential in order to achieve appropriate replication and good performance. The approach we explain is that of Becker and Seshadri, and we note here the modifications they made to the AK approach. These were: the use of monthly data rather than daily data; a reduced function set, but a larger terminal set, with more indicators; the use of a complexity-penalizing factor in the fitness function to avoid over-fitting; modifying the fitness function to consider the number of periods with well-performing returns, rather than just the total return over the test period.

Finally, the data we use (as in [2, 6-8, 44, 60, 61]) is the Standard and Poors 500 (S&P500) index – a fixed set of 500 stocks which aggregate to daily price indicators (opening, closing, high, low). When considering weekly and monthly trading, the opening price (for example) for a week or a month is the opening price on the first day of that week or month.

3.2.2 Function and Terminal Sets Overview

The function set and the terminal set are shown in the table, where ‘unit’, depending on the experiment, is either day, week or month. We use strongly typed GP [66], to automatically enforce appropriate inputs for each operator.

Function Set	Boolean operators: <ul style="list-style-type: none">• And (&&)• Or ()• Not (!) Relational operators: <ul style="list-style-type: none">• greater than (>)
---------------------	---

	<ul style="list-style-type: none"> less than ($<$)
Terminal Set	<p>Technical Indicators:</p> <p><i>Prices:</i></p> <ul style="list-style-type: none"> Opening price for the current unit (OPRICE$<T>$) Closing price for the current unit (CPRICE$<T>$) High price for the current unit (HPRICE$<T>$) Low prices for the current unit (LPRICE$<T>$) Closing price for the previous unit (CPRICE$<T-1>$) <p><i>Volumes:</i></p> <ul style="list-style-type: none"> Volume for current unit (VOL$<T>$) Volume for previous unit (VOL$<T-1>$) <p><i>Moving Averages:</i></p> <ul style="list-style-type: none"> 2-unit moving averages (MA-2) 3-unit moving averages (MA-3) 6-unit moving averages (MA-6) 10-unit moving averages (MA-10) <p><i>Rate of Change Indicators:</i></p> <ul style="list-style-type: none"> 3-unit rate of change (ROC-3) 12-unit rate of change (ROC-12) <p><i>Price Resistance Indicators:</i></p> <ul style="list-style-type: none"> The two previous 3-unit moving average minima <ul style="list-style-type: none"> The first previous local minima (MN-1) The second previous local minima (MN-2) The two previous 3-unit moving average maxima <ul style="list-style-type: none"> The first previous local maxima (MX-1) The second previous local maxima (MX-2) <p><i>Trend Line Indicators:</i></p> <ul style="list-style-type: none"> A lower resistance line based on the slope of the two previous minima (LRTL) An upper resistance line based on the slope of the two previous maxima (UTRL)

Table 3-1: The function set and terminal set for multi-objective strongly typed GP.

Please note that ‘unit’ in the table can be: day for daily data, week for weekly data and month for monthly data. The n -unit moving average at time m is the mean of the

closing prices of the n units from m back to $m-(n-1)$. The n -unit rate of change indicator measured at time m is: $(c(m) - c(m-(n-1))) \times 100 / c(m-(n-1))$, where $c(m)$ indicates the closing price for time m . Previous maxima MX1 and MX2 are obtained by considering the 3-unit moving averages at each point in the previous units. Of the two highest values, the one closest in time to the current is MX1, and the other is MX2. The two previous minima are similarly defined. Finally, to identify trend line indicators, the two previous maxima are used to define a line in the obvious way, and the extrapolated value of that line from the current time becomes the upper trend line indicator; the lower trend line indicator is defined similarly by using the two previous minima. More details of how to calculate technical indicators can be found in section 2.3.

3.2.3 The Fitness Function

There were three fitness functions in the single-objective formulations previously used [2, 6-8], and there are two versions of fitness functions presented in this section. The details of each one are given in the following.

3.2.3.1 Excess Return (ER) – the excess of the return over that of the buy-and-hold strategy. This is:

$$ER = r - r_{bh}$$

eq. 3-1

where r is the continuously compound return for a trading rule, and r_{bh} is the corresponding return from buy and hold.

3.2.3.2 Market Return (MR) – the total return on an investment in the financial market. It can be calculated as:

$$MR = e^r \times investment$$

eq. 3-2

where *investment* is the amount of money to invest (e.g. \$1,000).

To calculate r we use [2, 6-8]:

$$r = \sum_{t=1}^T r_t I_b(t) + \sum_{t=1}^T r_f I_s(t) + n \ln \left(\frac{1-c}{1+c} \right)$$

eq. 3-3

where:

$$r_t = \log P_t - \log P_{t-1}$$

eq. 3-4

which indicates the continuously compounded return, where P_t is the price at time t . $I_b(t)$ is 1 if the rule indicates *buy* at time t , and 0 otherwise. Similarly defined is the sell

signal, $I_s(t)$. The first component of r hence calculates return on investment over the times when the investor is in the market. The second component, $r_f(t)$ indicates the risk-free return, which is taken for any particular day t from US Treasury bill data (available from <http://research.stlouisfed.org/fred/data/irates/tb3ms>). Hence, the second component represents time out of market, assuming that the investor's funds are earning a standard risk-free interest. Finally, the third component corrects for transaction costs, where c denotes the one-way transaction cost. The cost of a single transaction is assumed to be 0.5% – e.g. \$5 for a transaction of volume \$1,000. The number of transactions during the period is n . This component estimates the compounded loss from the cost of transactions.

In eq. 3-1, the second main part of the fitness function, r_{bh} , is calculated as:

$$r_{bh} = \sum r_t + \ln\left(\frac{1-c}{1+c}\right) \quad \text{eq. 3-5}$$

where r_t is as indicated above, calculating the return of buying at the first day and selling at the last day of the period. To calculate *Total Buy-and-Hold Return (BHR)*, we use the following formula:

$$BHR = e^{r_{bh}} \times \text{investment} \quad \text{eq. 3-6}$$

3.2.3.3 Performance Consistency (PC) – this is what we call the other fitness function used in [6, 8].

Steps to calculate the PC fitness value are the following:

1. MR is calculated for each successive period of K units covering the entire test period.
2. BHR is calculated for each successive period.

3. *Risk-free return* for each successive period is computed by choosing the maximum of risk-free interest ($max-r_f$) in the current period and then multiply $max-r_f$ by a number of units in that current period.
4. For each successive period, if the *MR* of the current period is greater or equal to both the *risk-free return* and the *BHR* of this current period, then consider it to be a well-performing return.
5. The fitness value is the number of periods with well-performing returns. It can be a number from 0 to X , where there are X periods covering the entire test period. Please see Figure 3-3 for an example.



Figure 3-3: An example of monthly trading (unit = month) of 5 successive periods with total length at 60 months (5 years) for performance consistency (*PC*) at $K = 12$ months and $X = 5$ periods.

The remaining two fitness functions we describe have risk measurements integrated within them, and are called ‘risk-adjusted’ fitness functions. In this context, a risk-adjusted fitness measure is one which, in some way, reduces the reward for unstable behaviour, such as high variability in the return during the investment period over which the rule is evaluated. The *PC* fitness function, already described, can be considered a risk-adjusted measure. However the more traditional ways to incorporate risk adjustments are as described next.

3.2.3.4 Sharpe Ratio (*SHARO*) – this ratio incorporates a risk factor measured by the standard deviation of the returns. This is:

$$SHARO = \frac{\text{excess return}}{\text{standard deviation of return}}$$

eq. 3-7

where *excess return* is the rate of return for a portfolio subtracted by the risk-free rate.

3.2.3.5 *Modified Stirling Ratio (MSTLRO)* – ‘modified drawdown’ is used as a risk assessment in this risk-adjusted fitness function. It is defined by:

$$MSTLRO = \frac{\text{return}}{1 + \text{modified drawdown}}$$

eq. 3-8

where *modified drawdown* is max(drawdown, 2% of current position), and return and drawdown are measured as a percentage of traded assets [24].

The *excess return (ER)*, calculated as described, was originally the objective function in [2], while *market return (MR)* was used in [6, 8]. However, improvements in [6, 8] arose from two adjustments. One of these is an adjustment to fitness according to the size of the tree. Given an actual fitness value f , the adjusted fitness value *adjusted- f* modified by the ‘complexity-penalizing factor’ can be computed as in the following equation:

$$\text{adjusted} - f = \frac{5f}{\max(5, \text{actual} - \text{depth})}$$

eq. 3-9

where *actual-depth* is the depth of the tree being evaluated, and the constant 5 is a ‘desired’ depth. Another adjustment is the use of the *performance consistency (PC)* approach that we have described, instead of using *market return MR*. Evidence in [6, 8] indicated that this fitness function led to more consistent results.

Putting all these together, the most successful single objective approach arising from [2, 6, 8] calculates f , the PC-based fitness (i.e. a number from 0 to X , where there were X

periods covering the test data), and adjusts it for tree complexity, returning $5f/\max(5, \text{actual-depth})$ as the fitness of a rule.

In Becker and Seshadri's work, the employment of the PC term clearly results in improved performance. However they only report on the use of 12-month looking- up periods. We experiment with four different lengths for the "*PC period*", for each trading situation, namely 6, 12, 18 and 24 months periods for monthly trading. These are referred to later as PC_LK6, PC_LK12, PC_LK18 and PC_LK24. In the case of weekly and daily trading the PC Periods have been cut down to two periods: 12-unit for the representation of short period and 24-unit for long period; these are 12 and 24 weeks for weekly trading, and 12 and 24 days for daily trading.

3.2.4 Operators and Initialization

3.2.4.1 *Subtree-swap crossover* – we used standard subtree-swap crossover [54], which operates by interchanging the subtrees beginning at two randomly selected nodes in the first and second parent as illustrated in Figure 3-4, and it results in two new trees as their offspring.

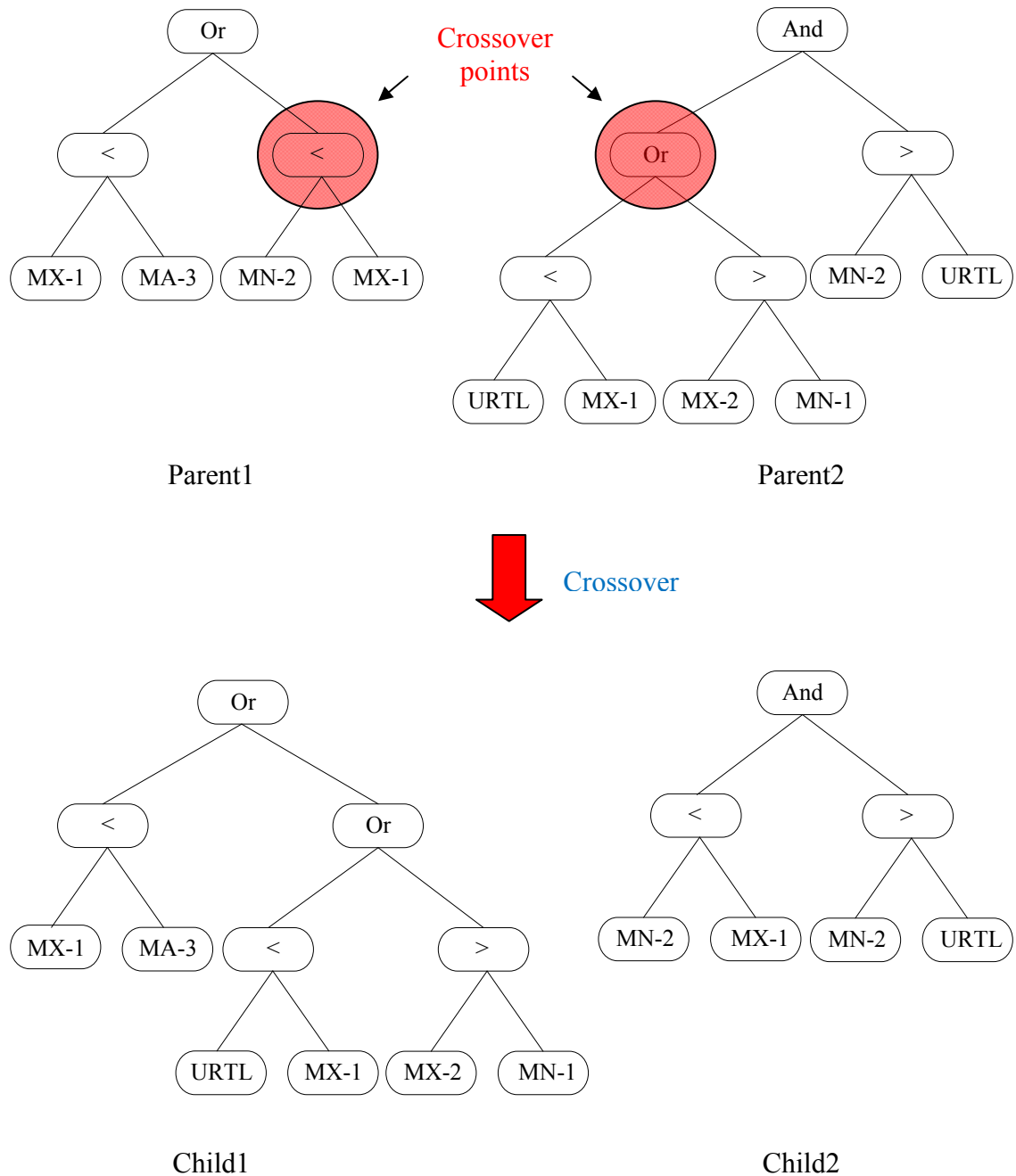


Figure 3-4: GP subtree-swap crossover. Two crossover points in both parents are indicated by red circles.

For mutation, we used the mutation operators described by Angeline [3] and Banzhaf et al. [4] together as follows.

3.2.4.2 *Subtree Mutation* – randomly select a target subtree and a source subtree and exchange between them (see Figure 3-5).

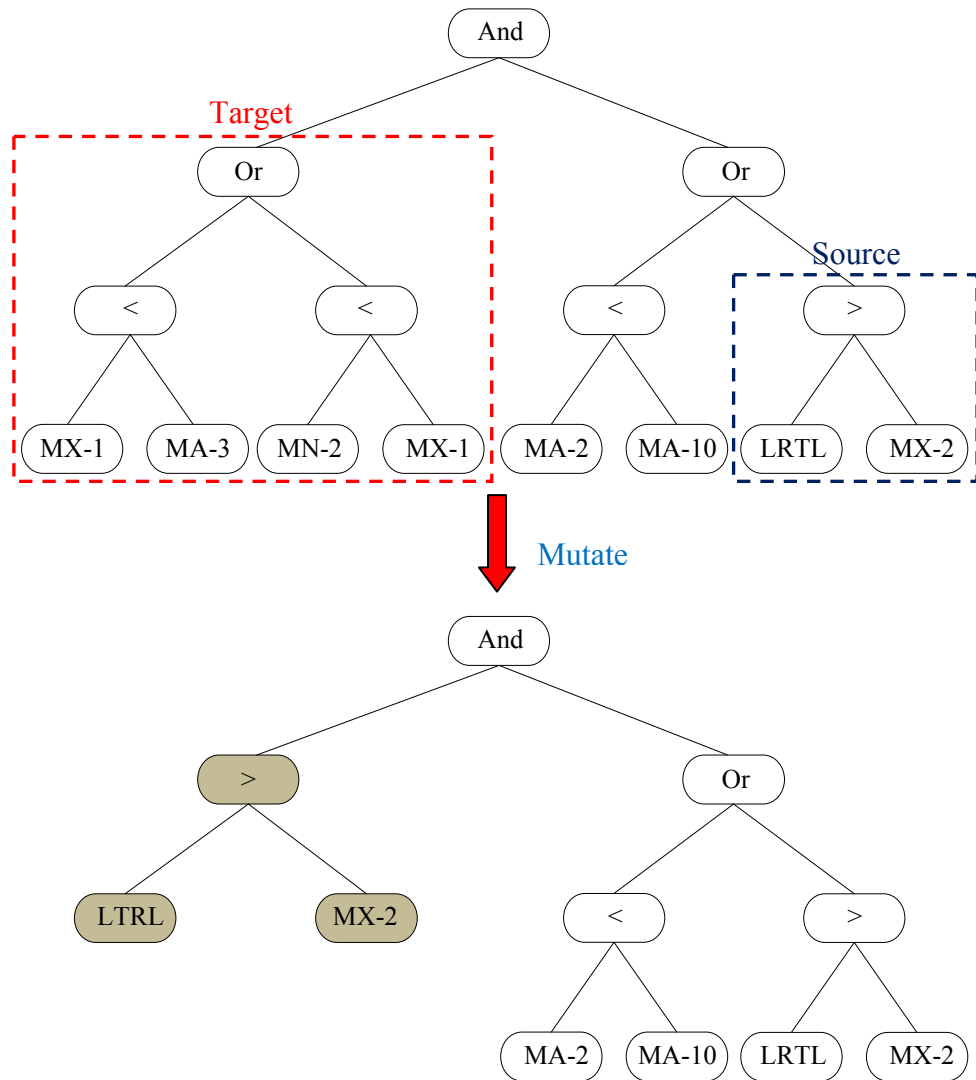


Figure 3-5: GP Subtree Mutation

3.2.4.3 *Point Mutation* – randomly select a single target node and exchange it with a random source node of the same class (see Figure 3-6).

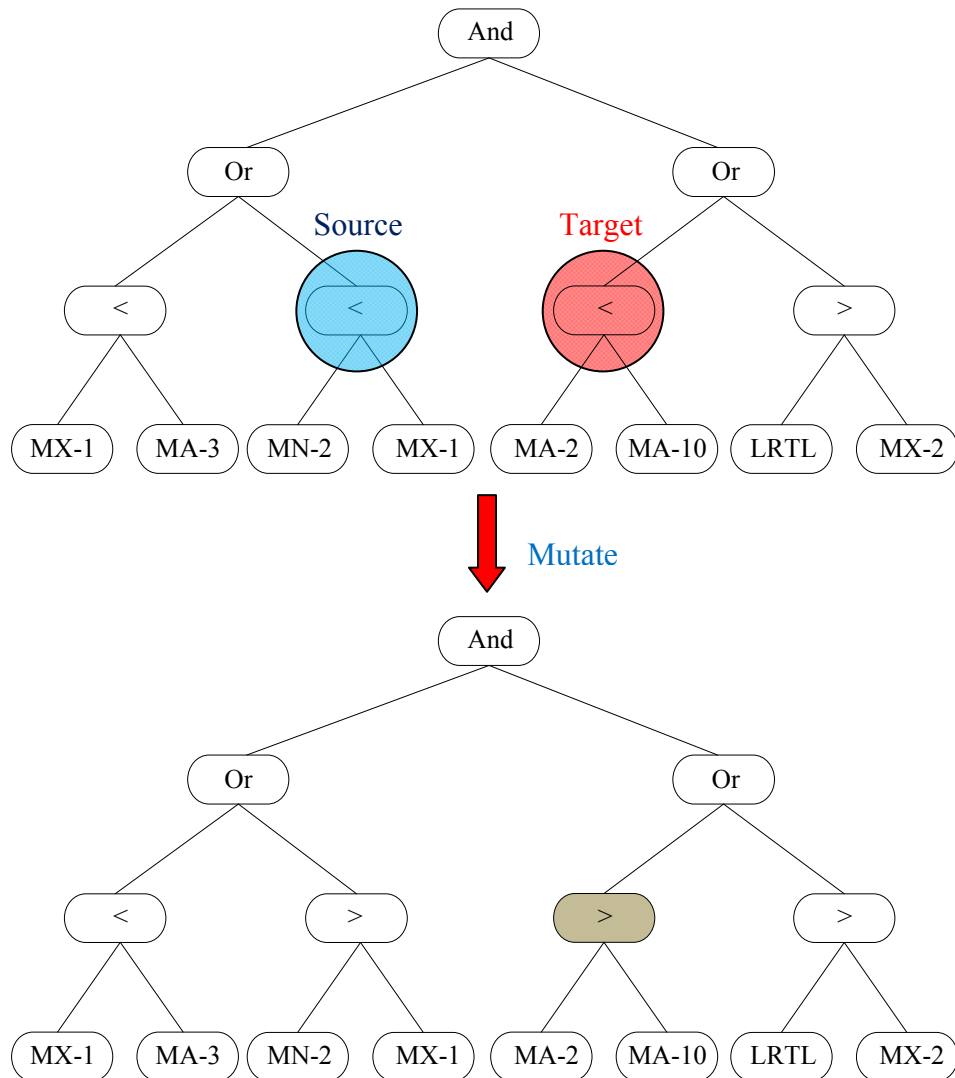


Figure 3-6: GP Point Mutation

3.2.4.4 *Permutation* – randomly select an internal node and reorder its argument subtrees (see Figure 3-7).

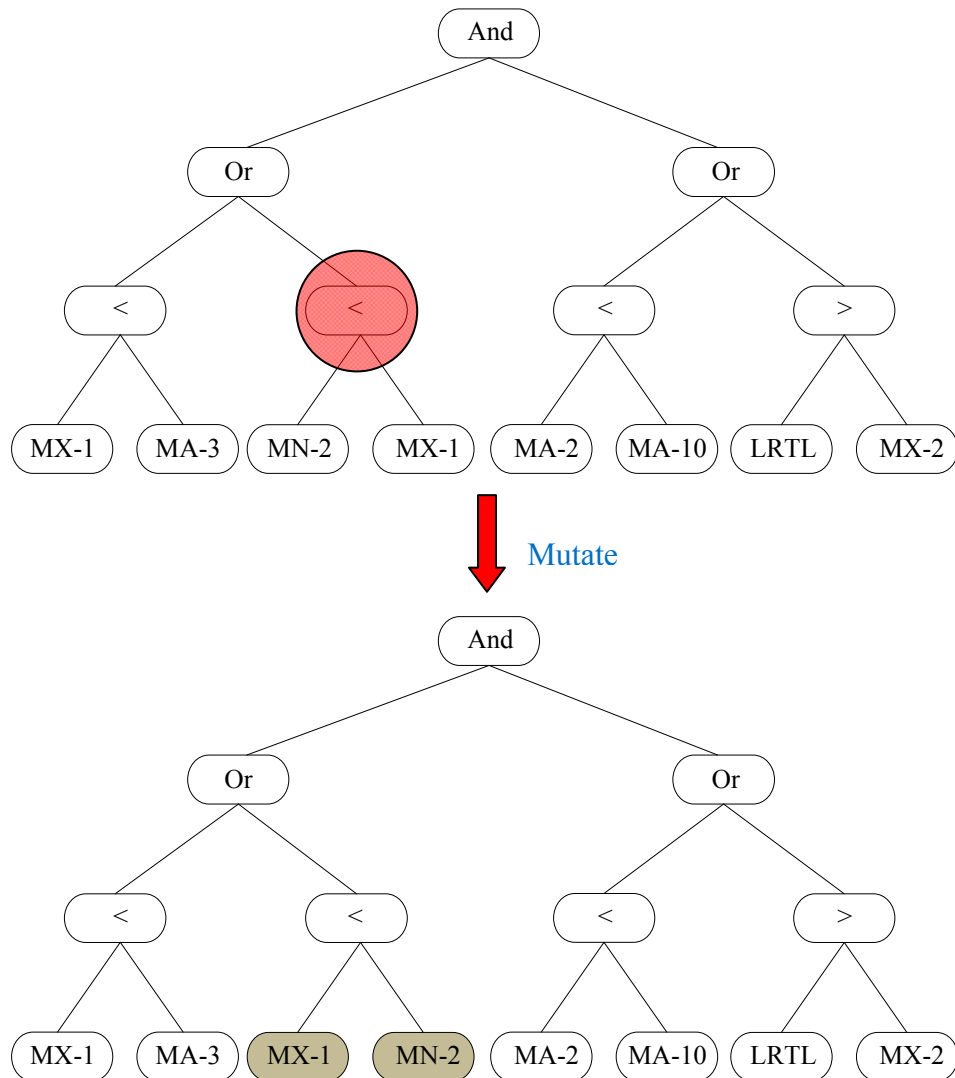


Figure 3-7: GP Permutation

3.2.4.5 *Hoist* – randomly select a subtree and generate a new individual form that subtree (see Figure 3-8).

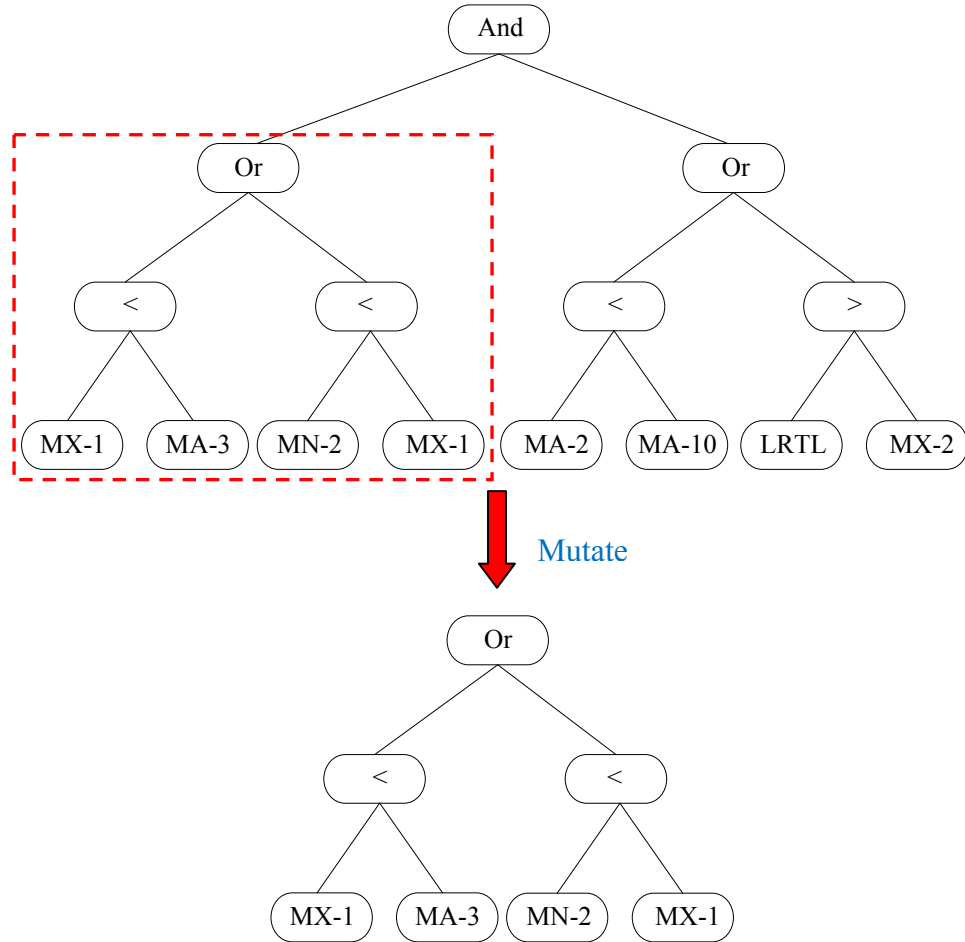


Figure 3-8: GP Hoist

In all experiments, the population was initialized by growing trees to a maximum depth of 7, but no further constraint was placed on tree size during evolution, other than the pressure offered by the objective function.

3.3 Experiments

3.3.1 GP Parameters

In all the experiments we describe, the GP system was as described in the last section, and used a population size of 500. In each generation, the current best was copied into the next generation, and the remainder was the product of 499 offspring-production events in which each such event was: apply crossover (probability 0.9) and then mutation (probability 0.1). For crossover, select two parents by binary tournament selection, perform crossover; for mutation, select one parent, and apply mutation (choosing uniformly from the available mutation operators). Each run continued for 50 generations.

Parameter	Description
Data Set	S&P 500 index
Algorithm	Steady State GP
Population Size	500
Iteration	50 generations
Initialization	Ramp half and half with depth range from 2 – 7
Crossover Rate	0.9
Mutation Rate	0.1
Selection Operator	Binary tournament selection
Mutation Operators	Subtree Mutation, Point Mutation, Permutation, Hoist
Crossover Operator	Subtree-swap crossover

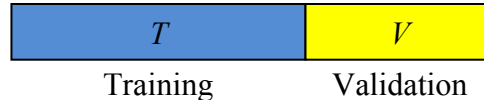
Table 3-2: GP Parameters of single-objective approach.

3.3.2 Data splits

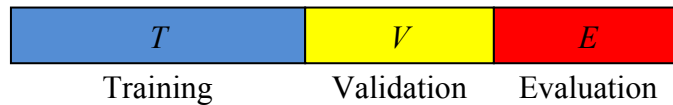
A data split indicates a particular period of time, itself divided into three contiguous periods of training, validation and test. In much of the earlier work in this thread of research, only one or a small number of data splits were used, which does not lead to results that are particularly robust in terms of the conclusions that can be made. In [60, 61] we used 4 different data splits for each of daily, weekly and monthly trading. In this chapter, we use eight different data splits for each of daily, weekly and monthly data (i.e. 24 different data splits altogether). In this way, for each trading frequency, we hope to underpin a good level of robustness in any claims about trends that emerge from the results.

In common with [2] and [6], the period 1960—1991 was generally used for training in the monthly-trading case. However, we explored two different approaches for choosing and evaluating a rule from the training run.

- *Regime 1*: the fittest rule found during training of T years (as measured on the training set) was applied to test data in an immediately succeeding period of V years.



- *Regime 2*: each rule found during the T year training period was validated against the ensuing V year period, and the rule that was best during this validation period was chosen, and tested over a further E year period beyond.



These two rule-selection regimes were each explored for 8 data period splits of monthly, weekly and daily data.

3.3.2.1 Monthly data splits

Table 3-3 specifies the eight different monthly data splits. Each is given a name (beginning with M or N_M), and the table indicates its range, length, and the buy-and-hold return for each of its 3 sub-periods (training/validation/evaluation). For example, when rule-selection regime 1 was used for data Monthly Split1 (M01), the rule chosen is the best on the 31-year training period, and this rule is evaluated on the subsequent 12 year period. When regime 2 is used for this split, the rule found while training on the 31-year period, which happened to be best on the subsequent 12 year period, was then evaluated on the further subsequent 5 year period.

Name		Period		Training Length (T)	Validation Length (V)	Evaluation Length (E)	B&H Return
		From	To				
Monthly Split1 M01	T	1960	1990	31 Years (372 Months)	12 Years (144 Months)	5 Years (60 Months)	5,457
	V	1991	2002				2,638
	E	2003	2007				1,652
Monthly Split2 M02	T	1960	1990	31 Years (372 Months)	8 Years (96 Months)	8 Years (96 Months)	5,457
	V	1991	1998				3,686
	E	1999	2006				1,142
Monthly Split3 M03	T	1960	1990	31 Years (372 Months)	9 Years (108 Months)	9 Years (108 Months)	5,457
	V	1991	1999				4,405
	E	2000	2008				609
Monthly Split4 M04	T	1960	1984	25 Years (300 Months)	12 Years (144 Months)	12 Years (144 Months)	2,764
	V	1985	1996				4,386
	E	1997	2008				1,207
Monthly Split5 N_M01	T	1963	1993	31 Years (372 Months)	12 Years (144 Months)	5 Years (60 Months)	7,319
	V	1994	2005				2,649
	E	2006	2010				998
Monthly Split6 N_M02	T	1964	1994	31 Years (372 Months)	8 Years (96 Months)	8 Years (96 Months)	6,061
	V	1995	2002				1,897
	E	2003	2010				1,415
Monthly Split7 N_M03	T	1962	1992	31 Years (372 Months)	9 Years (108 Months)	9 Years (108 Months)	6,029
	V	1993	2001				2,609
	E	2002	2010				1,085
Monthly Split8 N_M04	T	1962	1986	25 Years (300 Months)	12 Years (144 Months)	12 Years (144 Months)	3,351
	V	1987	1998				5,025
	E	1999	2010				1,013

Table 3-3: Details for each monthly data split – T: Training Period, V:

Validation Period, E: Evaluation Period.

In [2, 6, 8] the same training period is used, with testing only on the subsequent 12-year period, which corresponds to M01 in Table 3-3. Although this was not explicit in

[2, 6, 8], we think that the rules were chosen for evaluation in [2, 6, 8] via what we call regime 1. In Figure 3-9 and Figure 3-10, we show the S&P Index during the years 1960-2010, and graphically indicate the data splits used in our experiments: M01 - M04 and N_M01 - N_M04 respectively. In attempt to support our reasoning about the results (in case patterns can be observed that relate to market conditions during the data split), we next provide descriptions of overall trends for each monthly split. Notice how the different splits cover a range of different conditions, often involving salient challenges for the rule development process.

- *Regime 1:* In Splits 2-M02 and 3-M03, when evaluation regime 1 is used, the test period is one in which the market grew strongly, at a faster overall rate than during the training period. This is also true for Split4-M04 and Split8-N_M04, although in these cases the growth is at a lower rate. Outperforming B&H is always a stringent challenge, but moreso in such periods of growth. In split 1 (M01), Split6 (N_M02) and Split7 (N_M03), the challenge is more varied, with the test period covering 10 years of growth followed by 2 years of decline, 6 years of growth followed by 2 years of decline and 8 years of growth followed by a year of decline respectively. In the last and the most varied split, N_M01, the test period starts with 7 years of growth followed by 3 years of reduction, and ends with 2 years of growth.
- *Regime 2:* when rule-selection regime 2 is used for M01, we expect validation over a varied period to help select a robust rule that performs well in M01's growth-only evaluation period; we also expect this helps with Splits N_M01-N_M03 to get promising trading results in their more volatile evaluation periods. With Splits 2-M02 and 3-M03, regime 2 is challenged to produce a rule that performs well over a period that is roughly half decline and half growth, despite training and validation being done over periods largely in

growth. Even more challenging for regime2, Splits M04 and N_M04 have more volatile evaluation periods, whereas their training and validation periods show largely growth.

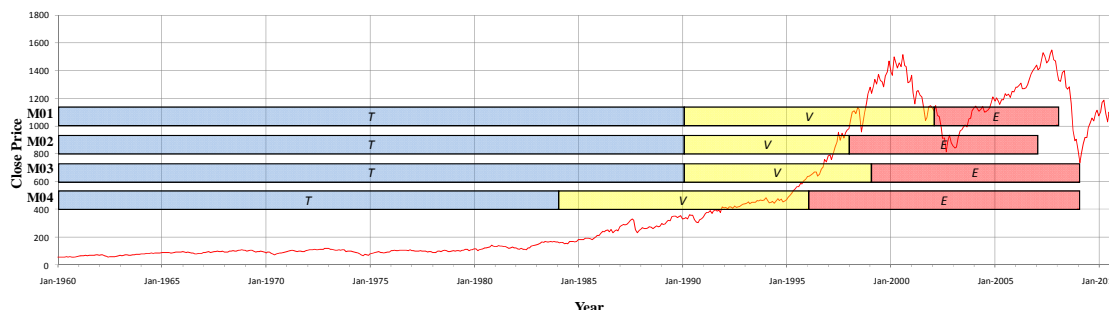


Figure 3-9: The S&P500 index over the period 1960-2010, illustrating four of the data splits for the case of monthly trading: M01-M04.

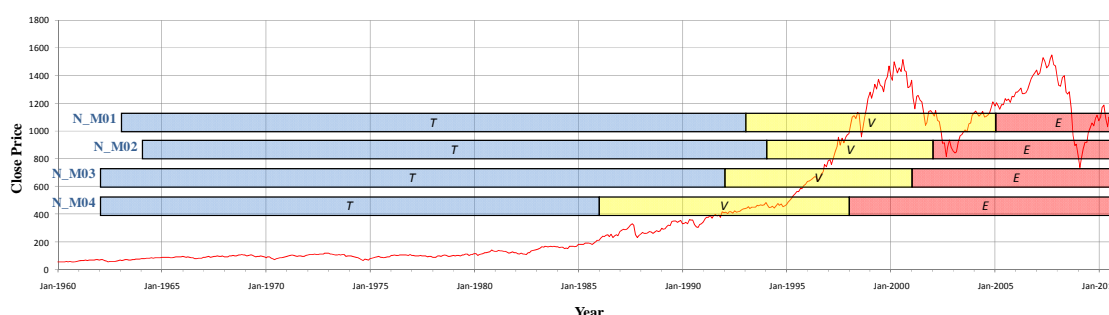


Figure 3-10: The S&P500 index over the period 1960-2010, illustrating four of the data splits for the case of monthly trading: N_M01-N_M04.

Data periods for weekly and for daily trading were chosen to be reasonably consistent with the monthly splits, in that the numbers of days or weeks involved corresponded with the number of months involved in the monthly splits. The details are as follows.

3.3.2.2 Weekly data splits

Table 3-4 specifies the weekly data splits that we have used. For each split the table shows the name (beginning with W or N_W), period range, period length and the buy-and-hold return for each of the three sub-periods (training/validation/evaluation).

Name		Period		Training Length (T)	Validation Length (V)	Evaluation Length (E)	B&H Return
		From	To				
Weekly Split1 W01	T	01-Jan-1960	31-Dec-1966	365 Weeks	157 Weeks	156 Weeks	1,328
	V	01-Jan-1967	31-Dec-1969				1,146
	E	01-Jan-1970	31-Dec-1972				1,257
Weekly Split2 W02	T	01-Jan-1972	31-Dec-1978	365 Weeks	157 Weeks	157 Weeks	932
	V	01-Jan-1979	31-Dec-1981				1,262
	E	01-Jan-1982	31-Dec-1984				1,322
Weekly Split3 W03	T	01-Jan-1984	31-Dec-1990	366 Weeks	156 Weeks	157 Weeks	1,927
	V	01-Jan-1991	31-Dec-1993				1,439
	E	01-Jan-1994	31-Dec-1996				1,588
Weekly Split4 W04	T	01-Jan-1996	31-Dec-2002	365 Weeks	156 Weeks	158 Weeks	1,460
	V	01-Jan-2003	31-Dec-2005				1,360
	E	01-Jan-2006	31-Dec-2008				716
Weekly Split5 N_W01	T	01-Jan-1962	31-Dec-1968	366 Weeks	156 Weeks	157 Weeks	1,439
	V	01-Jan-1969	31-Dec-1971				972
	E	01-Jan-1972	31-Dec-1974				686
Weekly Split6 N_W02	T	01-Jan-1974	31-Dec-1980	365 Weeks	156 Weeks	157 Weeks	1,365
	V	01-Jan-1981	31-Dec-1983				1,198
	E	01-Jan-1984	31-Dec-1986				1,479
Weekly Split7 N_W03	T	01-Jan-1986	31-Dec-1992	365 Weeks	156 Weeks	157 Weeks	2,046
	V	01-Jan-1993	31-Dec-1995				1,400
	E	01-Jan-1996	31-Dec-1998				1,976
Weekly Split8 N_W04	T	01-Jan-1998	31-Dec-2004	364 Weeks	157 Weeks	158 Weeks	1,231
	V	01-Jan-2005	31-Dec-2007				1,200
	E	01-Jan-2008	31-Dec-2010				848

Table 3-4: Details for each weekly data split – T: Training Period, V: Validation Period, E: Evaluation Period.

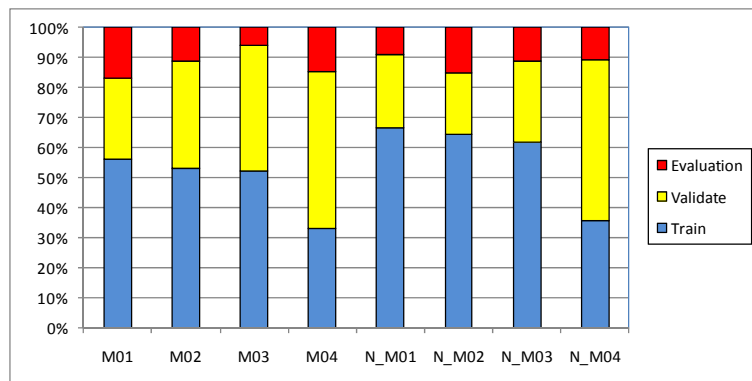
3.3.2.3 Daily data splits

Table 3-5 specifies the information about the eight daily data splits that we use, giving, for each split, the name (beginning with D or N_D), period range, period length and the buy-and-hold return for each of its sub-periods (training/validation/evaluation).

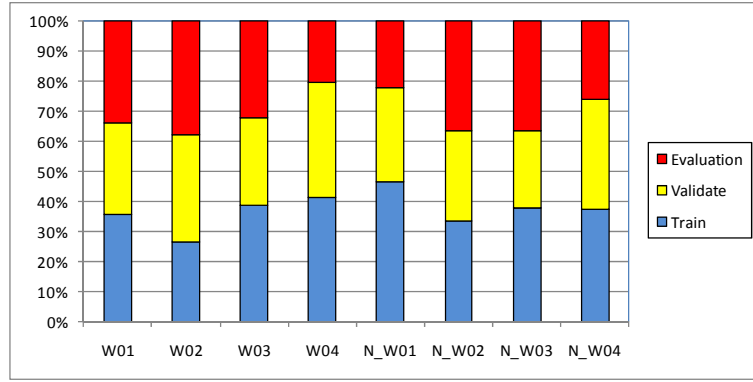
Name	Period		Training Length (T)	Validation Length (V)	Evaluation Length (E)	B&H Return
		From To				
Daily Split1 D01	T	01-Jan-1960 31-Jun-1961	377 Days	125 Days	126 Days	1,068
	V	01-Jul-1961 31-Dec-1961				1,086
	E	01-Jan-1962 31-Jun-1962				758
Daily Split2 D02	T	01-Jan-1975 30-Jun-1976	379 Days	126 Days	126 Days	1,504
	V	01-Jul-1976 31-Dec-1976				1,027
	E	01-Jan-1977 30-Jun-1977				926
Daily Split3 D03	T	01-Jan-1990 31-Jun-1991	378 Days	127 Days	126 Days	1,040
	V	01-Jul-1991 31-Dec-1991				1,093
	E	01-Jan-1992 31-Jun-1992				969
Daily Split4 D04	T	01-Jan-2006 30-Jun-2007	375 Days	127 Days	125 Days	1,192
	V	01-Jul-2007 31-Dec-2007				966
	E	01-Jan-2008 30-Jun-2008				863
Daily Split5 N_D01	T	01-Jan-1962 30-Jun-1963	377 Days	125 Days	126 Days	960
	V	01-Jul-1963 31-Dec-1963				1,079
	E	01-Jan-1964 30-Jun-1964				1,078
Daily Split6 N_D02	T	01-Jan-1977 30-Jun-1978	378 Days	126 Days	126 Days	880
	V	01-Jul-1978 31-Dec-1978				996
	E	01-Jan-1979 30-Jun-1979				1,060
Daily Split7 N_D03	T	01-Jan-1992 30-Jun-1993	379 Days	127 Days	125 Days	1,070
	V	01-Jul-1993 31-Dec-1993				1,028
	E	01-Jan-1994 30-Jun-1994				943
Daily Split8 N_D04	T	01-Jan-2008 30-Jun-2009	377 Days	127 Days	124 Days	620
	V	01-Jul-2009 31-Dec-2009				1,198
	E	01-Jan-2010 30-Jun-2010				914

Table 3-5: Details for each daily data split – T: Training Period, V: Validation Period, E: Evaluation Period.

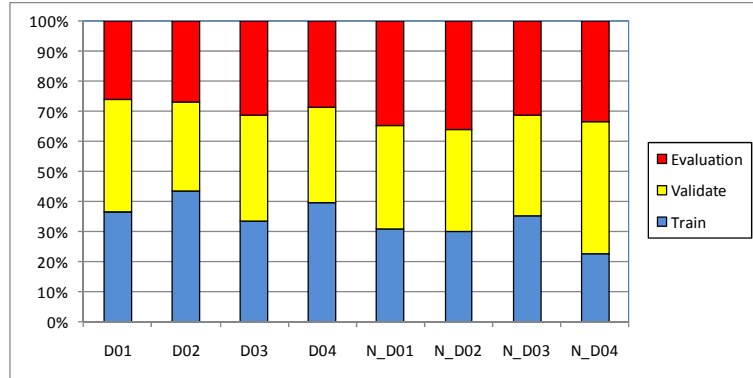
Finally, Figure 3-11 provides a graphic representation of the buy-and-hold returns for each of the data splits we have described. The market movements were net positive in each part of each data split, indicating that outperforming buy-and-hold was in all cases quite challenging.



a: Monthly Data



b: Weekly Data



c: Daily Data

Figure 3-11: Characterizing the buy-and-hold performance for each data split.

Monthly splits are on the top, weekly splits in the middle and monthly splits on the bottom. Each bar shows relative proportions of the buy-and-hold performance in the training (lower), validation (middle) and evaluation or test (upper) periods of the data split.

3.4 Results

For each trading period (monthly, weekly, daily), we performed 20 runs each for each combination of data split and performance consistency (PC) lookup-period, and we report results for each of rule-selection regime 1 and rule-selection regime 2. We summarize each set of runs in terms of performance in relation to buy-and-hold. Altogether there are 112 scenarios (8 data splits, 7 objectives and 2 rule selection and evaluation regimes). All results for monthly trading are summarized in Table 3-6 and

Table 3-7; weekly trading results are summarized in Table 3-9 and Table 3-10, and daily trading results are summarized in Table 3-11 and Table 3-12.

In the tables, each row indicates the mean over 20 runs of two pairs of related figures, and the number of times out of 20 that the result outperformed buy and hold. The first value of the first pair, ‘Excess over B&H (train)’, is the mean excess over buy and hold during the training period. That is, to give an example, if investing \$1,000 in a buy and hold strategy would yield \$5,000 over the training period, and the selected rule yielded \$20,000, this value would be \$15,000. In the second value of the first pair, this is re-expressed as a ratio – the return from the rule divided by the return from buy and hold. The second pair is more important, again expressing the excess and ratio in comparison to buy and hold, but this time on the appropriate test set. It should be noted that when we compare between the performances of rule-selection methods 1 and 2, we have to compare them with excess return ratio rather than raw excess return as the lengths of the test periods for regime 1 and regime 2 are different.

3.4.1 Monthly Trading

To help digest the raw monthly results in Table 3-6 and Table 3-7 and to make it much easier to compare between the results evaluated with regime 1 (especially M01 used in [2], [6] and [8]) and the results evaluated with regime 2, we provide Table 3-8 with summary notes, focusing on the test set results for each regime, as a function of the data splits. As we can see from the result of Split1-M01 in Table 3-6, and its summary in Table 3-8, it seems that monthly split M01 was clearly well-disposed to good performance with regime 1. Regime 1 simply means using all available data fully for the training process, with the intention of putting the best rule found during training into operation at the first opportunity after training – in practice, our experiments with M01, regime 1, evaluates how well this strategy would work if it were currently the end of

1991, and we generated a strategy (training on 1960-1991 data) to use from 1992 onwards. It is quite attractive to use all available data in this way, and this is clearly what was done in [2], [6] and [8]; however, our experiments suggest that the positive performance in these works was probably lucky and perhaps gave an over-optimistic view of the general promise of the method, since this strategy is still likely to overfit, and is sensitive to the data split. As the results in Table 3-8 show, the performance of regime 1 clearly depends on the details of the data split; regime 1 has strong performance in only 3 out of the 8 data splits.

#	Objectives	Regime	M01				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 55,789.24	11.223	\$ 2,364.50	1.896	100%
		2	\$ 49,257.59	10.026	\$ 111.83	1.068	100%
2	PC_LK6	1	\$ 5,316.46	1.974	\$ 987.79	1.374	90%
		2	\$ 4,525.02	1.829	\$ 70.02	1.042	70%
3	PC_LK12	1	\$ 7,811.04	2.431	\$ 848.57	1.322	95%
		2	\$ 8,306.98	2.522	\$ 13.65	1.008	85%
4	PC_LK18	1	\$ 12,841.11	3.353	\$ 790.38	1.300	95%
		2	\$ 10,119.59	2.854	\$ 61.15	1.037	70%
5	PC_LK24	1	\$ 45,773.13	9.388	\$ 1,784.27	1.676	100%
		2	\$ 44,166.22	9.093	\$ 88.36	1.053	95%
6	SHARO	1	\$ 35,524.43	7.510	\$ 2,049.23	1.777	100%
		2	\$ 31,947.87	6.854	\$ -70.47	0.957	55%
7	MSTLRO	1	\$ 23,415.84	5.291	\$ 1,280.68	1.485	90%
		2	\$ 20,535.76	4.763	\$ -206.38	0.875	20%

#	Objectives	Regime	M02				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 55,789.24	11.223	\$ -304.34	0.917	30%
		2	\$ 49,298.30	10.034	\$ 1,072.39	1.939	100%
2	PC_LK6	1	\$ 5,316.46	1.974	\$ 349.28	1.095	50%
		2	\$ 4,054.61	1.743	\$ 163.93	1.144	80%
3	PC_LK12	1	\$ 7,811.04	2.431	\$ -230.42	0.937	10%
		2	\$ 6,206.39	2.137	\$ 383.76	1.336	85%
4	PC_LK18	1	\$ 12,841.11	3.353	\$ 173.80	1.047	90%
		2	\$ 9,245.12	2.694	\$ 179.17	1.157	90%
5	PC_LK24	1	\$ 45,773.13	9.388	\$ -620.17	0.832	15%
		2	\$ 19,550.31	4.583	\$ 566.31	1.496	100%
6	SHARO	1	\$ 35,524.43	7.510	\$ -324.80	0.912	25%
		2	\$ 28,497.55	6.222	\$ 680.04	1.595	100%
7	MSTLRO	1	\$ 23,415.84	5.291	\$ -817.28	0.778	10%
		2	\$ 14,023.67	3.570	\$ 559.13	1.490	100%

#	Objectives	Regime	M03				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 55,789.24	11.223	\$ -419.07	0.905	30%
		2	\$ 49,298.30	10.034	\$ 1,421.46	3.334	100%
2	PC_LK6	1	\$ 5,316.46	1.974	\$ 494.39	1.112	55%
		2	\$ 4,054.61	1.743	\$ 127.69	1.210	90%
3	PC_LK12	1	\$ 7,811.04	2.431	\$ -282.84	0.936	10%
		2	\$ 6,142.53	2.126	\$ 562.55	1.924	85%
4	PC_LK18	1	\$ 12,841.11	3.353	\$ 93.53	1.021	85%
		2	\$ 9,245.12	2.694	\$ 251.07	1.412	95%
5	PC_LK24	1	\$ 45,773.13	9.388	\$ -753.66	0.829	15%
		2	\$ 17,477.08	4.203	\$ 706.77	2.161	100%
6	SHARO	1	\$ 35,524.43	7.510	\$ -454.47	0.897	25%
		2	\$ 28,497.55	6.222	\$ 1,054.47	2.731	100%
7	MSTLRO	1	\$ 23,415.84	5.291	\$ -1,029.81	0.766	10%
		2	\$ 14,023.67	3.570	\$ 968.93	2.591	100%

#	Objectives	Regime	M04				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 18,589.74	7.726	\$ 874.50	1.199	65%
		2	\$ 14,135.18	6.114	\$ 2,729.53	3.261	100%
2	PC_LK6	1	\$ 827.68	1.299	\$ 579.34	1.132	55%
		2	\$ 536.92	1.194	\$ 380.79	1.315	65%
3	PC_LK12	1	\$ 3,118.52	2.128	\$ -130.68	0.970	50%
		2	\$ 2,772.77	2.003	\$ 1,040.56	1.862	85%
4	PC_LK18	1	\$ 3,068.08	2.110	\$ 3,265.41	1.745	95%
		2	\$ 3,466.78	2.254	\$ 333.90	1.277	90%
5	PC_LK24	1	\$ 14,827.40	6.364	\$ 981.11	1.224	80%
		2	\$ 12,600.67	5.559	\$ 2,427.67	3.011	100%
6	SHARO	1	\$ 16,568.08	6.994	\$ 303.61	1.069	65%
		2	\$ 13,396.07	5.847	\$ 2,400.06	2.988	100%
7	MSTLRO	1	\$ 8,330.95	4.014	\$ -1,089.48	0.752	10%
		2	\$ 6,041.12	3.186	\$ 1,403.61	2.163	100%

Table 3-6: Summary of results for M01-M04 data splits of 7 different objectives: MR, 4 PC periods and 2 risk-adjusted approaches.

#	Objectives	Regime	N_M01				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 58,751.77	9.027	\$ 1,472.11	1.556	100%
		2	\$ 44,175.67	7.036	\$ 864.51	1.866	100%
2	PC_LK6	1	\$ 7,368.57	2.007	\$ 578.39	1.218	70%
		2	\$ 5,718.63	1.781	\$ 105.15	1.105	80%
3	PC_LK12	1	\$ 15,060.97	3.058	\$ 1,205.60	1.455	95%
		2	\$ 14,699.76	3.008	\$ 777.17	1.779	100%
4	PC_LK18	1	\$ 14,485.01	2.979	\$ 477.43	1.180	95%
		2	\$ 12,792.82	2.748	\$ 255.86	1.256	65%
5	PC_LK24	1	\$ 11,499.66	2.571	\$ 756.34	1.286	90%
		2	\$ 9,632.27	2.316	\$ 109.98	1.110	35%
6	SHARO	1	\$ 47,103.98	7.436	\$ 1,754.05	1.662	100%
		2	\$ 32,213.60	5.401	\$ 568.16	1.569	100%
7	MSTLRO	1	\$ 2,100.90	1.287	\$ -997.43	0.623	0%
		2	\$ 1,023.24	1.140	\$ 131.14	1.131	100%

#	Objectives	Regime	N_M02				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 47,658.68	8.863	\$ 1,232.84	1.650	100%
		2	\$ 37,774.68	7.232	\$ 1,077.90	1.762	100%
2	PC_LK6	1	\$ 9,562.29	2.578	\$ 681.22	1.359	100%
		2	\$ 7,548.40	2.245	\$ 131.91	1.093	100%
3	PC_LK12	1	\$ 11,683.85	2.928	\$ 729.29	1.384	95%
		2	\$ 12,055.68	2.989	\$ 920.07	1.650	95%
4	PC_LK18	1	\$ 31,440.05	6.187	\$ 1,110.80	1.586	100%
		2	\$ 23,146.65	4.819	\$ 853.57	1.603	80%
5	PC_LK24	1	\$ 39,814.49	7.569	\$ 1,063.35	1.561	100%
		2	\$ 29,469.47	5.862	\$ 1,075.90	1.760	100%
6	SHARO	1	\$ 37,906.88	7.254	\$ 1,356.55	1.715	100%
		2	\$ 28,471.96	5.698	\$ 650.02	1.459	80%
7	MSTLRO	1	\$ 2,904.55	1.479	\$ -412.75	0.782	0%
		2	\$ 2,163.49	1.357	\$ -274.44	0.806	0%

#	Objectives	Regime	N_M03				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 59,826.50	10.923	\$ 435.40	1.167	95%
		2	\$ 35,642.52	6.912	\$ 1,263.80	2.165	100%
2	PC_LK6	1	\$ 5,246.14	1.870	\$ 352.16	1.135	60%
		2	\$ 4,356.82	1.723	\$ 46.12	1.043	65%
3	PC_LK12	1	\$ 10,939.63	2.815	\$ 334.31	1.128	85%
		2	\$ 8,582.44	2.424	\$ 858.95	1.792	95%
4	PC_LK18	1	\$ 36,667.53	7.082	\$ 38.52	1.015	55%
		2	\$ 23,995.60	4.980	\$ 678.24	1.625	90%
5	PC_LK24	1	\$ 48,619.34	9.064	\$ 551.65	1.211	90%
		2	\$ 26,465.51	5.390	\$ 886.08	1.817	95%
6	SHARO	1	\$ 41,623.21	7.904	\$ 685.19	1.263	100%
		2	\$ 32,811.27	6.442	\$ 1,029.45	1.949	100%
7	MSTLRO	1	\$ 10,180.66	2.689	\$ -608.30	0.767	10%
		2	\$ 9,316.04	2.545	\$ 275.56	1.254	100%

#	Objectives	Regime	N_M04				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 19,216.96	6.735	\$ 156.97	1.031	60%
		2	\$ 14,702.54	5.388	\$ 2,158.53	3.131	100%
2	PC_LK6	1	\$ 1,816.64	1.542	\$ 1,187.09	1.236	65%
		2	\$ 1,545.50	1.461	\$ 236.88	1.234	95%
3	PC_LK12	1	\$ 4,714.48	2.407	\$ -179.54	0.964	10%
		2	\$ 3,825.26	2.142	\$ 1,087.32	2.073	90%
4	PC_LK18	1	\$ 14,352.79	5.283	\$ -627.09	0.875	20%
		2	\$ 11,311.27	4.375	\$ 1,628.03	2.607	90%
5	PC_LK24	1	\$ 16,007.46	5.777	\$ 85.99	1.017	55%
		2	\$ 11,383.46	4.397	\$ 1,768.24	2.746	100%
6	SHARO	1	\$ 16,309.51	5.867	\$ 389.53	1.078	45%
		2	\$ 12,734.30	4.800	\$ 1,615.76	2.595	100%
7	MSTLRO	1	\$ 4,774.56	2.425	\$ -2,730.80	0.457	0%
		2	\$ 2,730.67	1.815	\$ 483.46	1.477	100%

Table 3-7: Summary of results for N_M01-N_M04 data splits of 7 different objectives: MR, 4 PC periods and 2 risk-adjusted approaches.

It is also clear that regime 2 for monthly trading, in which the rule we select for trading is one that has been cross-validated on an intervening period, is a better choice, providing more reliable and consistent results and leading to ideal performance. This in itself is not surprising, but it suggests that this additional protection against over-fitting (over and above the measures used in the design of the fitness function) is worth using. The results of evaluation regime 2 from Table 3-8 gave us evidence that this approach always yielded rules that outperformed B&H and risk-free investment, whether the market was consistently rising (such as during the test period of M01), moderately rising (as in the test periods of M04 and N_M01-N_M04) or having mixed behaviour (as in the remaining splits), although the excess over B&H for the consistently rising market (M01) was modest and underperformed compared with regime 1; it was only in the M01 split that regime 2 was outperformed by regime 1.

Experiment Setup	Results	Results
	Evaluation regime 1	Evaluation regime 2
Split1-M01 T:31 yrs, V:12 yrs, E:5 yrs	Always outperformed B&H, for all configurations by around 90-100%.	Completely outperformed B&H for MR, consistently outperformed B&H around 70-95% for all PC periods,

		lose to B&H for all risk-adjusted approach.
Split2-M02 T:31 yrs, V:8 yrs, E:8 yrs	Poor performance at around 10-30% to beat B&H, except PC_LK6 and PC_LK18 with performance at 50% and 90% respectively.	Consistently outperformed B&H, for all configurations by roughly 80-100%.
Split3-M03 T:31 yrs, V:9 yrs, E:9 yrs	Lose to buy and hold, with performance rate at roughly 10-30%, with the exception of PC_LK6 and PC_LK18 with performance rate at 55% and 85% respectively.	Well performance over B&H, for all configurations around 85-100%.
Split4-M04 T:25 yrs, V:12 yrs, E:12 yrs	Modest excess over B&H for MR, PC_LK6, PC_LK12 and SHARO, consistently better for longer PC Periods of PC_LK18 and PC_LK24. Very poor performance for Modified Sterling Ratio (MSTRO).	Perfectly outperforming B&H for MR, the longest PC period-PC_LK24 and all risk-adjusted configurations, marginally dropping on performance for shorter PC periods.
Split5-N_M01 T:31 yrs, V:12 yrs, E:5 yrs	Always outperformed B&H, for all configurations by around 70-100%, except MSTLRO.	Strong performance against B&H for MR, PC_LK6, PC_LK12, SHARO and MSTLRO. However, weak performance for the remainder of two longer PC Periods.
Split6-N_M02 T:31 yrs, V:8 yrs, E:8 yrs	Very well performance with positive excess over B&H for all methods, except only one negative excess over B&H for	Considerably well and consistent performance over B&H for all configurations with the only single exception of

	MSTRO.	MSTRO.
Split7-N_M03 T:31 yrs, V:9 yrs, E:9 yrs	Noticeable excess over B&H for 4 methods: MR, PC_LK12, PC_LK24 and SHARO, modest excess over B&H for PC_LK6 and PC_LK18, and very low excess over B&H for MSTLRO.	Consistently outperformed B&H, for all configurations by around 90-100% with the sole exception of the shortest PC period, PC_LK6, at 65%
Split8-N_M04 T:25 yrs, V:12 yrs, E:12 yrs	Modestly outperformed B&H for half of methods: MR, PC_LK6, PC_LK24 and SHARO, but poor performance for the rest.	Always outperformed B&H, for all configurations by around 90-100%

Table 3-8: Summary notes comparing different setups.

If we now consider Figure 3-11 (upper) and attempt to understand relative performance in terms of the overall market movements in the data splits, we find that this is quite hard to do. Referring to Figure 3-9 and Figure 3-10, we see that the market conditions were fairly similar for the training and validation parts of each monthly split, and were ‘up and down’ for each of the 8 test periods.

3.4.2 Weekly Trading

Table 3-9 and Table 3-10 show the results, summarized in the same way as in monthly trading, but this time for the case of weekly trading. These tables clearly show less robust results for weekly trading. Having said that, it certainly seems that the method with regime 2 can find robust rules for weekly trading that outperform buy-and-hold in

all circumstances. However, again, it seems we cannot discern any pattern that underpins this from the basic summary of the data splits' buy-and-hold performance in Figure 3-11 (middle). Furthermore, regime 2 has proven its robustness again in the case of weekly trading with stronger performance on 6 periods over regime1: W01, W03, W04, N_W01, N_W03 and N_W04.

#	Objectives	Regime	W01				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 762.01	1.574	\$ 67.65	1.059	90%
		2	\$ 715.61	1.539	\$ 183.85	1.146	100%
2	PC_LK12	1	\$ 634.77	1.478	\$ 0.93	1.001	40%
		2	\$ 427.85	1.322	\$ 127.50	1.101	85%
3	PC_LK24	1	\$ 70.29	1.053	\$ -32.53	0.972	10%
		2	\$ 82.19	1.062	\$ 2.47	1.002	55%
4	SHARO	1	\$ 643.95	1.485	\$ 87.29	1.076	100%
		2	\$ 417.12	1.314	\$ 193.02	1.154	100%
5	MSTLRO	1	\$ 225.17	1.170	\$ 34.06	1.030	80%
		2	\$ 114.33	1.086	\$ -55.45	0.956	35%

#	Objectives	Regime	W02				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 2,205.73	3.367	\$ 577.77	1.458	100%
		2	\$ 1,982.79	3.127	\$ 243.50	1.184	100%
2	PC_LK12	1	\$ 1,257.22	2.349	\$ 232.72	1.184	95%
		2	\$ 1,114.68	2.196	\$ 100.88	1.076	85%
3	PC_LK24	1	\$ 1,314.56	2.410	\$ 330.58	1.262	100%
		2	\$ 751.09	1.806	\$ 103.20	1.078	70%
4	SHARO	1	\$ 1,419.85	2.523	\$ 464.10	1.368	100%
		2	\$ 1,415.63	2.519	\$ 91.48	1.069	100%
5	MSTLRO	1	\$ 938.97	2.007	\$ 276.81	1.219	100%
		2	\$ 1,098.74	2.179	\$ 17.18	1.013	60%

#	Objectives	Regime	W03				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 3,159.79	2.640	\$ 2.18	1.002	65%
		2	\$ 2,200.58	2.142	\$ 54.73	1.034	85%
2	PC_LK12	1	\$ 1,542.88	1.801	\$ -88.09	0.939	5%
		2	\$ 974.53	1.506	\$ -35.20	0.978	60%
3	PC_LK24	1	\$ 1,609.13	1.835	\$ -107.21	0.925	15%
		2	\$ 1,196.49	1.621	\$ -89.34	0.944	50%
4	SHARO	1	\$ 2,895.05	2.502	\$ -106.31	0.926	25%
		2	\$ 2,117.88	2.099	\$ 6.40	1.004	60%
5	MSTLRO	1	\$ 512.14	1.266	\$ -370.01	0.743	0%
		2	\$ 63.52	1.033	\$ -362.78	0.772	0%

#	Objectives	Regime	W04				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 2,406.69	2.648	\$ 34.17	1.025	60%
		2	\$ 1,916.45	2.313	\$ 538.49	1.752	100%
2	PC_LK12	1	\$ 1,414.68	1.969	\$ 69.28	1.051	80%
		2	\$ 903.83	1.619	\$ 286.48	1.400	85%
3	PC_LK24	1	\$ 2,019.71	2.383	\$ 12.28	1.009	50%
		2	\$ 1,279.71	1.877	\$ 358.24	1.500	95%
4	SHARO	1	\$ 1,452.31	1.995	\$ 87.49	1.064	90%
		2	\$ 686.09	1.470	\$ 552.89	1.772	100%
5	MSTLRO	1	\$ 693.23	1.475	\$ -166.13	0.878	5%
		2	\$ 847.45	1.580	\$ 524.94	1.733	100%

Table 3-9: Summary of results for W01-W04 data splits of 5 different objectives: MR, 2 PC periods and 2 risk-adjusted approaches.

#	Objectives	Regime	N_W01				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 883.09	1.614	\$ 219.58	1.226	95%
		2	\$ 820.33	1.570	\$ 249.35	1.363	95%
2	PC_LK12	1	\$ 37.37	1.026	\$ 38.28	1.039	70%
		2	\$ -20.10	0.986	\$ 95.02	1.139	85%
3	PC_LK24	1	\$ 69.53	1.048	\$ 20.09	1.021	45%
		2	\$ -58.03	0.960	\$ 29.72	1.043	75%
4	SHARO	1	\$ 850.66	1.591	\$ 287.46	1.296	95%
		2	\$ 662.86	1.461	\$ 394.52	1.575	100%
5	MSTLRO	1	\$ 393.82	1.274	\$ 339.71	1.349	100%
		2	\$ 321.22	1.223	\$ 334.52	1.488	100%

#	Objectives	Regime	N_W02				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 3,118.13	3.284	\$ 351.09	1.293	100%
		2	\$ 2,661.22	2.950	\$ 162.87	1.110	95%
2	PC_LK12	1	\$ 1,505.19	2.103	\$ 171.68	1.143	100%
		2	\$ 1,281.52	1.939	\$ 20.37	1.014	55%
3	PC_LK24	1	\$ 2,902.61	3.126	\$ 273.64	1.228	100%
		2	\$ 1,426.65	2.045	\$ -8.80	0.994	55%
4	SHARO	1	\$ 2,005.08	2.469	\$ 317.31	1.265	100%
		2	\$ 1,877.13	2.375	\$ 86.25	1.058	95%
5	MSTLRO	1	\$ 1,404.75	2.029	\$ 276.23	1.231	100%
		2	\$ 1,247.94	1.914	\$ 25.40	1.017	90%

#	Objectives	Regime	N_W03				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 2,925.03	2.430	\$ -57.76	0.959	25%
		2	\$ 1,961.64	1.959	\$ 29.71	1.015	75%
2	PC_LK12	1	\$ 1,117.37	1.546	\$ -99.45	0.929	10%
		2	\$ 1,073.52	1.525	\$ -34.87	0.982	50%
3	PC_LK24	1	\$ 1,869.76	1.914	\$ -129.89	0.907	10%
		2	\$ 1,293.92	1.632	\$ -32.75	0.983	55%
4	SHARO	1	\$ 2,778.50	2.358	\$ 37.59	1.027	70%
		2	\$ 2,365.10	2.156	\$ 59.83	1.030	60%
5	MSTLRO	1	\$ 200.95	1.098	\$ -247.06	0.824	0%
		2	\$ 93.06	1.045	\$ -434.63	0.780	0%

#	Objectives	Regime	N_W04				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 1,720.42	2.398	\$ 62.47	1.052	100%
		2	\$ 1,363.68	2.108	\$ 630.94	1.744	100%
2	PC_LK12	1	\$ 2,201.31	2.788	\$ 72.89	1.061	95%
		2	\$ 1,201.17	1.976	\$ 306.69	1.362	75%
3	PC_LK24	1	\$ 893.53	1.726	\$ 99.64	1.083	90%
		2	\$ 490.95	1.399	\$ 129.80	1.153	75%
4	SHARO	1	\$ 1,584.95	2.288	\$ 75.21	1.063	90%
		2	\$ 1,308.34	2.063	\$ 377.45	1.445	100%
5	MSTLRO	1	\$ 739.91	1.601	\$ 32.51	1.027	85%
		2	\$ 652.60	1.530	\$ 253.93	1.299	100%

Table 3-10: Summary of results for N_W01-N_W04 data splits of 5 different objectives: MR, 2 PC periods and 2 risk-adjusted approaches.

3.4.3 Daily Trading

Table 3-11 and Table 3-12 present the corresponding results for the case of daily trading, and the results indicate less reliable performance compared with weekly trading. For daily trading, Table 3-11 and Table 3-12 show that outperforming buy-and-hold is less likely, with strong performance in two of the eight data splits (D_04 and N_D04), modest performance in three of the data splits (D02, D03 and N_D03) and very poor performance for the rest (D01, N_D01 and N_D02). In comparison between regime 1 and regime 2, we found that regime 2 can bring about stronger robustness particularly in the two risk-adjusted methods (underperforming to regime 1 in only N_D02 split).

#	Objectives	Regime	D01				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 11.57	1.011	\$ -15.96	0.985	15%
		2	\$ 1.18	1.001	\$ -7.18	0.991	0%
2	PC_LK12	1	\$ -17.17	0.984	\$ -7.24	0.993	75%
		2	\$ -1.87	0.998	\$ -15.47	0.980	0%
3	PC_LK24	1	\$ -2.44	0.998	\$ -1.07	0.999	80%
		2	\$ -2.44	0.998	\$ -5.12	0.993	0%
4	SHARO	1	\$ 6.27	1.006	\$ -24.82	0.977	0%
		2	\$ -0.76	0.999	\$ 6.16	1.008	5%
5	MSTLRO	1	\$ -22.21	0.979	\$ -113.68	0.895	0%
		2	\$ -37.42	0.965	\$ 230.45	1.304	100%

#	Objectives	Regime	D02				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 179.02	1.119	\$ -8.83	0.991	25%
		2	\$ 132.55	1.088	\$ -1.10	0.999	35%
2	PC_LK12	1	\$ -27.56	0.982	\$ -13.05	0.987	30%
		2	\$ -21.55	0.986	\$ 13.51	1.015	75%
3	PC_LK24	1	\$ 69.67	1.046	\$ -18.42	0.982	10%
		2	\$ 48.22	1.032	\$ 0.26	1.000	10%
4	SHARO	1	\$ 163.63	1.109	\$ -12.64	0.988	20%
		2	\$ 127.21	1.085	\$ -1.47	0.998	35%
5	MSTLRO	1	\$ -132.00	0.912	\$ -9.20	0.991	55%
		2	\$ -211.13	0.860	\$ 72.15	1.078	100%

#	Objectives	Regime	D03				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 290.15	1.279	\$ -34.36	0.969	30%
		2	\$ 203.70	1.196	\$ 15.59	1.016	90%
2	PC_LK12	1	\$ 210.34	1.202	\$ -26.22	0.976	5%
		2	\$ 57.59	1.055	\$ -35.04	0.964	10%
3	PC_LK24	1	\$ 227.50	1.219	\$ -18.77	0.983	30%
		2	\$ 159.78	1.154	\$ 23.03	1.024	85%
4	SHARO	1	\$ 199.89	1.192	\$ -35.27	0.968	35%
		2	\$ 154.55	1.149	\$ 25.63	1.026	100%
5	MSTLRO	1	\$ 215.33	1.207	\$ -47.88	0.956	35%
		2	\$ 165.16	1.159	\$ 23.76	1.025	95%

#	Objectives	Regime	D04				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 13.65	1.011	\$ 5.91	1.006	90%
		2	\$ 8.51	1.007	\$ 11.74	1.014	30%
2	PC_LK12	1	\$ -16.37	0.986	\$ 26.22	1.027	95%
		2	\$ -16.85	0.986	\$ 13.80	1.016	40%
3	PC_LK24	1	\$ -64.27	0.946	\$ -2.99	0.997	75%
		2	\$ -40.21	0.966	\$ 14.02	1.016	35%
4	SHARO	1	\$ 16.53	1.014	\$ -0.84	0.999	70%
		2	\$ 14.96	1.013	\$ 6.56	1.008	35%
5	MSTLRO	1	\$ 49.91	1.042	\$ 15.88	1.016	85%
		2	\$ -91.09	0.924	\$ 135.10	1.157	95%

Table 3-11: Summary of results for D01-D04 data splits of 5 different objectives: MR, 2 PC periods and 2 risk-adjusted approaches.

#	Objectives	Regime	N_D01				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 177.31	1.185	\$ -41.12	0.962	0%
		2	\$ 145.56	1.152	\$ 14.45	1.013	85%
2	PC_LK12	1	\$ -8.28	0.991	\$ -132.82	0.877	0%
		2	\$ -24.05	0.975	\$ -45.02	0.958	0%
3	PC_LK24	1	\$ -15.61	0.984	\$ -81.94	0.924	0%
		2	\$ -4.04	0.996	\$ -11.78	0.989	10%
4	SHARO	1	\$ 201.62	1.210	\$ -48.24	0.955	0%
		2	\$ 124.85	1.130	\$ 9.07	1.008	70%
5	MSTLRO	1	\$ 142.83	1.149	\$ -83.20	0.923	0%
		2	\$ 140.98	1.147	\$ -38.11	0.965	45%

#	Objectives	Regime	N_D02				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 256.16	1.291	\$ 49.08	1.049	100%
		2	\$ 255.08	1.290	\$ -50.07	0.953	0%
2	PC_LK12	1	\$ 98.40	1.112	\$ 33.93	1.034	90%
		2	\$ 54.94	1.062	\$ -54.84	0.948	0%
3	PC_LK24	1	\$ 21.57	1.025	\$ 9.41	1.009	80%
		2	\$ 21.15	1.024	\$ -57.73	0.946	0%
4	SHARO	1	\$ 255.86	1.291	\$ 49.04	1.049	100%
		2	\$ 254.78	1.290	\$ -50.64	0.952	0%
5	MSTLRO	1	\$ 256.16	1.291	\$ 49.16	1.049	100%
		2	\$ 254.00	1.289	\$ -49.80	0.953	0%

#	Objectives	Regime	N_D03				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outperforming B&H
1	MR	1	\$ 21.46	1.020	\$ -9.07	0.991	65%
		2	\$ 9.37	1.009	\$ 13.80	1.015	100%
2	PC_LK12	1	\$ -9.18	0.991	\$ -21.73	0.979	10%
		2	\$ -82.41	0.923	\$ -18.46	0.980	10%
3	PC_LK24	1	\$ -20.80	0.981	\$ -87.44	0.915	0%
		2	\$ -40.32	0.962	\$ 25.28	1.027	100%
4	SHARO	1	\$ 11.24	1.011	\$ -3.20	0.997	85%
		2	\$ 6.01	1.006	\$ 11.53	1.012	100%
5	MSTLRO	1	\$ 4.40	1.004	\$ -42.27	0.959	0%
		2	\$ -2.48	0.998	\$ 34.37	1.036	100%

#	Objectives	Regime	N_D04				
			Excess over B&H (train)	Excess over B&H (train/ratio)	Excess over B&H (test)	Excess over B&H (test/ratio)	Outper- forming B&H
1	MR	1	\$ 1,024.99	2.653	\$ -196.99	0.836	0%
		2	\$ 527.97	1.852	\$ 91.62	1.100	100%
2	PC_LK12	1	\$ 631.91	2.019	\$ -104.52	0.913	30%
		2	\$ 332.24	1.536	\$ 34.66	1.038	75%
3	PC_LK24	1	\$ 896.61	2.446	\$ -166.59	0.861	0%
		2	\$ 389.35	1.628	\$ 25.74	1.028	60%
4	SHARO	1	\$ 1,052.40	2.697	\$ -212.40	0.823	0%
		2	\$ 684.98	2.105	\$ 92.43	1.101	100%
5	MSTLRO	1	\$ 793.97	2.281	\$ -219.38	0.817	0%
		2	\$ 561.59	1.906	\$ 90.92	1.099	100%

Table 3-12: Summary of results for N_D01-N_D04 data splits of 5 different objectives: MR, 2 PC periods and 2 risk-adjusted approaches.

3.4.4 Performance Comparison

Data Split	Best Method	Best of PC Periods	Best of Risk-adjusted Approach	PC approach vs. Risk-adjusted approach
Monthly	MR: 8/8	PC_LK24:7/8 PC_LK12:1/8	SHARO:8/8	BEST PC: 5/8 BEST RISK: 3/8
Weekly	MR: 4/8 SHARO: 4/8	PC_LK12:5/8 PC_LK24:3/8	SHARO:8/8	BEST RISK: 7/8 BEST PC: 1/8
Daily	MSTLRO: 5/8 SHARO: 2/8 MR: 1/8	PC_LK24:5/8 PC_LK12:3/8	MSTLRO: 5/8 SHARO: 3/8	BEST RISK: 8/8

Table 3-13: Summary note on performance for each method with regime 2 for monthly, weekly and daily data.

Now we consider the performance of each method in detail with help from the summary notes in Table 3-13, of which the last column is generated by selecting the best of the PC periods in the relevant split and then comparing that with the better of the two risk-adjusted methods in the same split. All such comparisons were done by using the means of the return over 20 runs. To illustrate, MR: 8/8 in the column ‘Best Method’ means that MR seems to be the best method in this case, outperforming all other methods in 8 out of 8 data splits. In other words, the mean return of MR is the higher than that of the other methods over each of the 8 data splits. To give another example, PC_LK24: 7/8 and PC_LK12: 1/8 in the column ‘Best of PC Periods’ means that from the total of 8 data splits, PC_LK24 has the highest mean in 7 of the data splits and PC_LK12 has the highest mean in only one data split. For the last example, SHARO: 8/8 in the column ‘Best of Risk-adjusted Approach’ means that SHARO has the highest mean of the two risk-adjusted methods in all 8 data splits.

Among the seven configurations for monthly trading, MR, whose objective is strongly driven by profit, was the one that outperformed all other configurations for all monthly periods in terms of mean result. Concerning period length for the Performance Consistency fitness function, the results indicated that the longest PC Period – PC_LK24 was the most robust and consistent configuration – outperforming other period lengths in 7 out of 8 monthly periods. In addition, comparing between the best PC method (BEST PC) and the better of the two risk-adjusted methods (BEST RISK) for each spilt, BEST PC seemed to perform more reliably than BEST RISK (SHARO in this case) with 5 out of 8.

Period	Best Method	Configurations to compare with the Best Method using one-tailed paired T-Test						
		config	PC_LK6	PC_LK12	PC_LK18	PC_LK24	SHARO	MSTLRO
M01	MR	mean	1722.021	1665.645	1713.154	1740.363	1581.526	1445.623
		1763.835	p-value	0.098447	5.59E-08	0.019793	0.056057	0.000219
								6.06E-07
M02	MR	config	PC_LK6	PC_LK12	PC_LK18	PC_LK24	SHARO	MSTLRO
		mean	1305.928	1525.763	1321.174	1708.306	1822.04	1701.127
		2214.391	p-value	4.71E-15	4.48E-10	1.29E-10	2.5E-06	9.13E-06
M03	MR	config	PC_LK6	PC_LK12	PC_LK18	PC_LK24	SHARO	MSTLRO
		mean	736.6939	1171.552	860.0693	1315.774	1663.466	1577.929
		2030.462	p-value	6.89E-17	3.36E-09	5.85E-11	7.16E-07	6.48E-05
M04	MR	config	PC_LK6	PC_LK12	PC_LK18	PC_LK24	SHARO	MSTLRO
		mean	1587.79	2247.561	1540.903	3634.667	3607.055	2610.609
		3936.528	p-value	3.1E-11	4.27E-08	1.53E-13	0.071374	0.090697
N_M01	MR	config	PC_LK6	PC_LK12	PC_LK18	PC_LK24	SHARO	MSTLRO
		mean	1103.15	1775.168	1253.858	1107.977	1566.164	1129.138
		1862.515	p-value	6.52E-10	9.06E-02	1.03E-06	3.04E-09	0.005124
N_M02	MR	config	PC_LK6	PC_LK12	PC_LK18	PC_LK24	SHARO	MSTLRO
		mean	1546.907	2335.074	2268.573	2490.896	2065.023	1140.558
		2492.902	p-value	2.6E-08	1.48E-01	0.070334	0.4945	0.004193
N_M03	MR	config	PC_LK6	PC_LK12	PC_LK18	PC_LK24	SHARO	MSTLRO
		mean	1131.123	1943.946	1763.235	1971.084	2114.446	1360.556
		2348.802	p-value	1.4E-07	3.74E-02	0.011556	0.06837	0.162259
N_M04	MR	config	PC_LK6	PC_LK12	PC_LK18	PC_LK24	SHARO	MSTLRO
		mean	1249.877	2100.316	2641.031	2781.237	2628.761	1496.459
		3171.527	p-value	7.53E-09	7.51E-05	0.048685	0.139176	0.058312

Table 3-14: Paired one-tailed T-Test on the hypothesis that the best method is MR for monthly trading, displaying mean and p-value for each configuration, spotting p-value > 0.10 with gray highlight.

To seek more concrete conclusions, Table 3-14 provides the results of one-tailed T-Tests (assuming unequal variance) comparing MR with each of the other configurations, for each data split. In all statistical comparisons in this thesis, we assume a cutoff p -value for significance of 0.1. For example, in data split M01, according to a paired one-tailed T-Test, the claim that MR (with the mean of \$1,763.835) is the best method seems justified, since the p -value of the T-test that compares MR with the method that gave second-best mean (PC_LK24) is < 0.10 .

Similarly, the claim that MR is the best method for monthly trading looks reasonable for data splits M02, M03, M04 and N_M01, however for the other three data splits, the p -values of the comparison with the method that gave second-best mean show that the comparison is inconclusive. Namely, MR shows no statistically significant difference from PC_LK24 in N_M02 and N_M04, and from SHARO in N_M03.

The performance of MR dropped significantly in the case of weekly trading, with more fickle data. As we can see in Table 3-13, it turned out that MR could achieve a higher mean than all other methods in 4 out of the 8 weekly periods (W02, W03, N_W02 and N_W04), whereas Sharpe Ratio (SHARO), the better of two risk-adjusted methods in the case of weekly data, similarly outperformed the other methods in the remaining 4 periods: W01, W04, N_W01 and N_W03. Moreover, it is clear that SHARO outperforms PC on weekly trading, with better results in 7 splits. This suggests that risk-related methods seem to perform better on more volatile data.

Again, we did one-tailed paired T-Tests on the weekly trading results, and report the findings in Table 3-15. In each case, the T-test is done to compare the method which returned the best mean value with the method that returned the second-best mean value. We find that MR was best method in periods W02, W03, N_W02 and N_W04 with p -values < 0.10 in each case, however in the other cases (where SHARO had the best

mean value), the comparison with the second-best method showed no statistical significance.

Period	Best Method	Configurations to compare with the Best Method using one-tailed paired T-Test				
W01	SHARO	config	MR	PC_LK12	PC_LK24	MSTLRO
		mean	1440.849	1384.502	1259.466	1201.551
		1450.018 p-value	0.331707	0.007071	3.93E-07	7.21E-09
W02	MR	config	PC_LK12	PC_LK24	SHARO	MSTLRO
		mean	1422.879	1425.196	1413.48	1339.178
		1565.496 p-value	0.000537	7.50E-04	2.67E-05	2.92E-07
W03	MR	config	PC_LK12	PC_LK24	SHARO	MSTLRO
		mean	1552.805	1498.664	1594.405	1225.219
		1642.733 p-value	0.02455	1.56E-03	0.009585	8.2E-11
W04	SHARO	config	MR	PC_LK12	PC_LK24	MSTLRO
		mean	1254.492	1002.48	1074.237	1240.939
		1268.885 p-value	0.268528	4.16E-05	1.42E-03	0.008959
N_W01	SHARO	config	MR	PC_LK12	PC_LK24	MSTLRO
		mean	935.3488	781.0203	715.7163	1020.52
		1080.516 p-value	0.008407	9.82E-05	7.55E-08	0.161508
N_W02	MR	config	PC_LK12	PC_LK24	SHARO	MSTLRO
		mean	1499.37	1470.2	1565.25	1504.396
		1641.875 p-value	0.002064	4.81E-04	0.008438	0.00019
N_W03	SHARO	config	MR	PC_LK12	PC_LK24	MSTLRO
		mean	2005.712	1941.13	1943.249	1541.368
		2035.83 p-value	0.328188	0.132769	1.04E-01	2.81E-08
N_W04	MR	config	PC_LK12	PC_LK24	SHARO	MSTLRO
		mean	1154.695	977.7959	1225.453	1101.934
		1478.938 p-value	0.001684	5.92E-08	5.32E-06	8.52E-10

Table 3-15: Paired one-tailed T-Test on the hypothesis that for weekly data splits of W01, W04, N_W01 and N_W03, the best method is SHARO and for the rest of weekly data splits: W02, W03, N_W01 and N_W04, the best method is MR. Each row displays mean and p-value for each configuration, and if p-value > 0.10, then it is highlighted with gray colour.

Considering the last row of Table 3-13, it seems clear that incorporating risk into the objective can bring about consistently better performance in the case of the highly fluctuating data of daily trading. Risk-adjusted methods are now much more prominent, and for the first time we see the Modified Sterling Ratio (MSTLRO) achieving better mean returns than the Sharpe Ratio (SHARO) in 5 data splits (5/8). The BEST RISK method outperformed the BEST PC method all daily splits. The suggestion is that

incorporating-risk objectives apparently can improve performance in more volatile environments.

Statistical tests (Table 3-16) confirm the prominence of MSTLRO in the four daily splits D01, D02, D04, and N_D03, with p-values < 0.10 when compared with the method that gave the second-best mean result in these cases, however for split N_D02 there was no statistically significant difference between MSTLRO and any of MR, PC_LK12, and SHARO). In periods D03 and N_D04, the claim that SHARO was the best method seems justified only in period D03, while in period N_D04 it seems there was not significant difference between SHARO and either MR or MSTLRO. Finally, MR showed the best mean result on N_D01, and T-Test suggests that this was statistically significant.

Period	Best Method	Configurations to compare with the Best Method using one-tailed paired T-Test				
D01	MSTLRO 988.4464	config	MR	PC_LK12	PC_LK24	SHARO
		mean	750.8226	742.5342	752.8806	764.1581
		p-value	2.34E-27	1.47E-16	1.92E-33	4.33E-14
D02	MSTLRO 998.1477	config	MR	PC_LK12	PC_LK24	SHARO
		mean	924.8952	939.5055	926.2618	924.5275
		p-value	1.53E-11	1.43E-08	5.78E-13	4.83E-11
D03	SHARO 994.6272	config	MR	PC_LK12	PC_LK24	MSTLRO
		mean	984.5875	933.9582	992.0299	992.7554
		p-value	0.085075	1.16E-07	2.62E-01	0.079643
D04	MSTLRO 998.1005	config	MR	PC_LK12	PC_LK24	SHARO
		mean	874.7393	876.7977	877.02	869.556
		p-value	8.61E-10	1.04E-08	6.20E-08	2.23E-10
N_D01	MR 1092.446	config	PC_LK12	PC_LK24	SHARO	MSTLRO
		mean	1032.984	1066.217	1087.065	1039.891
		p-value	1.2E-08	9.76E-06	0.076521	9.81E-05
N_D02	MSTLRO 1010.203	config	MR	PC_LK12	PC_LK24	SHARO
		mean	1009.926	1005.159	1002.273	1009.358
		p-value	0.446758	0.274933	1.54E-01	0.275898
N_D03	MSTLRO 977.3706	config	MR	PC_LK12	PC_LK24	SHARO
		mean	956.7973	924.5394	968.2754	954.5288
		p-value	4.57E-05	4.27E-08	6.17E-03	3.23E-07
N_D04	SHARO 1006.428	config	MR	PC_LK12	PC_LK24	MSTLRO
		mean	1005.624	948.6596	939.742	1004.915
		p-value	0.450187	0.000117	7.32E-04	0.442229

Table 3-16: Paired one-tailed T-Test on the hypothesis that for five daily data splits of D01, D02, D04, N_D02 and N_D03, the best method is MSTLRO, for the two daily data splits of D03 and N_D04, the best method is SHARO and

MR is the best method in the last daily data split of N_D01. Each row displays mean and p-value for each configuration, and if p-value > 0.10 , then it is highlighted with gray colour.

3.5 Rule Analysis

Given the promising trading performance often found in the results we have described, it seems appropriate to examine the details of the trading rules themselves. We performed some basic analyses to find any common or frequent structures that appear in the evolved trading rules. Our analysis by generating frequency tables for each of 2 levels, Level1 (concerning leaf nodes) and Level2 (subtrees with depth 2), with details as follows:

- **Level1** – we generate a frequency table for this level by counting the numbers of each technical indicator occurring at leaf nodes, and then sorting them and selecting only the top 10 highest frequency cases to show in the table. Those indicators not among the top 10 are aggregated into ‘Others’ in the table. The items of interest for a Level1 table are those surrounded by red circles in the illustration of Figure 3-12.
- **Level2** – The frequency table for this level is generated by counting subtrees with depth 2. Again, in the example illustration in Figure 3-12, the items of interest for Level2 table are surrounded by blue dashed-line boxes.

Separate such rule analyses have been done for each of the fitness functions considered in this chapter, and for each of monthly, weekly and daily trading. The raw findings and observations from these analyses are given in the following subsections.

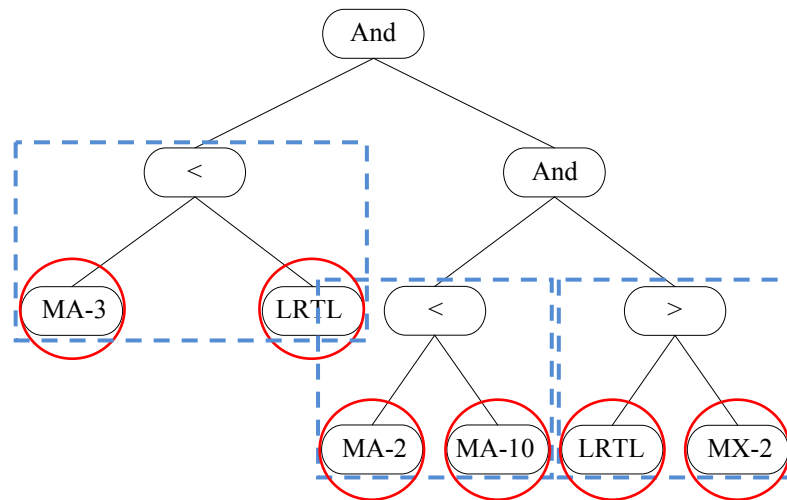


Figure 3-12: Rule analysis at Level1 and Level2; there are 6 technical indicators for Level1 (red circles) and 3 subtrees (at height-2) for Level2 (blue dash-line boxes).

3.5.1 Rule Analysis for Monthly Trading

3.5.1.1 Profit Driven Approach

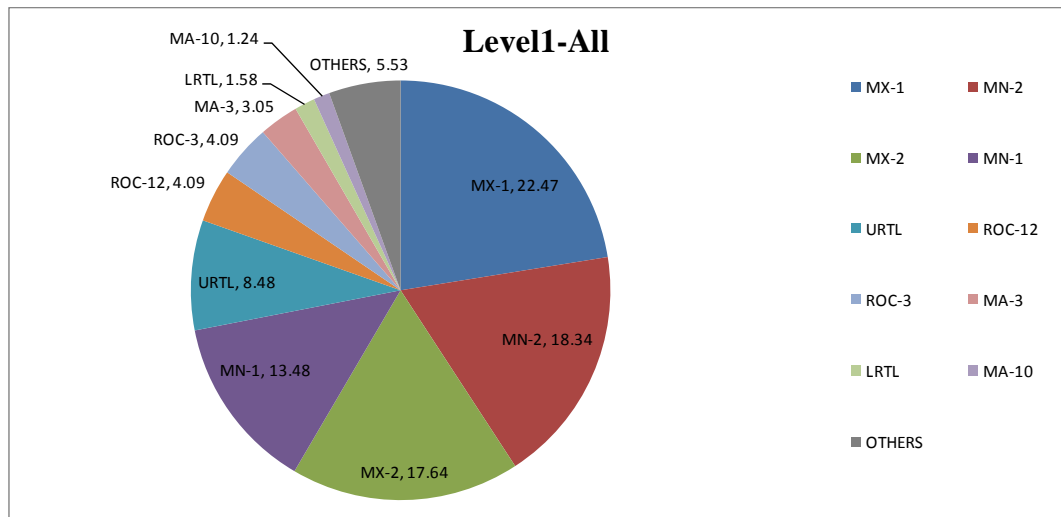


Figure 3-13: Level1 Monthly Proportion-Profit Driven Approach.

Level1				
#	All	MR	PC-LK12	PC-LK24
1	MX-1 (22.47)	MX-1 (25.59)	MX-1 (24.76)	MX-2 (22.29)
2	MN-2 (18.34)	MX-2 (20.96)	MN-2 (22.01)	MX-1 (17.52)
3	MX-2 (17.64)	MN-2 (19.15)	ROC-12 (7.61)	MN-1 (15.96)
4	MN-1 (13.48)	MN-1 (14.21)	ROC-3 (7.61)	MN-2 (15.32)
5	URTL (8.48)	URTL (9.34)	MN-1 (7.61)	URTL (8.53)
6	ROC-12 (4.09)	MA-3 (3.38)	URTL (6.63)	ROC-3 (4.59)
7	ROC-3 (4.09)	ROC-3 (1.96)	MA-3 (5.34)	ROC-12 (4.59)
8	MA-3 (3.05)	ROC-12 (1.96)	MA-10 (3.72)	LRTL (2.39)
9	LRTL (1.58)	LRTL (1.41)	VOL<T> (3.24)	MA-6 (1.38)
10	MA-10 (1.24)	CPRICE<T-1> (0.31)	VOL<T-1> (3.24)	MA-3 (1.38)
OTHERS	OTHERS (5.53)	OTHERS (1.73)	OTHERS (8.25)	OTHERS (6.06)

Table 3-17: Level1 Monthly Table-Profit Driven Approach.

The findings of the rule analysis when considering only the ‘profit-driven’ approaches on monthly trading (MR and PC) are given in Figure 3-13 and Table 3-17. As we can see the most active technical indicators are MX-1, MN-2, MX-2 and MN-1 respectively. These are indicators that consider recent local minima and local maxima, and together they appear in roughly 70% of the rules that emerge from the monthly trading experiments when the profit-driven fitness functions were used. To see more clearly the relationship between those indicators in this context, we need to consider the results at the higher level (Figure 3-14 and Table 3-18). This reveals that “MN-2 < MX-1” is the most common expression used in the selected trading rules, accounting for 22.07% of all depth-2 trees, while the second most-frequent is “MX-2 < MN1” (17.97%). It should be noted that there are 24 possible configurations of this binary relational expression in the form “TI1 *operator* TI2”, where TI1 and TI2 are technical indicators concerning recent local minima and maxima (MN-1, MN-2, MX-1 and MX-2) and *operator* is either < or >.

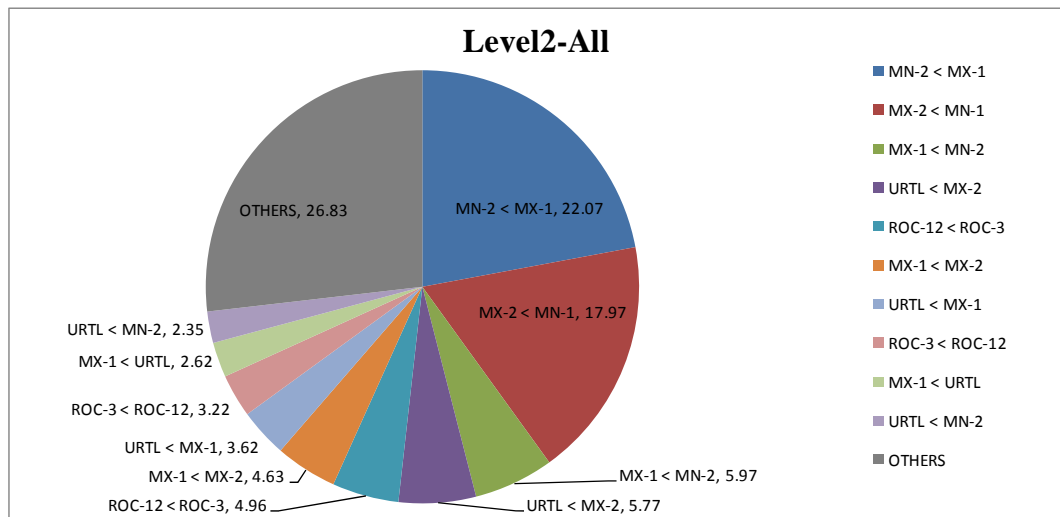


Figure 3-14: Level2 Monthly Proportion-Profit Driven Approach.

Level2				
#	All	MR	PC-LK12	PC-LK24
1	MN-2 < MX-1 (22.07)	MN-2 < MX-1 (24.33)	MN-2 < MX-1 (24.27)	MX-2 < MN-1 (21.47)
2	MX-2 < MN-1 (17.97)	MX-2 < MN-1 (23.23)	MX-1 < MN-2 (12.62)	MN-2 < MX-1 (18.17)
3	MX-1 < MN-2 (5.97)	URL < MX-2 (7.06)	ROC-3 < ROC-12 (9.39)	URL < MX-2 (7.34)
4	URL < MX-2 (5.77)	MX-1 < MX-2 (6.28)	MX-1 < URL (6.80)	ROC-12 < ROC-3 (6.61)
5	ROC-12 < ROC-3 (4.96)	URL < MX-1 (5.81)	ROC-12 < ROC-3 (5.83)	MX-1 < MX-2 (5.14)
6	MX-1 < MX-2 (4.63)	MX-1 < MN-2 (5.18)	MA-3 < MN-1 (5.83)	MN-1 < MX-2 (3.85)
7	URL < MX-1 (3.62)	MX-1 < MA-3 (3.92)	VOL<T> < VOL<T-1> (4.53)	MX-1 < MN-2 (3.12)
8	ROC-3 < ROC-12 (3.22)	URL < MN-2 (3.45)	MA-10 < MN-1 (2.59)	URL < MX-1 (2.94)
9	MX-1 < URL (2.62)	ROC-12 < ROC-3 (3.14)	MA-3 < MX-1 (1.94)	ROC-3 < ROC-12 (2.57)
10	URL < MN-2 (2.35)	MX-2 < MX-1 (1.88)	VOL<T-1> < VOL<T> (1.94)	MX-1 < URL (2.02)
OTHERS	OTHERS (26.83)	OTHERS (15.70)	OTHERS (24.27)	OTHERS (26.79)

Table 3-18: Level2 Monthly Table-Profit Driven Approach.

3.5.1.2 Risk-Adjusted Approach

When we look at the risk-adjusted fitness measures, the local minima and local maxima indicators are still the most active, appearing approximately 65% of the time in the selected trading rules as shown in Figure 3-15 and Table 3-19. When considering the connections between those indicators at level 2, we find the reverse of our finding from the profit-driven approaches: the expression “MX-2 < MN-1” is the second most active in the profit driven approach, but is now the most common at 26.06%, whereas “MN-2 < MX-1”, the most active in the profit-driven approach, turns out to be rather less common here at 11.27% (see Figure 3-16 and Table 3-20).

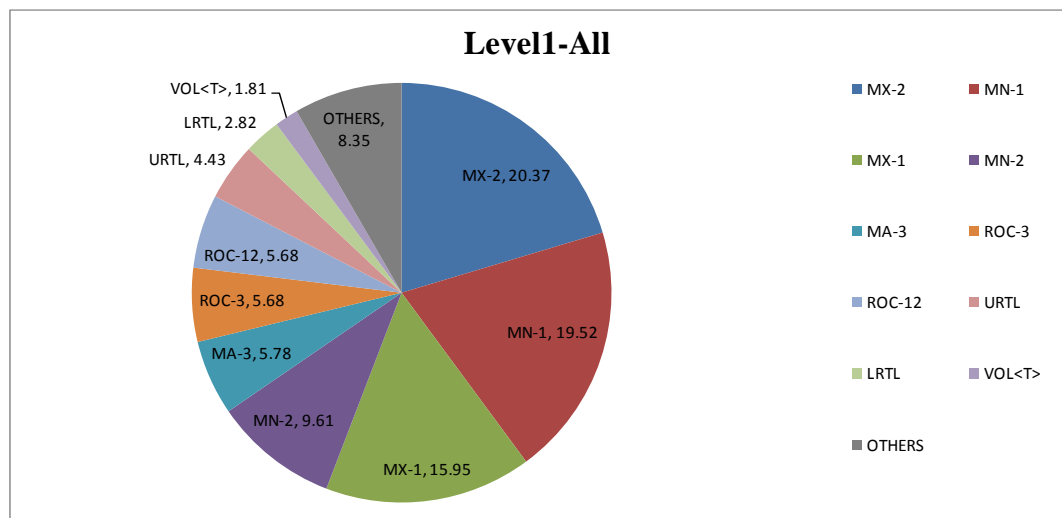


Figure 3-15: Level1 Monthly Proportion-Risk-Adjusted Approach.

Level1			
#	All	MSTLRO	SHARO
1	MX-2 (20.37)	MN-1 (18.31)	MX-2 (24.54)
2	MN-1 (19.52)	MX-2 (15.46)	MX-1 (22.21)
3	MX-1 (15.95)	MX-1 (8.55)	MN-1 (20.54)
4	MN-2 (9.61)	ROC-12 (7.68)	MN-2 (12.17)
5	MA-3 (5.78)	ROC-3 (7.68)	MA-3 (6.88)
6	ROC-3 (5.68)	MN-2 (6.58)	ROC-3 (4.00)
7	ROC-12 (5.68)	URLT (6.25)	ROC-12 (4.00)
8	URLT (4.43)	MA-3 (4.50)	URLT (2.88)
9	LRTL (2.82)	LRTL (4.28)	LRTL (1.58)
10	VOL<T> (1.81)	VOL<T-1> (3.84)	CPRICE<T-1> (0.37)
OTHERS	OTHERS (8.35)	OTHERS (16.89)	OTHERS (0.84)

Table 3-19: Level1 Monthly Table- Risk-Adjusted Approach.

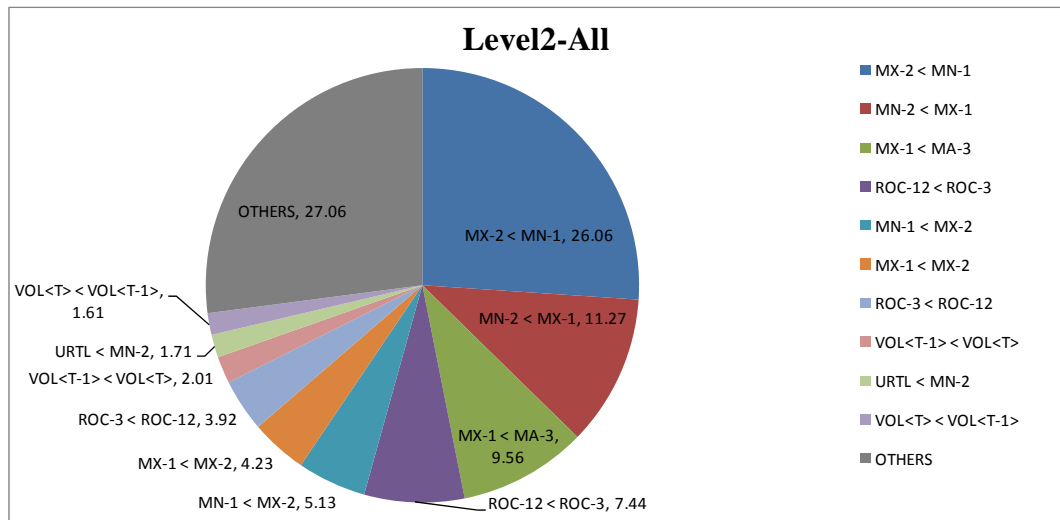


Figure 3-16: Level2 Monthly Proportion- Risk-Adjusted Approach.

Level2			
#	All	MSTLRO	SHARO
1	MX-2 < MN-1 (26.06)	MX-2 < MN-1 (20.39)	MX-2 < MN-1 (30.86)
2	MN-2 < MX-1 (11.27)	ROC-12 < ROC-3 (8.99)	MN-2 < MX-1 (16.73)
3	MX-1 < MA-3 (9.56)	MX-1 < MA-3 (6.36)	MX-1 < MA-3 (12.27)
4	ROC-12 < ROC-3 (7.44)	ROC-3 < ROC-12 (6.36)	MX-1 < MX-2 (7.43)
5	MN-1 < MX-2 (5.13)	MN-1 < MX-2 (5.26)	ROC-12 < ROC-3 (6.13)
6	MX-1 < MX-2 (4.23)	MN-2 < MX-1 (4.82)	MN-1 < MX-2 (5.02)
7	ROC-3 < ROC-12 (3.92)	VOL<T-1> < VOL<T> (4.17)	MX-1 < MN-2 (2.42)
8	VOL<T-1> < VOL<T> (2.01)	VOL<T> < VOL<T-1> (3.51)	MX-2 < MX-1 (2.04)
9	URLT < MN-2 (1.71)	MA-10 < CPRICE<T> (3.07)	ROC-3 < ROC-12 (1.86)
10	VOL<T> < VOL<T-1> (1.61)	URLT < MN-2 (2.41)	URLT < MX-1 (1.86)
OTHERS	OTHERS (27.06)	OTHERS (34.65)	OTHERS (13.38)

Table 3-20: Level2 Monthly Table- Risk-Adjusted Approach.

3.5.2 Rule Analysis for Weekly Trading

3.5.2.1 Profit Driven Approach

For the more volatile environment of weekly trading, the local minima and local maxima indicators group are able to maintain their prominence at about 56% (see

Figure 3-17). Moreover, the trend line indicators (LRTL and URTL), which themselves are calculated on the basis of the local minima and local maxima, now have more active roles, with URTL appearing in 11.29% and LRTL appearing in 8.69% of evolved rules. This is reflected in the results of the Level2 analysis, where the first and second most active expressions, “MX-2 < MN-1” and “MN2 < MX-1”, together appear 23.82% of the time, while the third most active depth-2 tree is “MX-1 < URTL” (7.26%) (see Figure 3-18).

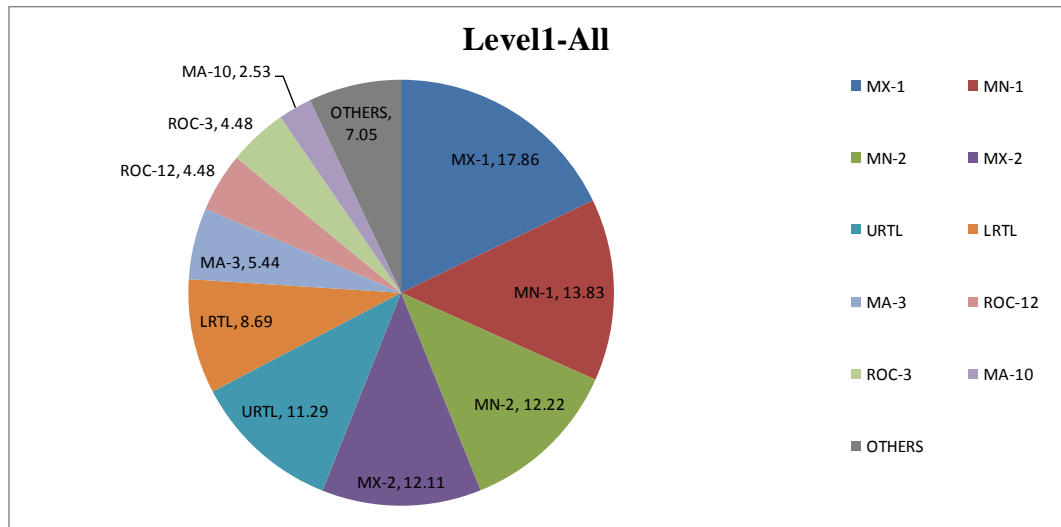


Figure 3-17: Level1 Weekly Proportion-Profit Driven Approach.

Level1				
#	All	MR	PC-LK12	PC-LK24
1	MX-1 (17.86)	MX-1 (19.16)	MX-1 (18.56)	MX-1 (15.62)
2	MN-1 (13.83)	MN-1 (16.76)	MN-2 (11.96)	URTL (14.86)
3	MN-2 (12.22)	MX-2 (16.04)	MN-1 (11.96)	MN-2 (12.80)
4	MX-2 (12.11)	MN-2 (11.94)	MX-2 (10.48)	MN-1 (12.04)
5	URTL (11.29)	LRTL (10.78)	URTL (9.23)	MX-2 (8.89)
6	LRTL (8.69)	URTL (9.98)	LRTL (8.88)	ROC-3 (7.05)
7	MA-3 (5.44)	MA-3 (6.15)	MA-3 (5.92)	ROC-12 (7.05)
8	ROC-12 (4.48)	MA-10 (3.65)	ROC-12 (5.81)	LRTL (5.97)
9	ROC-3 (4.48)	MA-6 (1.52)	ROC-3 (5.81)	MA-3 (4.12)
10	MA-10 (2.53)	ROC-3 (1.34)	MA-10 (3.08)	CPRICE<T> (2.82)
OTHERS	OTHERS (7.05)	OTHERS (2.67)	OTHERS (8.31)	OTHERS (8.79)

Table 3-21: Level1 Weekly Table-Profit Driven Approach.

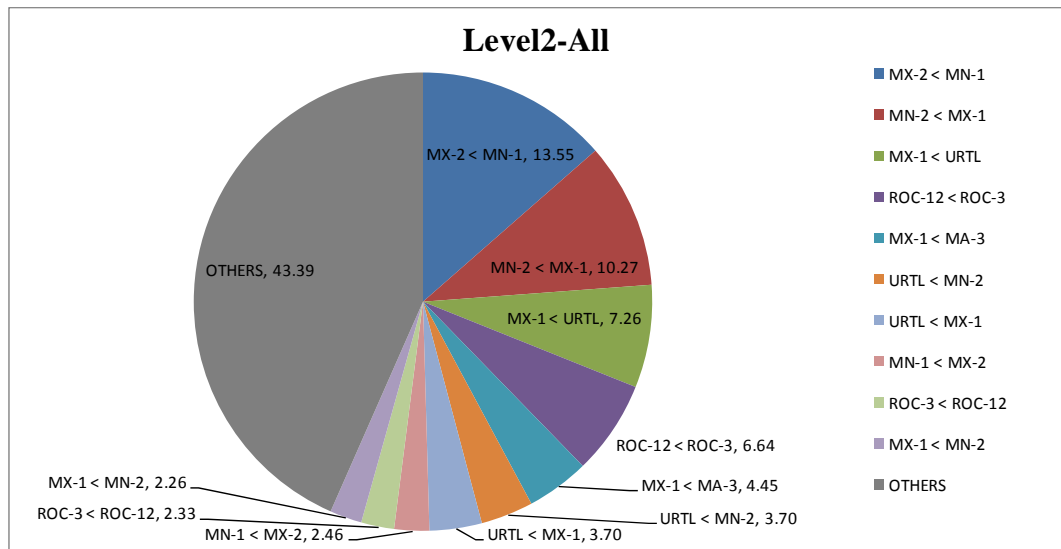


Figure 3-18: Level2 Weekly Proportion-Profit Driven Approach.

Level2				
#	All	MR	PC-LK12	PC-LK24
1	MX-2 < MN-1 (13.55)	MX-2 < MN-1 (19.79)	MX-2 < MN-1 (11.39)	MX-1 < URTL (11.28)
2	MN-2 < MX-1 (10.27)	MN-2 < MX-1 (11.94)	MN-2 < MX-1 (10.02)	ROC-12 < ROC-3 (10.20)
3	MX-1 < URTL (7.26)	MX-1 < MA-3 (6.60)	ROC-12 < ROC-3 (8.88)	MN-2 < MX-1 (8.46)
4	ROC-12 < ROC-3 (6.64)	LRTL < MA-10 (4.46)	MX-1 < URTL (7.52)	MX-2 < MN-1 (8.03)
5	MX-1 < MA-3 (4.45)	URLT < MN-2 (4.28)	MX-1 < MA-3 (4.56)	URLT < MN-2 (6.07)
6	URLT < MN-2 (3.70)	URLT < MX-1 (4.10)	URLT < MX-1 (4.33)	ROC-3 < ROC-12 (3.90)
7	URLT < MX-1 (3.70)	MX-1 < URTL (3.74)	ROC-3 < ROC-12 (2.73)	URLT < MX-1 (2.60)
8	MN-1 < MX-2 (2.46)	MX-1 < MN-2 (3.21)	MN-1 < MX-2 (2.51)	LRTL < URTL (1.95)
9	ROC-3 < ROC-12 (2.33)	MN-1 < MX-2 (3.21)	LRTL < MX-1 (2.51)	MN-1 < MA-3 (1.95)
10	MX-1 < MN-2 (2.26)	MA-3 < MX-1 (2.50)	MA-3 < MX-1 (2.28)	CPRICE<T> < MN-1 (1.95)
OTHERS	OTHERS (43.39)	OTHERS (36.19)	OTHERS (43.28)	OTHERS (43.60)

Table 3-22: Level2 Weekly Table-Profit Driven Approach.

3.5.2.2 Risk-Adjusted Approach

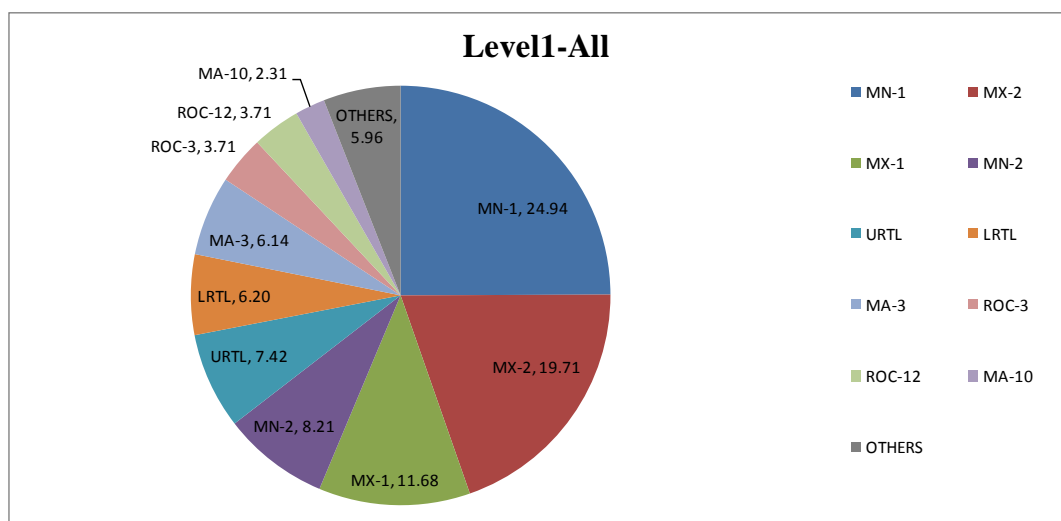


Figure 3-19: Level1 Weekly Proportion-Risk-Adjusted Approach.

Level1			
#	All	MSTLRO	SHARO
1	MN-1 (24.94)	MN-1 (24.81)	MN-1 (25.06)
2	MX-2 (19.71)	MX-2 (18.32)	MX-2 (20.98)
3	MX-1 (11.68)	MX-1 (9.54)	MX-1 (13.64)
4	MN-2 (8.21)	URTL (7.00)	MN-2 (10.26)
5	URTL (7.42)	ROC-3 (6.49)	LRTL (7.93)
6	LRTL (6.20)	ROC-12 (6.49)	URTL (7.81)
7	MA-3 (6.14)	MN-2 (5.98)	MA-3 (7.11)
8	ROC-3 (3.71)	MA-3 (5.09)	MA-10 (3.38)
9	ROC-12 (3.71)	LRTL (4.33)	ROC-12 (1.17)
10	MA-10 (2.31)	VOL<T-1> (2.42)	ROC-3 (1.17)
OTHERS	OTHERS (5.96)	OTHERS (9.54)	OTHERS (1.52)

Table 3-23: Level1 Weekly Table- Risk-Adjusted Approach.

As with the profit driven approach, Figure 3-19 shows that, when we look at the prominent indicators for risk-adjusted fitness functions on weekly trading, around 64% of the technical indicators were again from the local minima and local maxima group. In the Level2 analysis (from Figure 3-20), we find that the depth-2 tree with most prominence was “MX-2 < MN-1” at 28.47%; this proportion is quite high when compared to 13.55%, which was the highest proportion of any depth-2 tree in the profit driven approaches. Moreover, the second ranked case, “MX1 < MA3”, represents the first appearance so far of a moving-average indicator appearing in the most prominent structures.

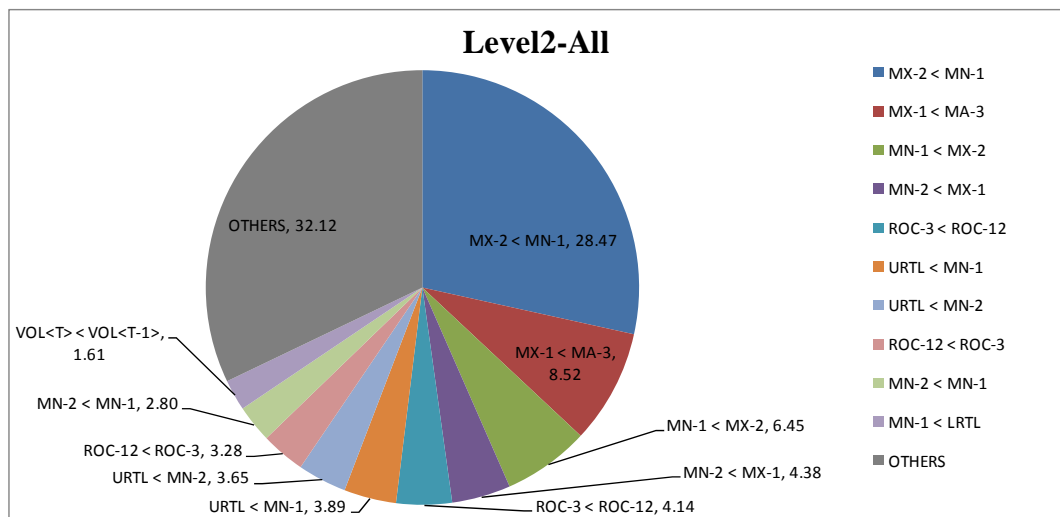


Figure 3-20: Level2 Weekly Proportion- Risk-Adjusted Approach.

Level2			
#	All	MSTLRO	SHARO
1	MX-2 < MN-1 (28.47)	MX-2 < MN-1 (25.70)	MX-2 < MN-1 (31.00)
2	MX-1 < MA-3 (8.52)	MX-1 < MA-3 (8.65)	MX-1 < MA-3 (8.39)
3	MN-1 < MX-2 (6.45)	MN-1 < MX-2 (7.38)	URTL < MN-2 (6.06)
4	MN-2 < MX-1 (4.38)	ROC-3 < ROC-12 (7.38)	MN-2 < MX-1 (5.83)
5	ROC-3 < ROC-12 (4.14)	URTL < MN-1 (6.62)	MN-1 < MX-2 (5.59)
6	URTL < MN-1 (3.89)	ROC-12 < ROC-3 (5.60)	MN-1 < LRTL (3.96)
7	URTL < MN-2 (3.65)	MN-2 < MN-1 (3.82)	LRTL < MA-10 (3.50)
8	ROC-12 < ROC-3 (3.28)	VOL<T> < VOL<T-1> (2.80)	LRTL < MX-1 (2.80)
9	MN-2 < MN-1 (2.80)	MN-2 < MX-1 (2.80)	MX-1 < MX-2 (2.33)
10	MN-1 < LRTL (2.31)	VOL<T-1> < VOL<T> (2.04)	MN-2 < MN-1 (1.86)
OTHERS	OTHERS (32.12)	OTHERS (27.23)	OTHERS (28.67)

Table 3-24: Level2 Weekly Table- Risk-Adjusted Approach.

3.5.3 Rule Analysis for Daily Trading

3.5.3.1 Profit Driven Approach

When we look at frequent structures found in evolved daily trading rules, the situation seems to change markedly, with Figure 3-21 and Table 3-25 revealing that there is no prominent case of a certain technical indicators being used significantly more than others. The results from Figure 3-22 also confirm this, with the top six indicators having not much difference between them, varying from 10.60% down to 7.04%. It is notable that those six top slots cover over 50% of the overall proportion. However, interestingly in the Level2 analysis, we find that rate of change indicators occupied 19.37% of binary expressions (Figure 3-22), which is high compared with the cases of monthly and weekly trading.

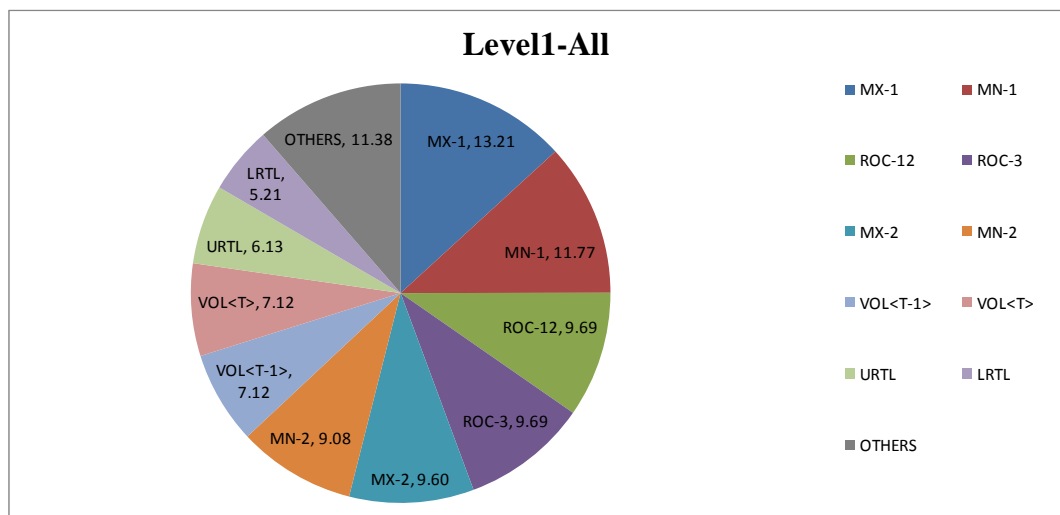


Figure 3-21: Level1 Daily Proportion-Profit Driven Approach.

Level1				
#	All	MR	PC-LK12	PC-LK24
1	MX-1 (13.21)	MX-1 (15.56)	MN-1 (12.10)	MX-1 (11.89)
2	MN-1 (11.77)	MN-1 (14.03)	MX-1 (12.10)	ROC-3 (10.92)
3	ROC-12 (9.69)	MX-2 (13.14)	MX-2 (9.65)	ROC-12 (10.92)
4	ROC-3 (9.69)	MN-2 (10.46)	ROC-3 (9.65)	VOL<T> (9.95)
5	MX-2 (9.60)	ROC-3 (8.42)	ROC-12 (9.65)	VOL<T-1> (9.95)
6	MN-2 (9.08)	ROC-12 (8.42)	MN-2 (9.22)	MN-1 (9.34)
7	VOL<T-1> (7.12)	VOL<T> (5.87)	LRTL (8.36)	URTL (8.98)
8	VOL<T> (7.12)	VOL<T-1> (5.87)	VOL<T-1> (5.19)	MN-2 (7.65)
9	URTL (6.13)	LRTL (4.85)	VOL<T> (5.19)	MX-2 (6.19)
10	LRTL (5.21)	URTL (4.08)	URTL (5.04)	LRTL (2.91)
OTHERS	OTHERS (11.38)	OTHERS (9.31)	OTHERS (13.83)	OTHERS (11.29)

Table 3-25: Level1 Daily Table-Profit Driven Approach.

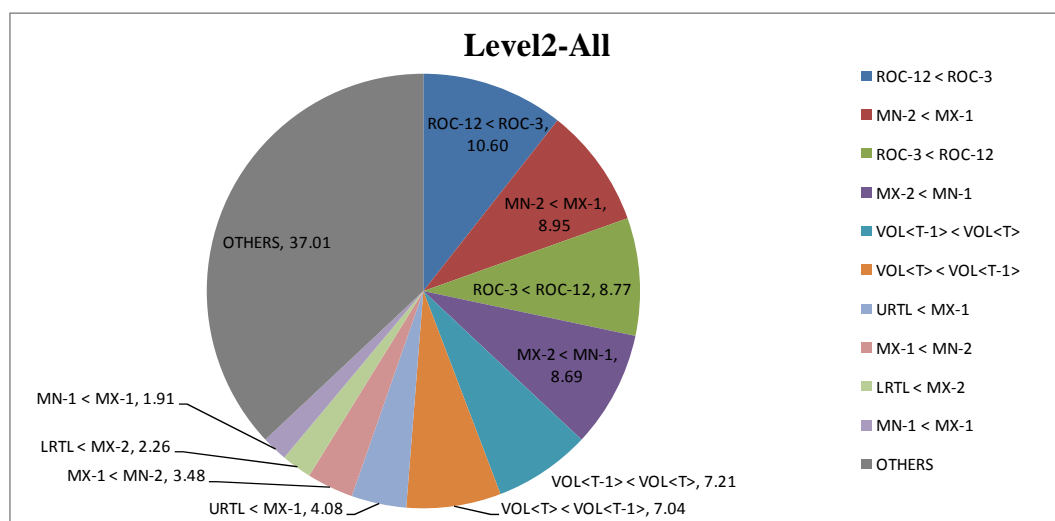


Figure 3-22: Level2 Daily Proportion-Profit Driven Approach.

Level2				
#	All	MR	PC-LK12	PC-LK24
1	ROC-12 < ROC-3 (10.60)	MX-2 < MN-1 (12.50)	ROC-12 < ROC-3 (12.10)	ROC-12 < ROC-3 (13.59)
2	MN-2 < MX-1 (8.95)	ROC-3 < ROC-12 (10.71)	MN-2 < MX-1 (10.95)	VOL<T> < VOL<T-1> (10.19)
3	ROC-3 < ROC-12 (8.77)	MN-2 < MX-1 (10.71)	MX-2 < MN-1 (8.36)	VOL<T-1> < VOL<T> (9.71)
4	MX-2 < MN-1 (8.69)	VOL<T-1> < VOL<T> (6.38)	ROC-3 < ROC-12 (7.20)	ROC-3 < ROC-12 (8.25)
5	VOL<T-1> < VOL<T> (7.21)	ROC-12 < ROC-3 (6.12)	VOL<T-1> < VOL<T> (5.19)	URTL < MX-1 (8.25)
6	VOL<T> < VOL<T-1> (7.04)	VOL<T> < VOL<T-1> (5.36)	VOL<T> < VOL<T-1> (5.19)	MN-2 < MX-1 (5.58)
7	URTL < MX-1 (4.08)	MN-1 < MX-1 (5.36)	LRTL < MN-1 (4.32)	MX-2 < MN-1 (5.34)
8	MX-1 < MN-2 (3.48)	MX-1 < MN-2 (4.59)	LRTL < MX-2 (3.46)	MX-1 < MN-2 (3.40)
9	LRTL < MX-2 (2.26)	MN-1 < MX-2 (3.06)	MA-3 < MN-1 (3.17)	URTL < MN-1 (1.46)
10	MN-1 < MX-1 (1.91)	LRTL < MX-2 (2.81)	MX-1 < MN-2 (2.31)	MA-3 < MX-1 (1.46)
OTHERS	OTHERS (37.01)	OTHERS (32.40)	OTHERS (37.75)	OTHERS (32.77)

Table 3-26: Level2 Daily Table-Profit Driven Approach.

3.5.3.2 Risk-Adjusted Approach

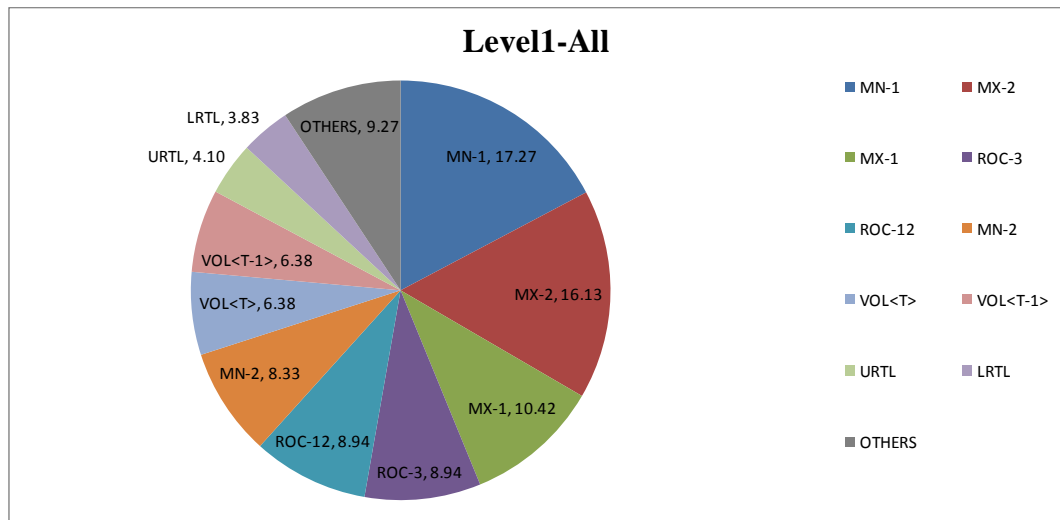


Figure 3-23: Level1 Daily Proportion-Risk-Adjusted Approach.

Level1			
#	All	MSTLRO	SHARO
1	MN-1 (17.27)	MN-1 (22.17)	MX-2 (13.48)
2	MX-2 (16.13)	MX-2 (19.35)	MX-1 (13.48)
3	MX-1 (10.42)	ROC-3 (8.48)	MN-1 (13.24)
4	ROC-3 (8.94)	ROC-12 (8.48)	ROC-12 (9.31)
5	ROC-12 (8.94)	MN-2 (8.04)	ROC-3 (9.31)
6	MN-2 (8.33)	MX-1 (6.70)	MN-2 (8.58)
7	VOL<T> (6.38)	URTL (5.21)	VOL<T-1> (8.33)
8	VOL<T-1> (6.38)	VOL<T> (4.02)	VOL<T> (8.33)
9	URTL (4.10)	VOL<T-1> (4.02)	LRTL (4.78)
10	LRTL (3.83)	LRTL (2.68)	URTL (3.19)
OTHERS	OTHERS (9.27)	OTHERS (10.86)	OTHERS (7.97)

Table 3-27: Level1 Daily Table- Risk-Adjusted Approach.

Now considering the rule structure analysis for risk-adjusted approaches in daily trading, we see some similarities with the case of the profit-driven approaches, with the top six indicators accounting for over 70% (see Figure 3-23). On the other hand, the highest proportion for any Level2 structure (in Figure 3-24) is 19.22%, which is considerably higher than in the profit driven case (10.60%).

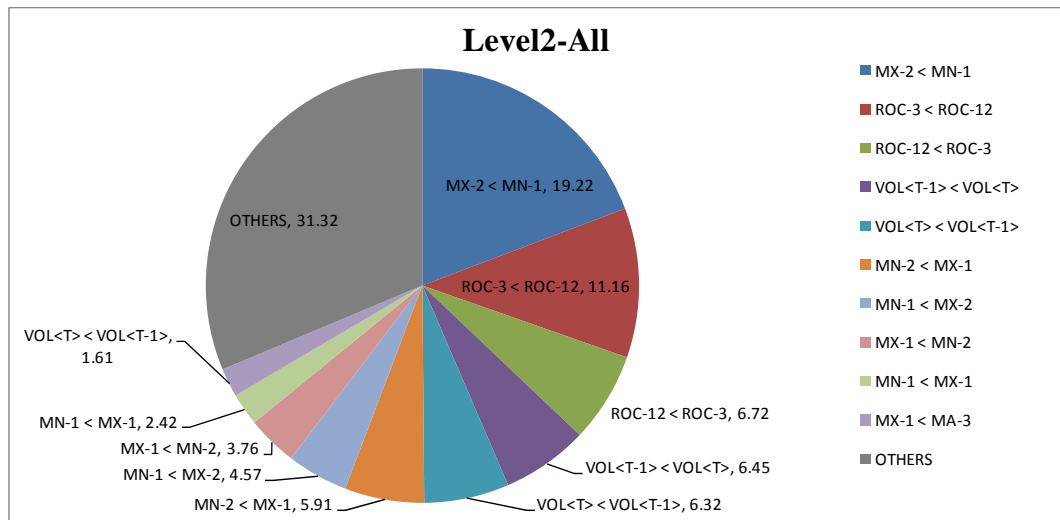


Figure 3-24: Level2 Daily Proportion- Risk-Adjusted Approach.

Level2			
#	All	MSTLRO	SHARO
1	MX-2 < MN-1 (19.22)	MX-2 < MN-1 (27.38)	MX-2 < MN-1 (12.50)
2	ROC-3 < ROC-12 (11.16)	ROC-3 < ROC-12 (11.31)	ROC-3 < ROC-12 (11.03)
3	ROC-12 < ROC-3 (6.72)	MN-1 < MX-2 (5.95)	VOL<T-1> < VOL<T> (9.07)
4	VOL<T-1> < VOL<T> (6.45)	ROC-12 < ROC-3 (5.65)	VOL<T> < VOL<T-1> (7.60)
5	VOL<T> < VOL<T-1> (6.32)	VOL<T> < VOL<T-1> (4.76)	ROC-12 < ROC-3 (7.60)
6	MN-2 < MX-1 (5.91)	MN-2 < MX-1 (4.46)	MN-2 < MX-1 (7.11)
7	MN-1 < MX-2 (4.57)	VOL<T-1> < VOL<T> (3.27)	MX-1 < MN-2 (4.90)
8	MX-1 < MN-2 (3.76)	URL < MN-2 (2.68)	MN-1 < MX-1 (4.41)
9	MN-1 < MX-1 (2.42)	MX-1 < MA-3 (2.38)	MN-1 < MX-2 (3.43)
10	MX-1 < MA-3 (2.15)	MX-1 < MN-2 (2.38)	LRTL < MX-2 (3.19)
OTHERS	OTHERS (31.32)	OTHERS (29.76)	OTHERS (29.17)

Table 3-28: Level2 Daily Table- Risk-Adjusted Approach.

3.6 Discussion and Conclusions

The discovery of technical trading rules by Genetic Programming is an interesting and active thread of research, in which a string of research articles have been published in the past decade, exhibiting varied levels of success. The chief difficulty is that, although successful rules can be found, these are often not competitive with straightforward “buy and hold” (in upwardly moving markets) or the exploitation of risk-free investments (in downward markets). Building on Allen and Karjalainen’s work in 1999 [2], however, Becker and Seshadri’s approach [6, 8] was one of few so far that have shown more promise, and we have replicated and explored that work further in this chapter to discover guidelines on how reliably to generate effective rules. With modification to the approach used for choosing the rule to evaluate by using a validation set, our experiments find that the Becker and Seshadri’s variations on the original AK approach can provide fairly robust generation of rules which outperform buy-and-hold. Having said that, we found that the basic setup used in [6, 8] is sensitive to the data periods involved, and it is clearly better to use a validation set to choose the rule (rule-selection regime 2). This fact is revealed evidently in 8 different monthly data-split scenarios, one involving an upward market during the test period, five splits having mildly volatile movement, and the other two being more volatile. However, this was in the context of monthly trading. It is reasonable to suppose that Becker and Seshadri’s approach might be a salient factor in the ability to beat a buy-and-hold strategy for more volatile data. In consequence, we examined this by testing Becker and Seshadri’s approach in the context of each of weekly and daily trading. While we found robust technical trading rules which outperform buy-and-hold in the case of monthly trading, it turned out that it was relatively less likely to obtain reliable and consistent trading rules for daily trading, and the situation for weekly trading is somewhere in between. In more detail, it seems that this approach is capable of finding rules that outperform buy-and-hold, even when

tested in upwardly-moving markets, but the performance depends on the data split, and as we move from monthly to daily trading, this dependence on the data split seems to increase sharply. Having said that, in summary, our results indicated that regime 2 has shown its capability to perform consistently and robustly than regime 1 in each of monthly, weekly and daily trading.

We consider now the performance of the different fitness functions, focussing on the ‘regime 2’ rule selection scheme. In the monthly trading environment, Market Return (MR) consistently gave the highest returns, but its performance dropped significantly in the more volatile environments of weekly and daily data. This implies that a profit driven method is less reliable in a more constantly changing environment. In the weekly and daily trading situations, incorporating risk measures into the fitness model, for instance the Sharpe Ratio, showed distinct benefits. Considering risk measures seems to be a key factor for performing well and robustly in volatile environments.

Concerning period length in the Performance Consistency (PC) fitness function, in our experiments the most robust PC period length in monthly and daily trading was the longest period – PC_LK24 (meaning 24 months and 24 days respectively) whereas for weekly trading the PC period with the shortest length seemed best (PC_LK12 – 12 weeks).

From our attempt to grasp the basic structures that appeared in our evolved technical trading rules, we found that technical indicators from the group of local minima and maxima were the most active in all cases of monthly and weekly trading, and were joined in weekly trading by an additional active group of trend line indicators. Overall, the most active Level2 expressions in both monthly and weekly environments were “MN-2 < MX-1” and “MX-2 < MN-1”. However, in daily trading there was no

particularly prominent indicator. Technical indicators in daily trading rules were found from various groups of indicators – no group dominating others.

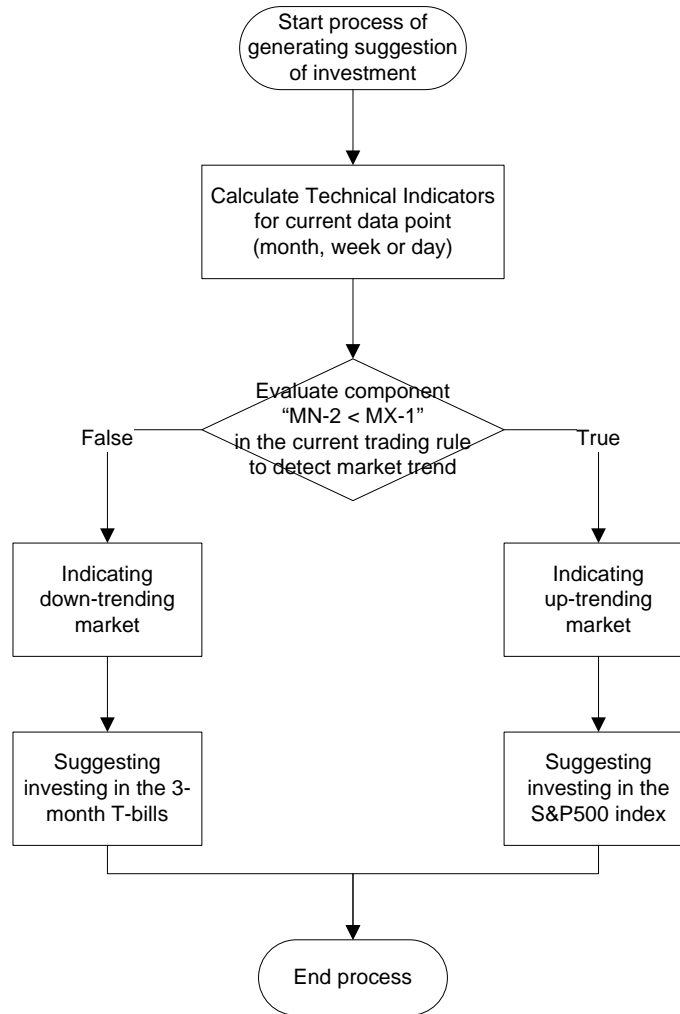


Figure 3-25: The suggestion of investment by evaluating the component “MN-2 < MX-1” in the trading rule.

For an overall summary, we find that “MN-2 < MX-1” and “MX-2 < MN-1” are important. These come out strongly from the rule analysis. It suggests that we might have found interesting new and simple rules that could be of general use as technical trading rules. Let us consider the first component-“MN-2 < MX-1”, the simple diagram in Figure 3-25 explains what this component is saying in the current trading rule (See Appendix A.1 for examples of trading rules with the single-objective approach). If the component becomes true, then it indicates an up-trending market as illustrated in the

case 1 of Figure 3-26 and this suggests investing in the S&P500 index. In case that the component returns false, it indicates a downward market as shown in case 2 of Figure 3-26. As a result, this suggests investing in the risk free investment of 3-month T-bills. Similarly, the second component-“MX-2 < MN1” is interpreted in the same way as in Figure 3-25 and it associates with case 3 and case 4 in Figure 3-26 for indicating up-trending and down-trending markets respectively. The reason why these two expressions were the most active components in both cases of monthly and weekly trading may come from the overreaction of investors in the markets. In downwardly moving markets, most investors fear and panic that their holding stocks will go down badly so that they sell those stocks rapidly making stock prices fall downward below their intrinsic values. In an upwardly trending market, investors are driven by greed to yield massive profits from their holding stocks, resulting in stock prices moving upward above intrinsic equity values. Making profits from the behaviour of market participants in financial markets rather than trading on market fundamentals can lead to market bubbles. The study in [80] also confirms these market behaviours due to market participants’ emotions.

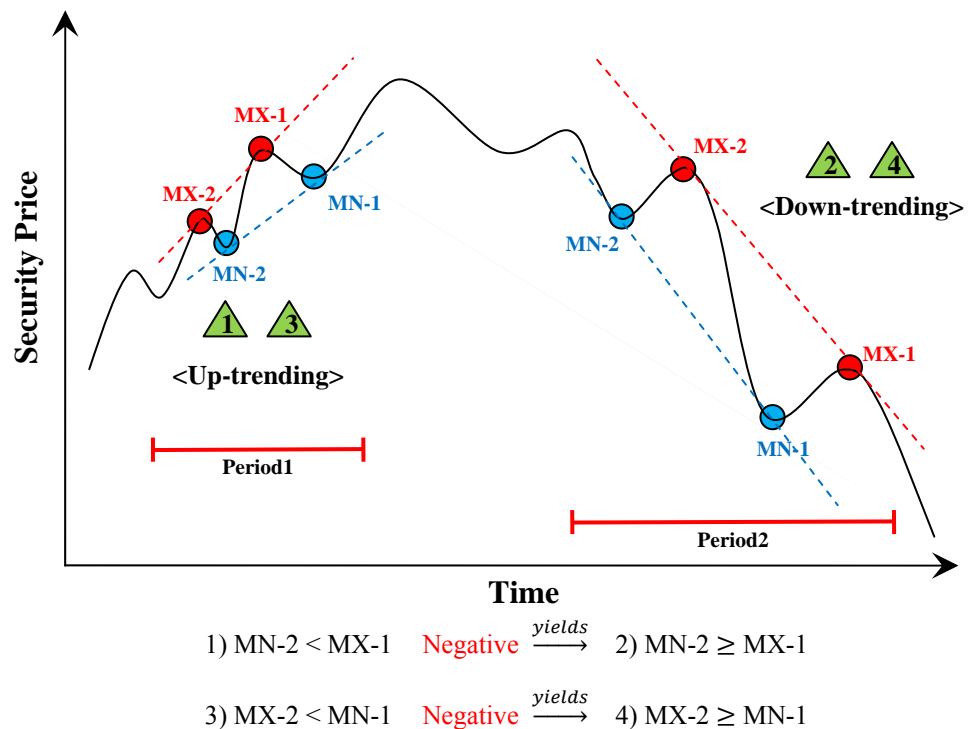


Figure 3-26: Indication of up-trending market when case 1: “ $MN-2 < MX-1$ ” and case 3 “ $MX-2 < MN-1$ ” becoming true, and Indication of down-trending market when negative forms of case 1 (“ $MN-2 \geq MX-1$ ”) and case 2 (“ $MX-2 \geq MN-1$ ”) becoming true.

Although the work reported here has some clear limitations, it has included a basic replication, validation and extension of the approach explored in [2, 6, 8], and confirms this line of work as promising for future exploration. Additional directions include the use of multi-objective formalizations, to provide a more principled way to handle both the performance consistency and complexity aspects of the fitness function, should be investigated. We perform such an investigation in the next chapter, along with several other multi-objective configurations, and compare the results with the results in this chapter from single-objective approaches. A comparison and exploration with Grammatical Evolution (GE) also seems a promising direction, since GE is now being explored with success in financial applications [11]. We perform such a comparison in a later chapter, in the wider context of exploring a new approach to GE.

4. Multiobjective Algorithms for Evolved Technical Trading Rules

4.1 Introduction

In the previous chapter we focused on using GP, one of the main branches of Evolutionary Algorithms (EAs), for discovering technical trading rules with single objective approaches. The approaches we used can be classified into two main groups: the profit maximizing approach, and the risk adjustment approach. Here we turn instead to multi-objective approaches, in which we can have two or more objective functions to consider simultaneously. In a multi-objective approach, all objectives are optimised simultaneously, and we are able to consider and explore trade-offs among the objective functions. This brings us more options to explore for evolving sophisticated technical trading rules than in the single objective approach which is restricted to return only a single scalar value as a fitness value.

This chapter provides a comprehensive study to find out whether improved performance at more frequent trading interventions can be achieved by exploiting a multi-objective approach in comparison to a single-objective approach.

The remainder of this chapter is organized as follows:

- Section 4.2 proposes configurations for single and multi-objective approaches which are classified into two groups: profit driven approaches and risk-adjusted approaches. It also provides information on the basic GP algorithm that we use for multi-objective optimisation.
- Section 4.3 gives details of the parameters used in our multi-objective GP algorithm, such as archive size and crossover rate.

- Section 4.4 present the results of our experiments on monthly, weekly and daily data from two angles: the relative performance of the multi-objective strategies over single-objective strategies, and the relative performance of the multi-objective strategies compared with buy and hold.
- Section 4.5 provides a basic analysis of the structures found in the technical trading rules generated from the results in section 4.4
- Finally Section 4.6 is a conclusion including: the main outcomes, comparison between single objective and multi-objective approaches, and comparison between profit-driven and risk-adjusted approaches.

4.2 GP Approach and Multi-objective Characterization

4.2.1 GP Approach

The overall approach we used in this chapter is in many details the same as the overall approach used for single-objective configurations in the previous chapter, but this time we explore the use of GP with a multi-objective methodology. Nevertheless we present a reminder of the basic GP approach with a brief description in this section. The GP tree (see Figure 3-1) comprises two types of binary operators at internal nodes and various kinds of indicators at leaf nodes, and the same function and terminal set as in the single-objective approach are used to construct the GP tree; they are Boolean operators and relational operators for the function set, and six groups of technical indicators for terminal set: Price, Volumes, Moving Averages (MA), Rate of Change (ROC), Price Resistance and Trend Line. When evaluating the GP tree, it returns the result as a Boolean value which is interpreted as a trading signal. If the value is True, then this is a *buy signal*; otherwise it is a *sell signal*. Please refer to sections 3.2.1, 3.2.2 and 3.2.4 for more details of the GP approach used, which applies equally to all the experiments in this chapter.

In the next section we give some details of multi-objective configurations used in the experiments in this chapter. Essentially, what we mean by a ‘configuration’ is a subset of either two or three specific objectives. Each individual objective is essentially the same as the corresponding single-objective method described in the previous chapter. For this reason, details of how to calculate each objective are not given here, since they have been given already in section 3.2.3.

4.2.2 Single and Multi-objective Approaches

The experiments in this chapter are mainly classified into two groups: (a) profit driven approaches that reward trading rules on the basis of their returns, and (b) risk-adjusted approaches that incorporate penalties based on the chance of loss.

4.2.2.1 Profit driven approaches

The experiments reported based on profit-driven approaches involve three main separate types of objective, specified in Table 4-1.

#	Objective	Description
1	CMR	Market Return (essentially equivalent to excess return)
2	CPC_LK12	Performance Consistency with 12-unit periods
3	CPC_LK24	Performance Consistency with 24-unit periods

Table 4-1: Three objectives of single-objective approach.

In each case, the initial “C” indicates that fitness is modified by the complexity-penalizing factor as indicated in eq. 3-9. In CPC_LK12, for example, and an experiment involving monthly trading on which the unseen test data cover a 60-month period, fitness (before the complexity modification) is either 0, 1, 2, 3, 4 or 5, according to in how many of the separate 12-month periods the rule was able to outperform both buy and hold and risk-free return.

In the multi-objective approaches tested, each used a combination of either two or three objectives, chosen from the Table 4-1 as well as the additional list of objectives in Table 4-2.

#	Objective	Description
1	MR	Market Return
2	2MR	2 Separated Market Return (Divide the period into 2 sub-periods and use MR as fitness value for each one) – where used, obviously this counts as two objectives
3	PC_LK12	Performance Consistency with 12-unit periods
4	PC_LK24	Performance Consistency with 24-unit periods
5	CXP	Complexity Penalizing Factor – standalone measure of the tree complexity – simply the depth of the tree
6	2CMR	2MR weighted by complexity penalizing factor

Table 4-2: Six objectives of multi-objective approach.

In total, to test profit-driven approaches, we test 9 distinct multi-objective configurations, as specified in Table 4-3.

#	Configuration	No. of Objective	Description
1	MR-CXP	2	MR and CXP
2	PC_LK12-CXP	2	PC with 12-unit periods, and CXP
3	PC_LK24-CXP	2	PC with 24-unit periods, and CXP
4	2MR-CXP	3	MR for two sub-periods, and CXP
5	MR-PC_LK12-CXP	3	MR, PC with 12-unit periods, and CXP
6	MR-PC_LK24-CXP	3	MR, PC with 24-unit periods, and CXP

7	2CMR	2	as 2MR, but both objectives complexity penalized
8	CMR-CPC_LK12	2	MR and PC with 12-unit periods, but both objectives complexity penalized
9	CMR-CPC_LK24	2	MR and PC with 24-unit periods, but both objectives complexity penalized

Table 4-3: Nine multi-objective configurations of profit driven approach.

4.2.2.2 Risk-adjusted approaches

Two basic risk-adjusted fitness functions are involved in our risk-adjusted approaches, as specified in Table 4-4.

#	Objective	Description
1	SHARO	Sharpe Ratio
2	MSTLRO	Modified Stirling Ratio

Table 4-4: Three risk-adjusted objectives of single-objective approach.

In the risk-adjusted multi-objective approaches, each configuration comprised either two, three or four objectives, picked up from Table 4-4 and Table 4-5.

#	Objective	Description
1	MR	Market Return
2	PC_LK12	Performance Consistency with 12-unit periods
3	PC_LK24	Performance Consistency with 24-unit periods
4	MMDD	Modified Drawdown – $\max(\text{drawdown}, \text{threshold\% of current position})$; drawdown are measured as a percentage of traded

		assets
5	CXP	Complexity Penalizing Factor – standalone measure of the tree complexity – simply the depth of the tree

Table 4-5: Five objectives used in risk-adjusted multi-objective approaches.

In total we have 20 distinct multi-objective configurations to test risk-adjusted approaches. These are detailed in Table 4-6.

#	Configuration	No. of Objective	Description
1	SHARO-CXP	2	Sharpe Ratio and CXP
2	MSTLRO-CXP	2	Modified Stirling Ratio and CXP
3	MR-MMDD-CXP	3	MR, Modified Drawdown and CXP
4	SHARO-PC_LK12-CXP	3	Sharpe Ratio, PC with 12-unit periods and CXP
5	SHARO-PC_LK24-CXP	3	Sharpe Ratio, PC with 24-unit periods and CXP
6	MSTLRO-PC_LK12-CXP	3	Modified Stirling Ratio, PC with 12-unit periods and CXP
7	MSTLRO-PC_LK24-CXP	3	Modified Stirling Ratio, PC with 24-unit periods and CXP
8	MR-PC_LK12-SHARO-CXP	4	MR, PC with 12-unit periods, Sharpe Ratio and CXP
9	MR-PC_LK24-SHARO-CXP	4	MR, PC with 24-unit periods, Sharpe Ratio and CXP
10	MR-PC_LK12-MSTLRO-CXP	4	MR, PC with 12-unit periods, Modified Stirling Ratio and CXP

11	MR-PC_LK24-MSTLRO-CXP	4	MR, PC with 24-unit periods, Modified Stirling Ratio and CXP
12	CMR-MMDD	2	MR complexity penalized and Modified Drawdown
13	SHARO-CPC_LK12	2	Sharpe Ratio and PC with 12-unit periods complexity penalized
14	SHARO-CPC_LK24	2	Sharpe Ratio and PC with 24-unit periods complexity penalized
15	MSTLRO-CPC_LK12	2	Modified Stirling Ratio and PC with 12-unit periods complexity penalized
16	MSTLRO-CPC_LK24	2	Modified Stirling Ratio and PC with 24-unit periods complexity penalized
17	CMR-CPC_LK12-SHARO	3	Sharpe Ratio, MR and PC with 12-unit periods, but both objectives complexity penalized
18	CMR-CPC_LK24-SHARO	3	Sharpe Ratio, MR and PC with 24-unit periods, but both objectives complexity penalized
19	CMR-CPC_LK12-MSTLRO	3	Modified Stirling Ratio, MR and PC with 12-unit periods, but both objectives complexity penalized
20	CMR-CPC_LK24-MSTLRO	3	Modified Stirling Ratio, MR and PC with 24-unit periods, but both objectives complexity penalized

Table 4-6: Twenty multi-objective configurations of risk-adjusted approach.

The multi-objective algorithmic approach we use in this study is a classic and straightforward approach, essentially similar in all details to NSGA II as described in [23]. An archive was maintained, with fixed size 10, containing the approximation to the Pareto front so far, and using crowding distance as the arbiter when new individuals were generated that would otherwise overflow the archive. However, the key element that we need to describe is how the returned Pareto front was exploited to make trading decisions on unseen test data. This was done as follows.

As in Chapter 3, experiments were done in the context of three continuous trading periods, called training, validation, and evaluation. In all single-objective experiments reported in this thesis, GP evolves a rule, guided by performance in the training period, but at the same time recording (but not using) the performance of each evaluated rule on the validation period. The rule that performed best on the validation period is then evaluated over the evaluation period, and this is the result returned and summarised in our results section (how this is handled for multi-objective algorithms is discussed next). This regime of training/validation/evaluation (which we call regime 2) was found clearly more robust in previous work [60, 61], and further in Chapter 3, than a straightforward training/evaluation split in which we use the rule that was best on training data.

In the context of multi-objective approaches, all used the same configuration as in single-objective experiments, except that instead of evaluating a single rule, we always use a set of rules, and this set is the current content of the non-dominated archive. We test two approaches: in the ‘*Majority-Voting*’ approach (MJV) the set of rules is used simply to make a majority decision: each individual rule either signals *buy* or *sell* in the current environment. The signal from the set is the majority decision; in the case of a tie, we simply use the decision made by the rule (breaking other ties randomly) that was flagged as achieving the best return (MR) on the validation data. For comparison, we

also test a basic ‘*Normal Trading*’ approach, in which the result of a set of rules is taken to be the average result of the rules treated individually. This corresponds, in the trading context, to giving each rule an equal share of the money to invest – given transaction costs that are strictly in proportion to the amount traded, this arguably yields valid results, though (as we expected and as we will see) is not a particularly effective approach.

4.3 Experiments

We used the GP mutation operators described by Angeline [3] and Banzhaf et al. [4] together as described in section 3.2.4, and standard subtree-swap crossover [54]. No constraint was placed on tree size (other than selection pressure from one or more of the objectives in some configurations), however the population was initialized by growing trees starting at depth of 2 to a maximum depth of 7. The population size was always 500 (with archive maximum size 10 in the multi-objective approaches, though archives tended to be much smaller). As discussed, the multi-objective method was NSGA-II; in a single-objective run, in each generation, the current best was copied into the next generation, and the rest were then produced by crossover of two parents (probability 0.9) or mutation of a single parent (probability 0.1), with parents selected via binary tournament selection. Each run on training data continued for 50 generations.

Parameter	Description
Data Set	S&P 500 index
Algorithm	Steady State GP with NSGA-II
Population Size	500 with archive maximum size 10
Iteration	50 generations

Initialization	Ramp half and half with depth range from 2 – 7
Crossover Rate	0.9
Mutation Rate	0.1
Selection Operator	Crowding binary tournament selection
Mutation Operators	Subtree Mutation, Point Mutation, Permutation, Hoist
Crossover Operator	Subtree-swap crossover

Table 4-7: GP Parameters for multi-objective approach.

We have run our experiments with 8 different data splits for each of monthly, weekly and daily data. These are the same data splits used to run single-objective experiments in the previous chapter, so for more information on each data split please go to section 3.3.2.

4.4 Results

Every combination of algorithm/data-split/regime including, where applicable, MJV or ‘*Normal Trading*’ was repeated 20 times independently, and the results are summarised in this section. We summarise by providing graphs that indicate (a) the relative performance of the multi-objective strategies compared with single-objective strategies, and (b) the relative performance of the multi-objective strategies in comparison with buy and hold. In each case, the display of results is oriented to show the picture across all data splits relevant to the experiment.

4.4.1 Profit Driven Approach Results

Figure 4-1 shows the performance of all multi-objective methods in comparison (implicitly) with single-objective configurations for monthly trading. For example,

2CMR (MJV) achieves 75% on this plot. This means that in 75% (i.e. 15) of 20 independent tests, where each test includes an experiment of both this method (2CMR (MJV)) and all three single-objective approaches on each of the 8 monthly data splits, 2CMR (MJV) outperformed *all* of the single-objective methods. The orange line is used to separate methods between the two types of ways that rules were selected for evaluation in the multi-objective approach: the normal trading group and the majority-voting trading group.

Please note that the bottom 3 methods (MR-CXP, PC_LK12-CXP and PC_LK24-CXP) in Figure 4-1 comprise two objectives, in which the first objective is the same as one of the comparative single-objective approaches, and the second objective is a complexity-penalizing factor. These are to be contrasted with single objective approaches (CMR, CPC_LK12 and CPC_LK24 respectively) in which a single scalar value is used to represent the information in both objectives. We are going to refer to those 3 configurations as the ‘*Basic Group*’, and the remaining 7 methods above the basic group are defined as the ‘*Combination Group*’. Likewise, the 3 methods above the orange line – MR-CXP (MJV), PC_LK12-CXP (MJV) and PC_LK24-CXP (MJV) – are the plots of the basic group with the majority-voting method, and the others above the orange line are the plots of the combination group with the majority-voting method.

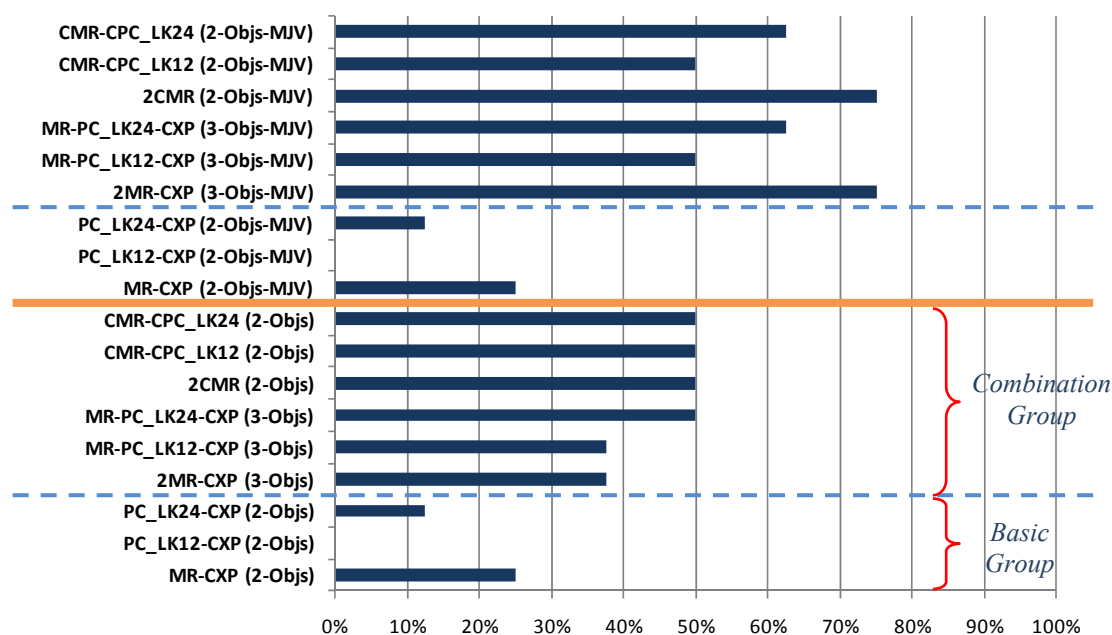


Figure 4-1: Performance of multi-objective methods in comparison with single-objective configurations for monthly trading splits. E.g. 2CMR (MJV) achieves 75%, meaning that: averaged over all 8 monthly data splits, 2CMR (MJV), in 75% of the 20 independent tests, outperformed all of the single-objective methods.

The relative performance of multi-objective vs. single objective in the monthly trading environment does not seem highly convincing from Figure 4-1 at first sight, since there is no majority that extends beyond the 50% mark. However, recall that each single multi-objective approach is compared against *all* single-objective methods for each bar. Also, it is quite clear that the majority-voting (MJV) techniques consistently perform strongly against single-objective methods. In addition, it is notable that all configurations in the basic group are uncompetitive with the single-objective approach (however recall again that each is compared against *all* single-objective methods).

Turning to performance in relation to buy and hold, Figure 4-2 shows, for each multi-objective approach, how often (averaged over the 8 monthly splits) the return

from trading over the evaluation period outperformed buy and hold. This seems to give a fairly convincing result for multi-objective approaches in monthly trading.

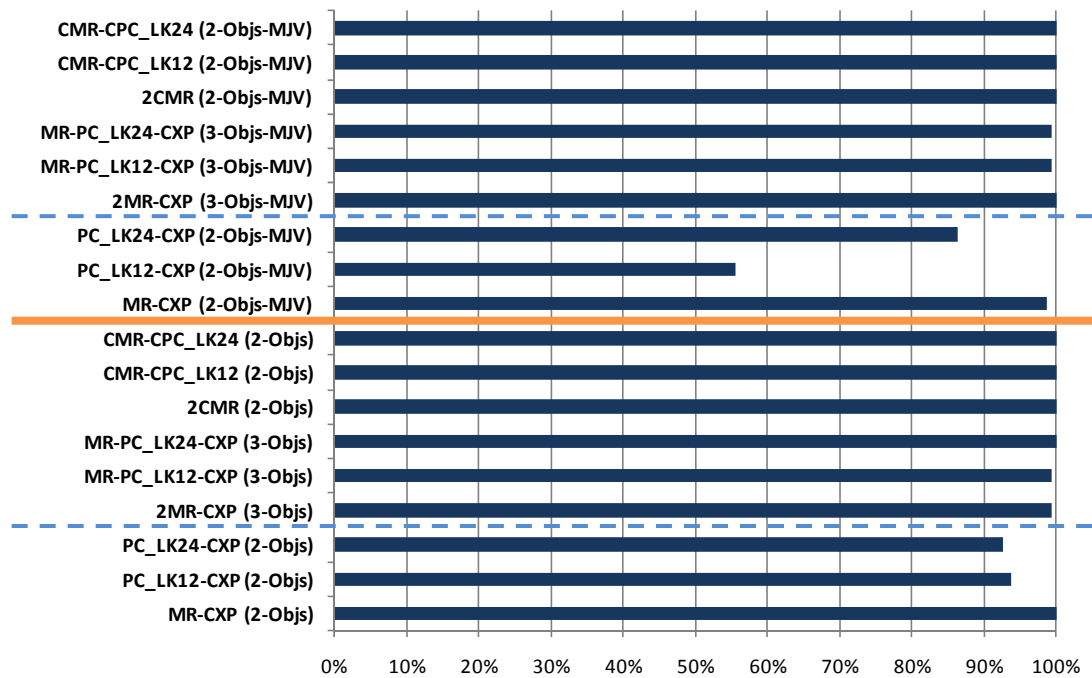


Figure 4-2: Performance of multi-objective methods in comparison with the buy-and-hold strategy for monthly trading splits. E.g. most algorithms achieve 100%, meaning that: averaged over all 8 monthly data splits, the method outperformed buy and hold in all of the 20 independent tests.

In order to provide more statistically detailed comparison between single-objective approaches and multi-objective approaches, paired one-tailed T-Tests are used with a cut-off significance level p -value of 0.10, assuming unequal variance; such a T-Test is done for each of the eight monthly data periods, comparing the ‘best’ single-objective (SO) configuration and the two ‘best’ multi-objective (MO) configurations in each case. The results of these comparisons are found in Table 4-8. The ‘best’ SO method for each data split is chosen by selecting the one with the highest return of all single-objective methods, while the two best MO methods are chosen by selecting the top two methods

from the bar chart in Figure 4-1. From the results in this table, we can claim justifiably that a multi-objective approach performed better than single-objective approaches in five of the monthly data splits: M01 and N_M01-N_M04. In two of the data splits, M02 and M03, the best SO method was able to beat both of the best MO configurations with p-values < 0.10 , while for data split M04 there were statistically no significant differences between the ‘best’ SO method and the two chosen MO configurations. In conclusion, the T-Test results in Table 4-8 indicate better performance from multi-objective approaches in comparison with single objective approaches in 5 out of 8 monthly data splits.

Period	The Best SO Method	Two Best MO Configurations to compare with the Best SO Method using one-tailed paired T-Test		
M01	MR (1-Obj) 1763.834845	config	2MR-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	1783.593935	1784.462333
		p-value	0.00966471	0.012213321
M02	MR (1-Obj) 2214.390803	config	2MR-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	2110.350325	2040.242261
		p-value	0.009522746	0.000682611
M03	MR (1-Obj) 2030.461758	config	2MR-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	1795.148287	1725.723073
		p-value	0.000782516	4.1091E-05
M04	MR (1-Obj) 3936.527613	config	2MR-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	3991.965932	4013.471773
		p-value	0.399129231	0.292392665
N_M01	MR (1-Obj) 1862.514801	config	2MR-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	2047.801221	2080.485376
		p-value	0.003379889	0.000536555
N_M02	MR (1-Obj) 2492.901991	config	2MR-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	2944.873747	2992.682382
		p-value	0.000594646	0.000250337
N_M03	MR (1-Obj) 2348.802023	config	2MR-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	2990.475575	3000.717418
		p-value	0.000236299	0.000302369
N_M04	MR (1-Obj) 3171.526994	config	2MR-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	3911.753109	3992.831974
		p-value	0.000989902	0.000192483

Table 4-8: Paired one-tailed T-Test for the comparison between the best single-objective method and the top two best multi-objective methods for monthly trading of profit driven approach, choosing the top two highest performances, displaying means and p-values for the top two best multi-objective methods and highlighting with gray colour if p-values > 0.10 .

We now try to gain some insight into the relative performance of the multi-objective approaches with each other. The T-Test results in Table 4-9 are done, for each data split in turn, between the two multi-objective approaches that gave the highest mean return in that data split. In monthly split M02, we find that the claim that the majority-voting (MJV) version of 2MR-CXP method performed better than the majority-voting (MJV) version of 2CMR method is supported with a *p*-value of 0.0712; but for the remainder of the monthly data splits, there was no statistically significant difference between the top two MO methods.

Period	The First Best MO Method	Compare with the Second Best MO Method using one-tailed paired T-Test	
M01	2MR-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	1784.462333
		p-value	0.462211591
M02	2MR-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	2040.242261
		p-value	0.071238398
M03	2MR-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	1725.723073
		p-value	0.159149291
M04	2MR-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	4013.471773
		p-value	0.446572389
N_M01	2MR-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	2080.485376
		p-value	0.133749123
N_M02	2MR-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	2992.682382
		p-value	0.222922955
N_M03	2MR-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	3000.717418
		p-value	0.347851031
N_M04	2MR-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	3992.831974
		p-value	0.173446616

Table 4-9: Paired one-tailed T-Test for the comparison of the top two best multi-objective methods for monthly trading of profit driven approach, displaying means and p-values for the second best multi-objective methods and highlighting with gray colour if p-values > 0.10.

Turning now to the more frequent trading interventions of weekly trading, Figure 4-3 and Figure 4-4 show the results (corresponding respectively to Figure 4-1 and Figure 4-2) over the weekly trading splits.

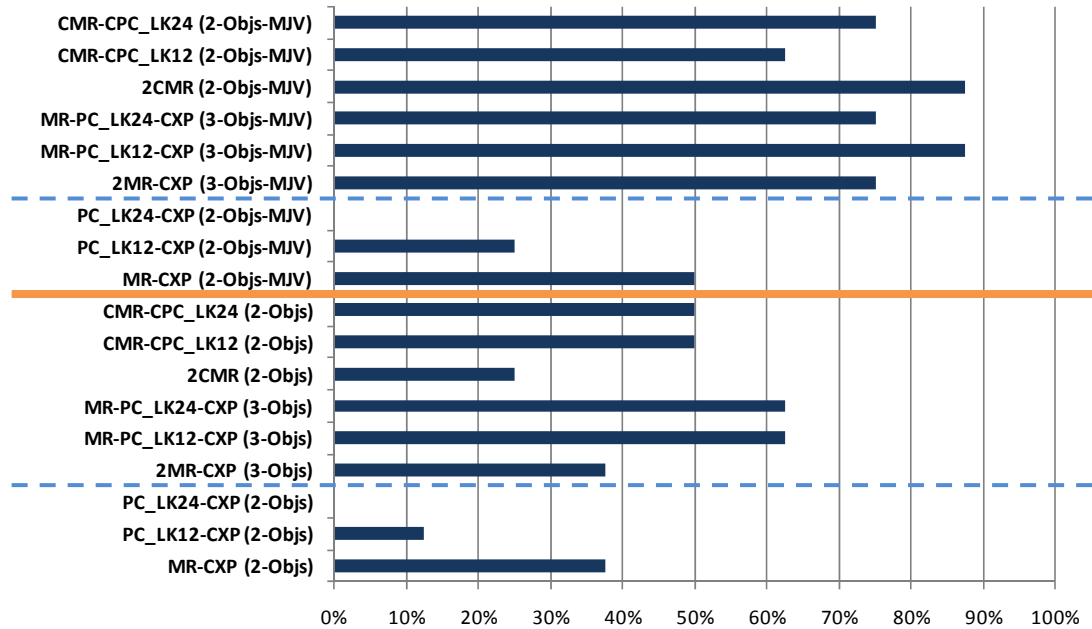


Figure 4-3: Performance of multi-objective methods in comparison with single-objective configurations for weekly trading splits.

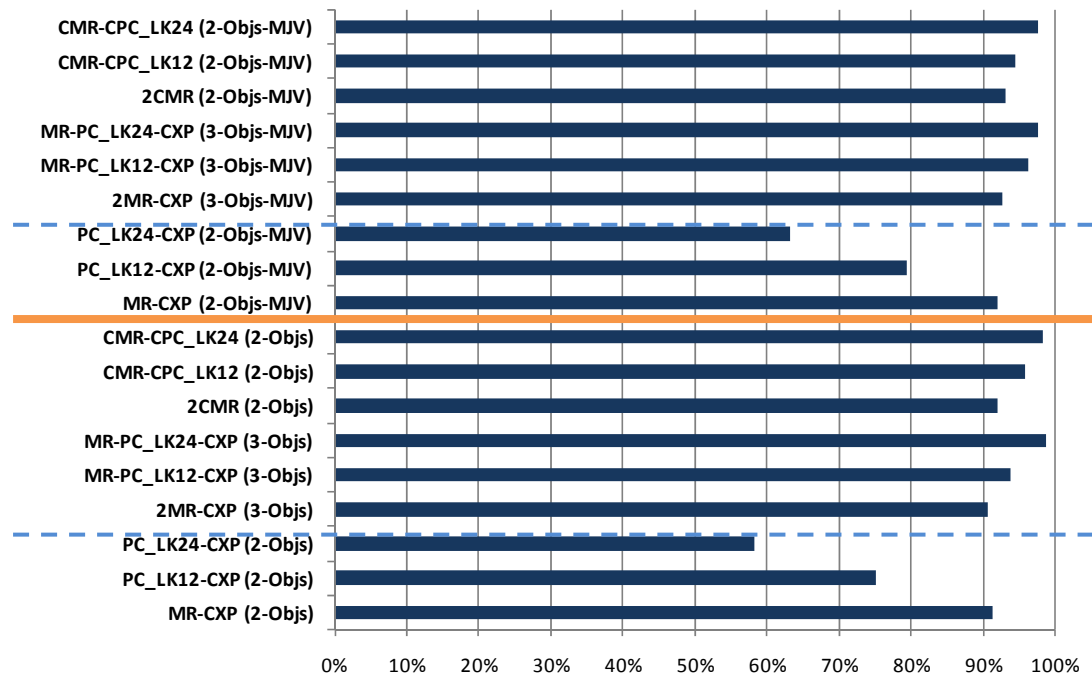


Figure 4-4: Performance of multi-objective methods in comparison with the buy-and-hold strategy for weekly trading splits.

The superiority of multi-objective methods over single objective methods seems to be clearer in the weekly trading environment. It seems clear that multi-objective methods can somehow spread the risk (associated with more volatility) over multiple rules in a way that boosts performance (hitting a peak of 87.5% for weekly trading compared with peaking at 75% in case of monthly trading); this notion is reflected in the observation that both ‘Normal Trading’ and MJV (excluding the basic group) do better in this context than they did in the monthly trading context with the single exception of 2CMR.

Period	The Best SO Method	Two Best MO Configurations to compare with the Best SO Method using one-tailed paired T-Test		
W01	MR (1-Obj) 1440.849036	config	MR-PC_LK12-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	1500.484875	1481.925363
		p-value	0.024283601	0.049667157
W02	MR (1-Obj) 1565.496149	config	MR-PC_LK12-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	1704.66839	1591.497252
		p-value	0.000897204	0.251642391
W03	MR (1-Obj) 1642.733456	config	MR-PC_LK12-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	1643.724201	1682.801142
		p-value	0.479493747	0.019521543
W04	MR (1-Obj) 1254.49184	config	MR-PC_LK12-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	1246.352613	1349.322365
		p-value	0.441329696	0.001024257
N_W01	MR (1-Obj) 935.3488046	config	MR-PC_LK12-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	1001.400532	1080.940048
		p-value	0.158863588	0.001958461
N_W02	MR (1-Obj) 1641.874987	config	MR-PC_LK12-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	1682.594035	1683.119233
		p-value	0.092660824	0.094328728
N_W03	MR (1-Obj) 2005.712205	config	MR-PC_LK12-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	2091.198734	1977.025164
		p-value	0.108154536	0.328853037
N_W04	MR (1-Obj) 1478.938414	config	MR-PC_LK12-CXP (3-Objs-MJV)	2CMR (2-Objs-MJV)
		mean	1554.84372	1547.494597
		p-value	0.105120963	0.163243807

Table 4-10: Paired one-tailed T-Test for the comparison between the best single-objective method and the top two best multi-objective methods for weekly trading of profit driven approach, choosing the top two highest performances, displaying means and p-values for the top two best multi-objective methods and highlighting with gray colour if p-values > 0.10.

The results of paired one-tailed T-Test from Table 4-10 provided supporting evidence that multi-objective methods have superior performance over single objective methods in the case of weekly trading, since the average returns of 2CMR (MJV) were higher than the average returns of all single-objective methods in five periods (W01, W03, W04, N_W01 and N_W02), but no statistically significant difference is found in the three remaining periods (W02, N_W03 and N_W04). In the case of the configuration MR-PC_LK12-CXP (MJV), its average returns were higher than the average return of single-objective methods in three periods (W01, W02 and N_W02) with statistical significance, but with no statistically significant difference in the five periods (W03, W04, N_W01, N_W03 and N_W04). 2CMR (MJV) seems to be favoured over MR-PC_LK12-CXP (MJV) as the number of weekly periods in which it clearly beats SO methods was higher (5 vs. 3), and this finds some confirmation in the results of comparison between these two methods in Table 4-11. In the three weekly splits: W01, N_W02 and N_W04, there was no statistically significant difference between the top two MO methods. However, in W03, W04 and N_W01, the claim that 2CMR (MJV) is better than MR-PC_LK12-CXP (MJV) looks justified, with p-values of 0.0058, 0.0205 and 0.0594 respectively. Meanwhile, in the two weekly periods W02 and N_W03, the T-test justifies a claim that MR-PC_LK12-CXP (MJV) is better than 2CMR (MJV) in these contexts, with p-values of 0.0019 and 0.0036 respectively.

Period	The First Best MO Method	Compare with the Second Best MO Method using one-tailed paired T-Test	
W01	MR-PC_LK12-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	1481.925363
		p-value	0.2853189
W02	MR-PC_LK12-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	1591.497252
		p-value	0.001864569
W03	MR-PC_LK12-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	1682.801142
		p-value	0.005788796
W04	MR-PC_LK12-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	1349.322365
		p-value	0.020546585
N_W01	MR-PC_LK12-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	1080.940048
		p-value	0.059350574
N_W02	MR-PC_LK12-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	1683.119233
		p-value	0.488473137
N_W03	MR-PC_LK12-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	1977.025164
		p-value	0.003627715
N_W04	MR-PC_LK12-CXP (3-Objs-MJV)	config	2CMR (2-Objs-MJV)
		mean	1547.494597
		p-value	0.455385016

Table 4-11: Paired one-tailed T-Test for the comparison of the top two best multi-objective methods for weekly trading of profit driven approach, displaying means and p-values for the second best multi-objective methods and highlighting with gray colour if p-values > 0.10.

Finally, we see the corresponding results for the daily trading environment in Figure 4-5 and Figure 4-6. These figures clearly tell a different story. Figure 4-6 reveals that the multi-objective methods that we have tested can outperform buy and hold at best roughly half the time in the daily trading environment. However it should still be recognised that this in itself is not particularly poor performance – all of the ‘buy-and-hold’ results compared against in this chapter have the advantage, perhaps unfair in reality, of being able to choose between either the buy-and-hold return or the risk-free return, whichever would have generated most over the trading period. What is certainly clear, however, from Figure 4-5 is that the multi-objective methods we have studied are outperformed by the single objective methods in the daily trading context.

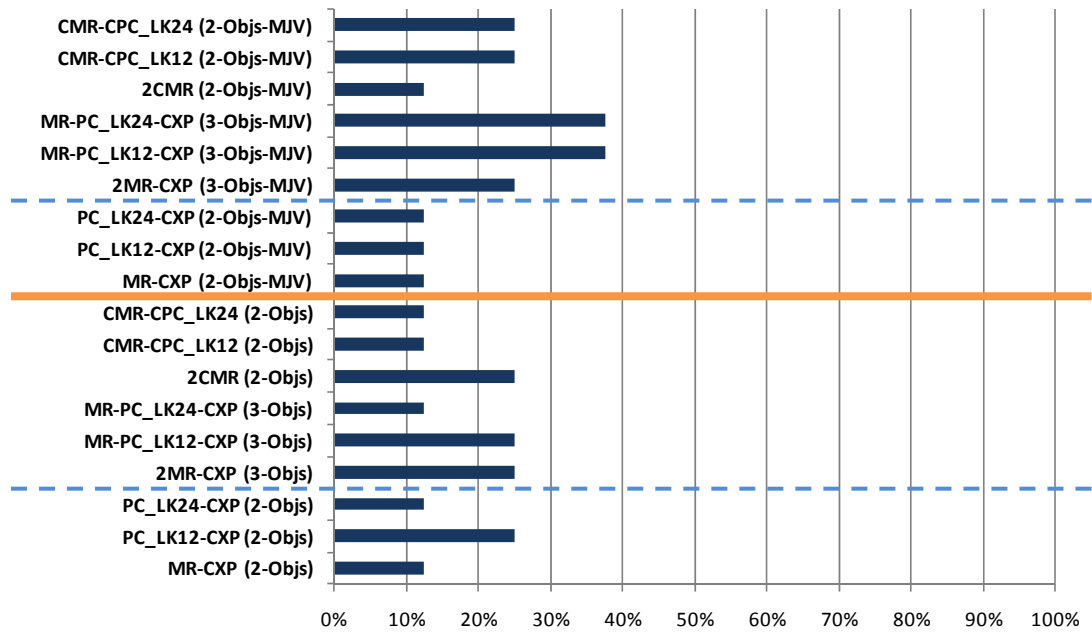


Figure 4-5: Performance of multi-objective methods in comparison with single-objective configurations for daily trading splits.

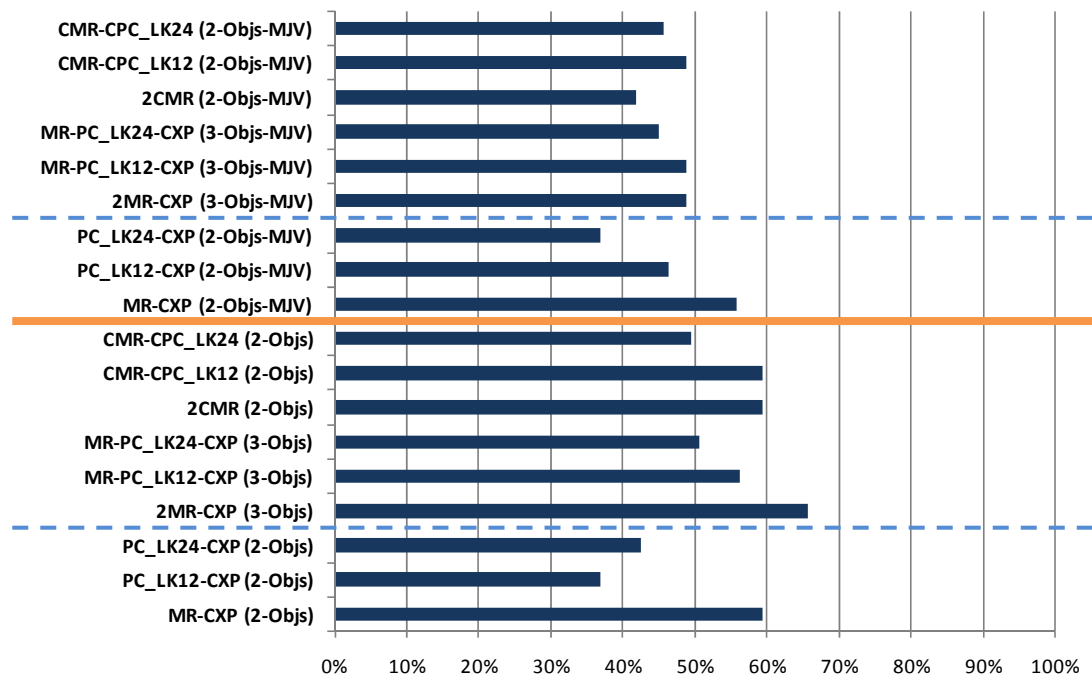


Figure 4-6: Performance of multi-objective methods in comparison with the buy-and-hold strategy for daily trading splits.

Considering paired one-tailed T-Test for daily trading in Table 4-12, we find varied results. To begin with, there are three daily periods (D01, D02 and D04) in which there is no statistically significant difference between the MO method with highest return

and each of the best SO methods of these periods. Meanwhile, the “N/A” in period D01 indicates that T-tests could not be done, since the results of the best SO method (PC_LK24) in this period, the top two MO methods (MR-PC_LK12-CXP (MJV) and MR-PC_LK24-CXP (MJV)) are identical, so there is clearly no statistical difference between them. In the three periods of D03, N_D02 and N_D04, a multi-objective approach beat the best SO method with statistical significance, while this situation was reversed in splits N_D01 and N_D03. To sum up, it’s quite difficult to say which of single-objective configurations or multi-objective configurations have the most potential in the daily trading environment.

Period	The Best SO Method	Two Best MO Configurations to compare with the Best SO Method using one-tailed paired T-Test		
D01	PC_LK24 (1-Obj)	config	MR-PC_LK12-CXP (3-Objs-MJV)	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	752.8806005	752.8806005
		p-value	N/A	N/A
D02	PC_LK12 (1-Obj)	config	MR-PC_LK12-CXP (3-Objs-MJV)	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	934.3635294	924.3738095
		p-value	0.182478451	0.016198131
D03	PC_LK24 (1-Obj)	config	MR-PC_LK12-CXP (3-Objs-MJV)	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	992.3213583	1000.954876
		p-value	0.481207703	0.014951126
D04	PC_LK24 (1-Obj)	config	MR-PC_LK12-CXP (3-Objs-MJV)	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	870.1194886	873.6949207
		p-value	0.229978179	0.387953361
N_D01	MR (1-Obj)	config	MR-PC_LK12-CXP (3-Objs-MJV)	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	1080.37146	1078.337019
		p-value	7.72508E-05	9.37335E-06
N_D02	MR (1-Obj)	config	MR-PC_LK12-CXP (3-Objs-MJV)	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	1030.146855	1021.835405
		p-value	0.001555362	0.03381536
N_D03	PC_LK24 (1-Obj)	config	MR-PC_LK12-CXP (3-Objs-MJV)	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	945.2472045	962.7447119
		p-value	0.000661701	0.179103212
N_D04	MR (1-Obj)	config	MR-PC_LK12-CXP (3-Objs-MJV)	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	1014.703968	1005.881851
		p-value	0.058536588	0.486526644

Table 4-12: Paired one-tailed T-Test for the comparison between the best single-objective method and the top two best multi-objective methods for daily trading of profit driven approach, choosing the top two highest performances, displaying means and p-values for the top two best multi-objective methods and highlighting with gray colour if p-values > 0.10.

The results of selected one-tailed paired T-Tests for daily trading in Table 4-13 revealed that in each of the four daily periods, D02, D03, N_D03 and N_D04, the claim that the best MO method performed better than the second-best MO method (choosing MO methods according to their performance in the context of Figure 4-5) was reasonable (p -values < 0.1), while in the remaining four daily periods, there was statistically no significant difference between them.

Period	The First Best MO Method	Compare with the Second Best MO Method using one-tailed paired T-Test	
D01	MR-PC_LK12-CXP (3-Objs-MJV)	config	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	752.8806005
	752.8806005	p-value	N/A
D02	MR-PC_LK12-CXP (3-Objs-MJV)	config	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	924.3738095
	934.3635294	p-value	0.001748318
D03	MR-PC_LK12-CXP (3-Objs-MJV)	config	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	1000.954876
	992.3213583	p-value	0.072819032
D04	MR-PC_LK12-CXP (3-Objs-MJV)	config	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	873.6949207
	870.1194886	p-value	0.290213616
N_D01	MR-PC_LK12-CXP (3-Objs-MJV)	config	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	1078.337019
	1080.37146	p-value	0.262586141
N_D02	MR-PC_LK12-CXP (3-Objs-MJV)	config	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	1021.835405
	1030.146855	p-value	0.183587211
N_D03	MR-PC_LK12-CXP (3-Objs-MJV)	config	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	962.7447119
	945.2472045	p-value	0.027050523
N_D04	MR-PC_LK12-CXP (3-Objs-MJV)	config	MR-PC_LK24-CXP (3-Objs-MJV)
		mean	1005.881851
	1014.703968	p-value	0.061818838

Table 4-13: Paired one-tailed T-Test for the comparison of the top two best multi-objective methods for daily trading of profit driven approach, displaying means and p-values for the second best multi-objective methods and highlighting with gray colour if p-values > 0.10 .

When we consider these results in terms of the relative performance of the nine different ways of characterising performance with multiple objectives, in both the combination and basic groups, there seem to be two fairly clear observations that emerge. These are best revealed by appeal to Figure 4-2, Figure 4-4 and Figure 4-6, in

which performance is compared directly with buy and hold, rather than confounded by the variations in performance of the single-objective methods. First, as already noted, the ‘*Majority-Voting*’ approach (MJV) tends to lead to better performance for monthly and weekly trading, although in daily trading the ‘*Normal Trading*’ approach, simply spreading the investment decision over each rule, seems clearly to have an advantage, and this seems independent of other aspects of the multi-objective configuration. We expect to achieve a better understanding of this in future work, which will look at variations on the majority voting approach, which give individual rules different weights according to measures of the risk and return associated with them. However, turning to comparison of the nine objective combinations, one other clear observation is that the least effective approach, in monthly and weekly trading, was always a method within the basic group. This is also true for both monthly and weekly trading when we consider the configurations in the majority-voting trading group (the top 9 methods in all plots). Considering methods in the basic group, it is notable that MR-CXP is the only one of the multi-objective configurations that did not include an objective that related to risk. When PC_LK12 or PC_LK24 (or their complexity modified variants) were involved, the corresponding objectives essentially gave a measure of the variance in performance of the rule over the trading period, which is precisely what is attempted in a risk-based metric such as the Sharpe ratio. Those three methods in the basic group themselves could not perform well compared with single-objective approaches, however when combining them together with a complexity-penalizing factor to form two-objective or three-objective methods, the new methods can bring about better performance, as reflected in Figure 4-1 and Figure 4-3. Similarly, when 2MR or 2CMR were involved, these broke down the (perhaps modified) excess return measure into two separate periods, again revealing information about variation in performance and hence providing a simple estimate of the rule’s (or ruleset’s) risk.

It therefore seems clear that risk-oriented measures are valuable in this context for monthly or weekly trading, and this value is exploited well by the multi-objective approaches that incorporated such measures. The reverse seems to be the case, however, in the more volatile context of daily trading. We speculate that this may be because large variations in the returns of a particular rule (or ruleset) are far more frequent in the daily context; so, metrics that estimate that variation, especially over smaller timescales, are likely to be more affected by noise. However it is worth pointing out that our experimental setup may mitigate against finding robust rules for daily trading, simply because the technical indicator primitives we use are forced, in that case, to consider only relatively brief timescales (see section 3.2.2). For example, rules can refer at most to a 10-day moving average.

4.4.2 Risk-adjusted Approach Results

The single-objective approach, being used to compare with all multi-objective configurations in this section, consist of five methods: the same three methods as in the profit driven approach, and two more methods: the *Sharpe Ratio (SHARO)* and *Modified Sterling Ratio (MSTLRO)*.

Please note that the bottom two methods (SHARO-CXP and MSTLRO-CXP) in Figure 4-7 each comprise two objectives, the first of which is the same as in the two single-objective risk-adjusted approaches, and the second objectives are complexity-penalizing factors. We are going to refer to those 2 configurations as ‘*Risk Basic Group*’, and the remaining 18 methods above the basic group are defined as the ‘*Risk Combination Group*’. Likewise, the two methods above the orange line – SHARO-CXP (MJV) and MSTLRO-CXP (MJV) – are the plots of risk basic group with majority-voting trading decision, and the others above the orange line are the plots of the risk combination group with the same decision technique.

In this section, we therefore have many multi-objective configurations incorporating risk factors, which will be tested and compared against the results of single-objective configurations, and the profit-driven multi-objective configurations in the previous section.

First, we consider Figure 4-7, which summarises the performance of the risk-based multi-objective configurations in comparison with single-objective approaches, in the context of monthly trading. If we consider only the ‘Normal Trading’ variants, we see that the relative performance of multi-objective vs. single configurations go in the same direction as we saw with the profit-driven multi-objective approaches, in which few configurations can outperform single-objective methods more than 50% of the time. The majority-voting (MJV) configurations show similar performance.

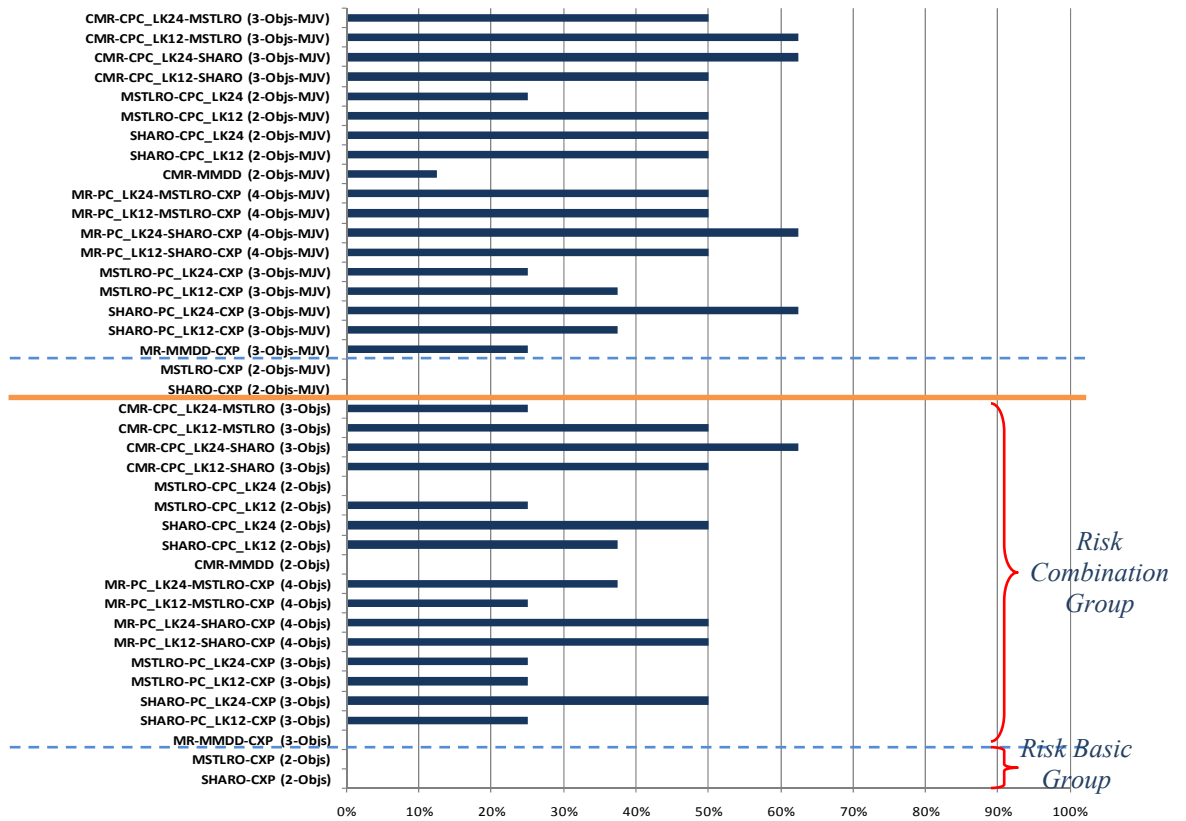


Figure 4-7: Performance of multi-objective methods (incorporating risk) in comparison with single-objective configurations for monthly trading splits.

In the same way, it can be seen from Figure 4-8 below that the multi-objective approaches outperform buy-and-hold reliably, over 90% of the time for all configurations, again similar to the performance of the profit driven approach.

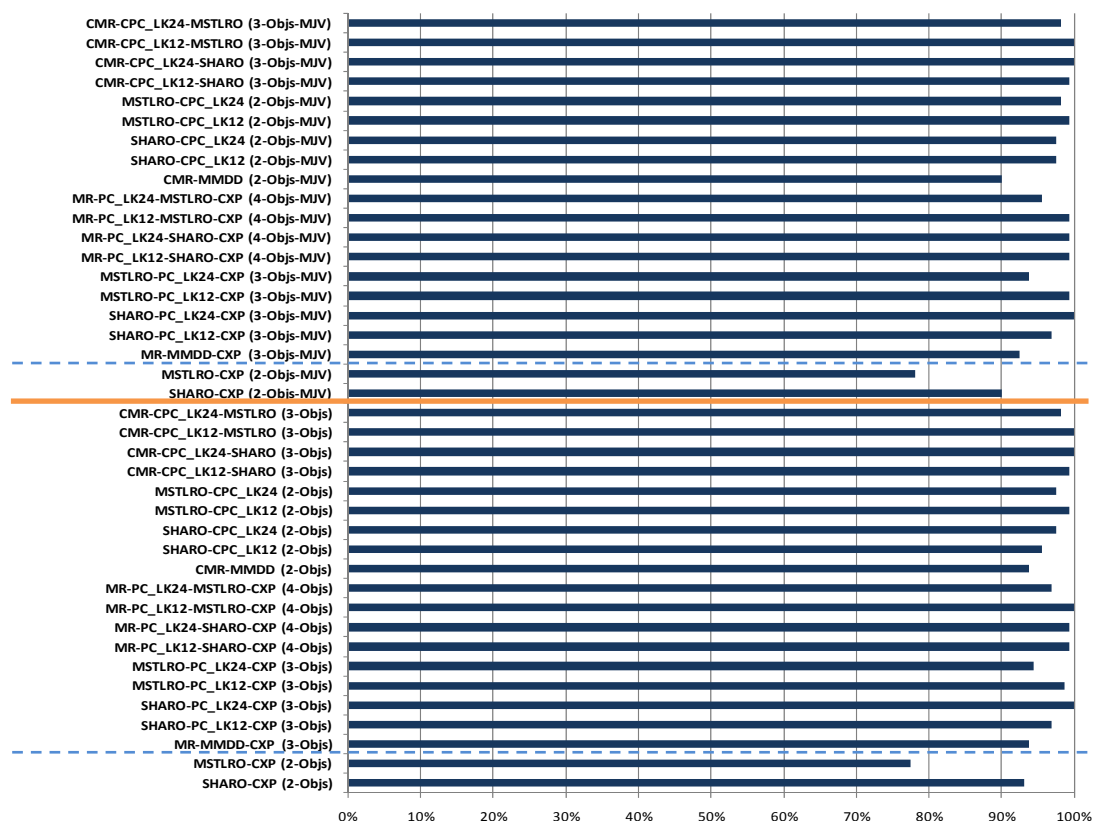


Figure 4-8: Performance of multi-objective methods (incorporating risk) in comparison with the buy-and-hold strategy for monthly trading splits.

Turning now to comparison statistics, Table 4-14 provide the results of one-tailed paired T-Tests (assuming unequal variance) for each of the eight monthly data splits, comparing the ‘best’ single-objective method with the ‘best’ multi-objective methods, using the cut-off p -value of 0.10. Recall that the best SO method for each data period is selected by choosing the one with the highest average return among all single-objective configurations; in contrast, the two best MO methods are picked by looking up the top two highest performance methods from the bar chart in Figure 4-7, and if there are more than two methods with equal best performance, then the selections are just decided arbitrarily. The selected MO methods in this case are CMR-CPC_LK12-SHARO and CMR-CPC_LK12-MSTLRO (MJV).

Period	The Best SO Method	Two Best MO Configurations to compare with the Best SO Method using one-tailed paired T-Test		
M01	MR (1-Obj) 1763.834845	config	CMR-CPC_LK12-SHARO (3-Objs)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1746.735319	1762.290111
		p-value	0.116460297	0.426383841
M02	MR (1-Obj) 2214.390803	config	CMR-CPC_LK12-SHARO (3-Objs)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1868.424373	1969.898131
		p-value	3.24165E-08	8.28095E-05
M03	MR (1-Obj) 2030.461759	config	CMR-CPC_LK12-SHARO (3-Objs)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1611.124535	1865.316405
		p-value	1.30301E-09	0.009339986
M04	MR (1-Obj) 3936.527614	config	CMR-CPC_LK12-MSTLRO (3-Objs)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	3753.032734	4097.57303
		p-value	0.14806396	0.172953235
N_M01	MR (1-Obj) 1862.514801	config	CMR-CPC_LK12-SHARO (3-Objs)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1898.212886	1949.266556
		p-value	0.273430398	0.106959671
N_M02	MR (1-Obj) 2492.901991	config	CMR-CPC_LK12-SHARO (3-Objs)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	2726.671321	2726.76254
		p-value	0.043480252	0.029670059
N_M03	MR (1-Obj) 2348.802023	config	CMR-CPC_LK12-SHARO (3-Objs)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	2704.211256	2817.719059
		p-value	0.016129797	0.005836562
N_M04	MR (1-Obj) 3171.526994	config	CMR-CPC_LK12-SHARO (3-Objs)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	3172.293249	3633.185833
		p-value	0.498621035	0.013805083

Table 4-14: Paired one-tailed T-Test for the comparison between the best single-objective method and the top two multi-objective methods for monthly trading of risk-adjusted approach, choosing the top two highest performances, displaying means and p-values for the top two best multi-objective methods and highlighting with gray colour if p-values > 0.10.

The results in Table 4-14 show that there were no statistical difference between the best SO method and two best MO methods in M01, M04 and N_M01. However in two of the remaining data splits (M02 and M03) the best SO approach was significantly better than the MO approaches, while in N_M02, N_M03 and N_M04 did the MO approaches outperform the best SO approach with statistical significance. When we consider the comparisons between the MO methods only, Table 4-15 indicates that CMR-CPC_LK12-MSTLRO was more effective method than CMR-CPC_LK12-SHARO in 5 out of the 8 monthly data splits, with no statistically significant difference found in the remaining 3 splits. To sum up, the results from Table 4-14 do not provide

a clear message about either SO or MO approaches in the context of monthly trading using risk-based approaches.

Period	The First Best MO Method	Compare with the Second Best MO Method using one-tailed paired T-Test	
M01	CMR-CPC_LK12-SHARO (3-Objs) 1746.735319	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1762.290111
		p-value	0.082722129
M02	CMR-CPC_LK12-SHARO (3-Objs) 1868.424373	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1969.898131
		p-value	0.041565911
M03	CMR-CPC_LK12-SHARO (3-Objs) 1611.124535	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1865.316405
		p-value	0.000191505
M04	CMR-CPC_LK12-SHARO (3-Objs) 3753.032734	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	4097.57303
		p-value	0.002160792
N_M01	CMR-CPC_LK12-SHARO (3-Objs) 1898.212886	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1949.266556
		p-value	0.132747619
N_M02	CMR-CPC_LK12-SHARO (3-Objs) 2726.671321	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	2726.76254
		p-value	0.49956642
N_M03	CMR-CPC_LK12-SHARO (3-Objs) 2704.211256	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	2817.719059
		p-value	0.12270325
N_M04	CMR-CPC_LK12-SHARO (3-Objs) 3172.293249	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	3633.185833
		p-value	0.001489808

Table 4-15: Paired one-tailed T-Test for the comparison of the top two best multi-objective methods for monthly trading of risk-adjusted approach, displaying means and p-values for the second best multi-objective methods and highlighting with gray colour if p-values > 0.10.

Moving on to the weekly trading environment, the performance of multi-objective methods (specifically in the MJV group) evidently outranks single objective methods with the best performances of 4 configurations peaking at 75% as indicated in Figure 4-9, and all configurations are more effective than buy and hold in the range of 70 to 90%, with the sole exception of MSTLRO-CXP which makes up roughly 50%. It is clear that when integrating risk assessment into the objectives, the multi-objective methods can compete favourably against single-objective approaches. However, compared to the results for weekly trading using profit-driven approaches, the risk-

adjusted approaches show a somewhat lower performance advantage over single-objective approaches (best performance at 87.5% for profit driven approach as opposed to 75% for risk-adjusted approach).

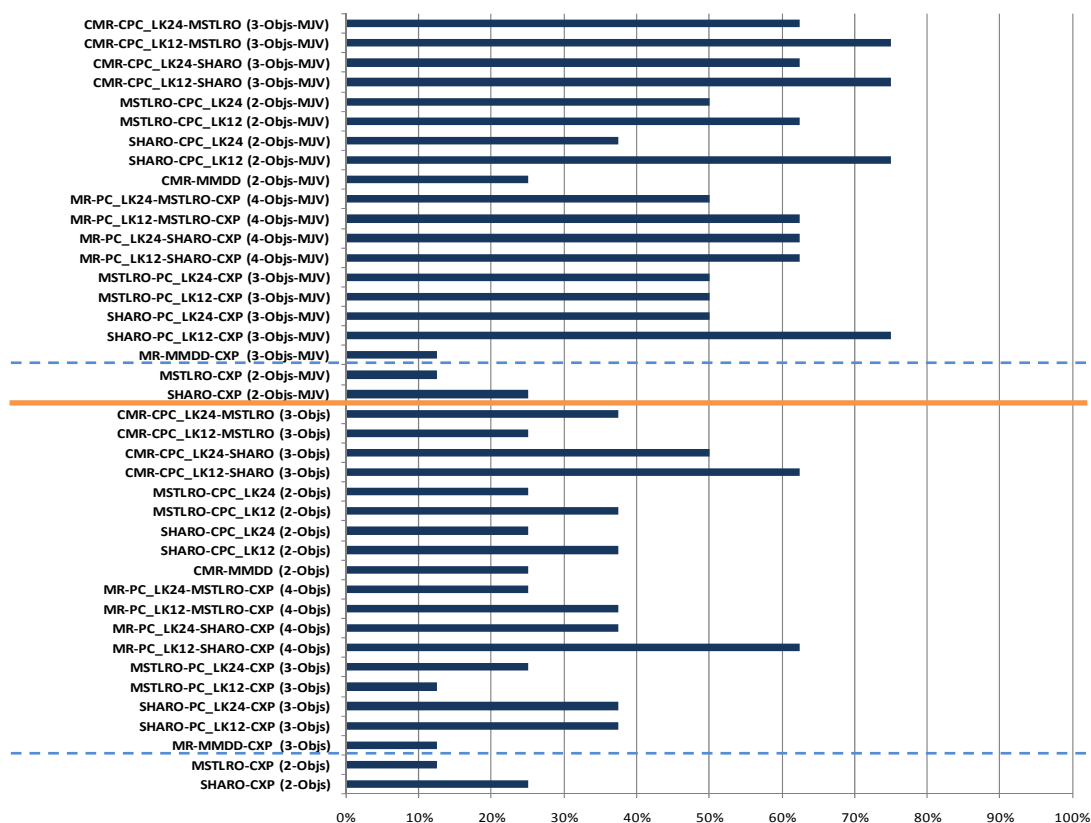


Figure 4-9: Performance of multi-objective methods (incorporating risk) in comparison with single-objective configurations for weekly trading splits.

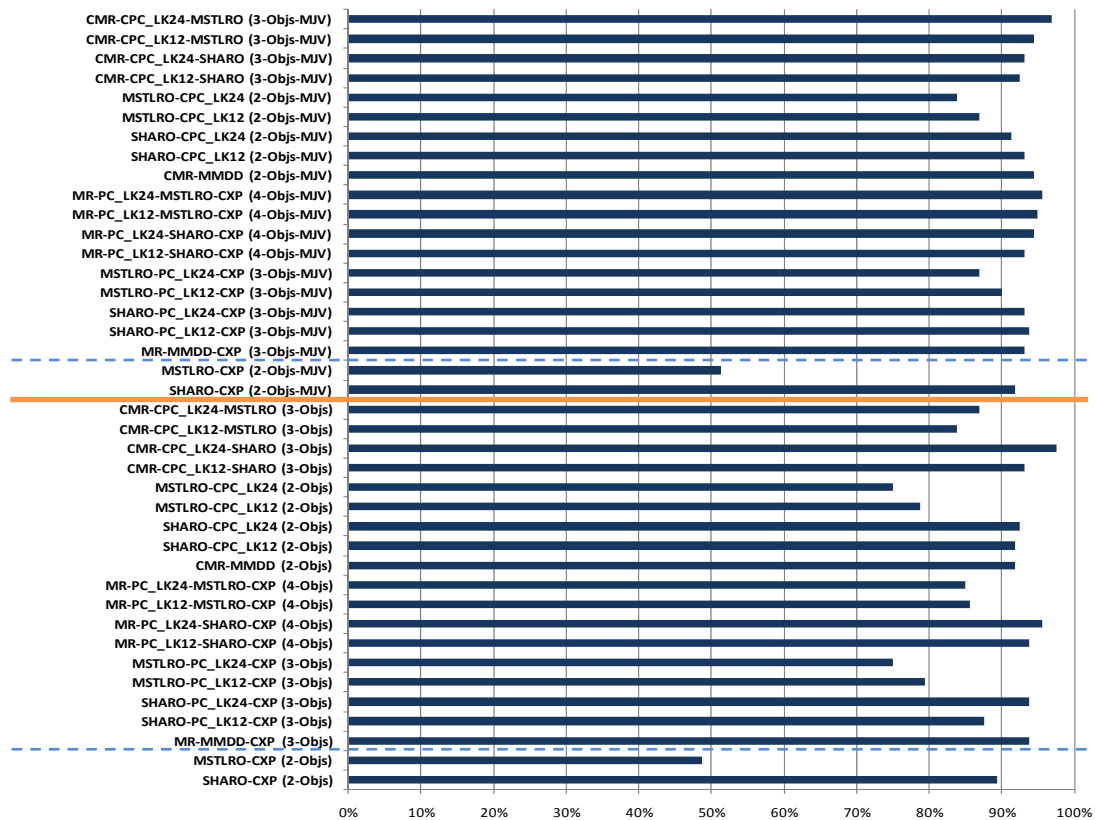


Figure 4-10: Performance of multi-objective methods (incorporating risk) in comparison with the buy-and-hold strategy for weekly trading splits.

The T-Test results in Table 4-16 back up the conclusions that can be made from the bar charts of Figure 4-9, namely that multi-objective methods have superior performance over single objective methods in the weekly trading environment; the T-Tests comparing CMR-CPC_LK12-MSTLRO with the best SO methods are significant in 6 out of the 8 weekly splits.

Period	The Best SO Method	Two Best MO Configurations to compare with the Best SO Method using one-tailed paired T-Test		
W01	SHARO (1-Obj)	config	SHARO-PC_LK12-CXP (3-Objs-MJV)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1526.430557	1493.674586
		p-value	0.000243113	4.88E-02
W02	MR (1-Obj)	config	SHARO-PC_LK12-CXP (3-Objs-MJV)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1634.087095	1589.735882
		p-value	0.01916127	9.20E-02
W03	MR (1-Obj)	config	SHARO-PC_LK12-CXP (3-Objs-MJV)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1622.192314	1629.25407
		p-value	0.088002551	2.69E-01
W04	SHARO (1-Obj)	config	SHARO-PC_LK12-CXP (3-Objs-MJV)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1320.598383	1344.309764
		p-value	0.007889285	4.42E-03
N_W01	SHARO (1-Obj)	config	SHARO-PC_LK12-CXP (3-Objs-MJV)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1109.015479	1192.063086
		p-value	0.284289014	2.84E-02
N_W02	MR (1-Obj)	config	SHARO-PC_LK12-CXP (3-Objs-MJV)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1678.81507	1647.309495
		p-value	0.238141524	4.00E-01
N_W03	SHARO (1-Obj)	config	SHARO-PC_LK12-CXP (3-Objs-MJV)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	2060.737877	2047.245114
		p-value	0.333778783	4.29E-01
N_W04	MR (1-Obj)	config	SHARO-PC_LK12-CXP (3-Objs-MJV)	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1376.221481	1392.270999
		p-value	0.006325206	3.61E-02

Table 4-16: Paired one-tailed T-Test for the comparison between the best single-objective method and the top two best multi-objective methods for weekly trading of risk-adjusted approach, choosing the top two highest performances, displaying means and p-values for the top two best multi-objective methods and highlighting with gray colour if p-values > 0.10.

To find out if there is any statistical difference between the two apparent best MO methods, SHARO-PC_LK12-CXP (MJV) and CMR-CPC_LK12-MSTLRO (MJV), Table 4-17 was prepared, showing no statistically significant difference between these two methods in 6 of the 8 weekly splits.

Period	The First Best MO Method	Compare with the Second Best MO Method using one-tailed paired T-Test	
W01	SHARO-PC_LK12-CXP (3-Objs-MJV) 1526.430557	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1493.674586
		p-value	0.129978926
W02	SHARO-PC_LK12-CXP (3-Objs-MJV) 1634.087095	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1589.735882
		p-value	0.046135441
W03	SHARO-PC_LK12-CXP (3-Objs-MJV) 1622.192314	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1629.25407
		p-value	0.325754354
W04	SHARO-PC_LK12-CXP (3-Objs-MJV) 1320.598383	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1344.309764
		p-value	0.204960084
N_W01	SHARO-PC_LK12-CXP (3-Objs-MJV) 1109.015479	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1192.063086
		p-value	0.016478652
N_W02	SHARO-PC_LK12-CXP (3-Objs-MJV) 1678.81507	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1647.309495
		p-value	0.227919207
N_W03	SHARO-PC_LK12-CXP (3-Objs-MJV) 2060.737877	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	2047.245114
		p-value	0.387661539
N_W04	SHARO-PC_LK12-CXP (3-Objs-MJV) 1376.221481	config	CMR-CPC_LK12-MSTLRO (3-Objs-MJV)
		mean	1392.270999
		p-value	0.355657378

Table 4-17: Paired one-tailed T-Test for the comparison of the top two best multi-objective methods for weekly trading of risk-adjusted approach, displaying means and p-values for the second best multi-objective methods and highlighting with gray colour if p-values > 0.10.

The similar performance pattern of the profit driven approach for daily trading data repeats again here; both Figure 4-11 and Figure 4-12 reveal that multi-objective methods cannot get the better of single-objective methods, where no configuration in Figure 4-11 can reach a performance of 30% with single exception of MSTLRO-CXP, and all configurations excluding MSTLRO-CXP method only outperform buy-and-hold trading approximately just half of the time.

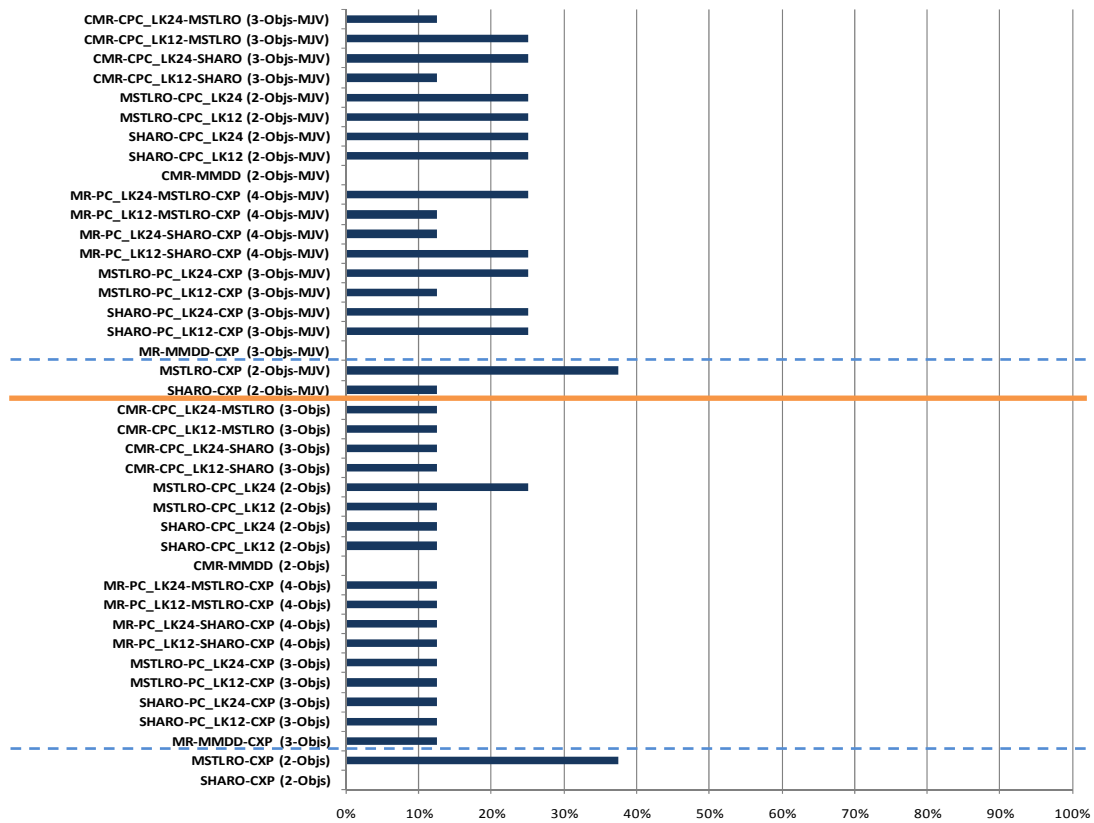


Figure 4-11: Performance of multi-objective methods (incorporating risk) in comparison with single-objective configurations for daily trading splits.

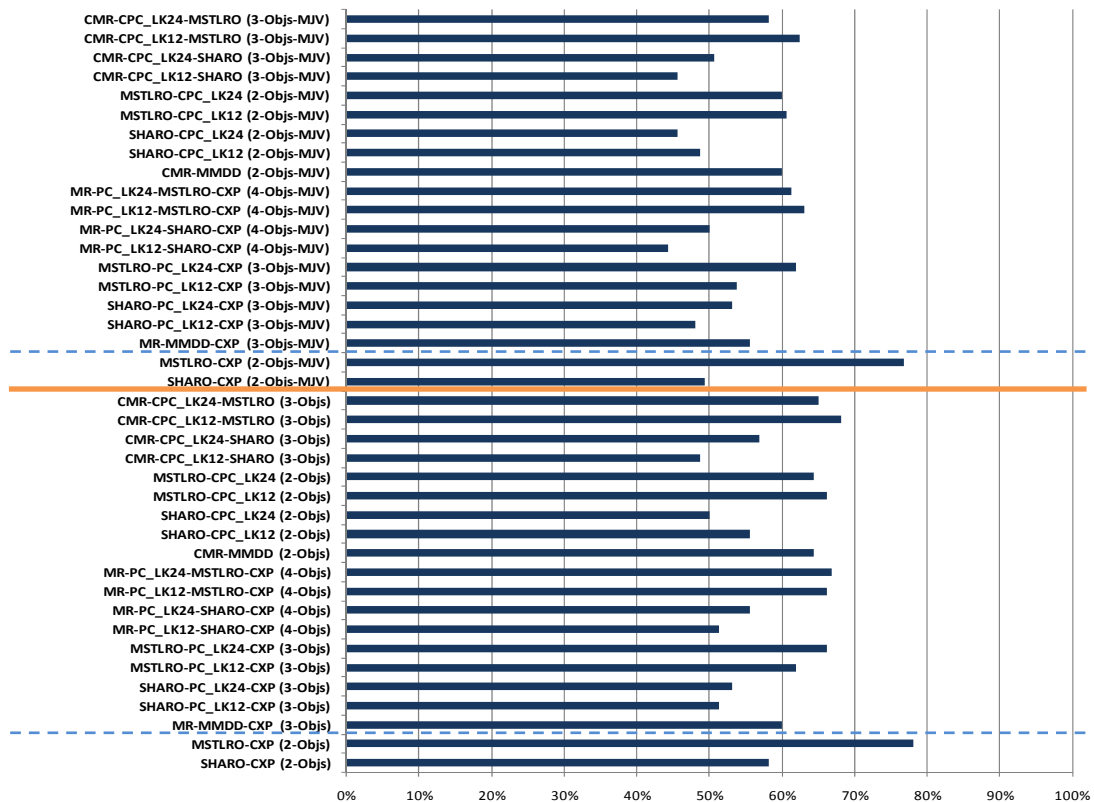


Figure 4-12: Performance of multi-objective methods (incorporating risk) in comparison with the buy-and-hold strategy for daily trading splits.

The best-performing MO method in the daily context seems clearly to be MSTLRO-CXP (MJV), and the second-best seems to be the ‘normal trading’ version of the same configuration. As we can see from Table 4-19, the superiority of MSTLRO-CXP (MJV) over its normal trading counterpart is statistically significant in four of the data splits. However, Table 4-18 makes it clear that the best SO method in each period was usually significantly better than the best MO method.

Period	The Best SO Method	Two Best MO Configurations to compare with the Best SO Method using one-tailed paired T-Test		
D01	MSTLRO (1-Obj) 988.4463669	config	MSTLRO-CXP (2-Objs)	MSTLRO-CXP (2-Objs-MJV)
		mean	986.8262942	990.4949016
		p-value	0.286674079	4.72E-02
D02	MSTLRO (1-Obj) 998.1477372	config	MSTLRO-CXP (2-Objs)	MSTLRO-CXP (2-Objs-MJV)
		mean	1000.708644	1003.128107
		p-value	0.219760602	8.55E-02
D03	SHARO (1-Obj) 994.6271843	config	MSTLRO-CXP (2-Objs)	MSTLRO-CXP (2-Objs-MJV)
		mean	995.5443138	993.9195622
		p-value	0.322937439	4.12E-01
D04	MSTLRO (1-Obj) 998.1005266	config	MSTLRO-CXP (2-Objs)	MSTLRO-CXP (2-Objs-MJV)
		mean	955.4950048	959.3966463
		p-value	0.000629181	7.76E-03
N_D01	MR (1-Obj) 1092.446401	config	MSTLRO-CXP (2-Objs)	MSTLRO-CXP (2-Objs-MJV)
		mean	1029.820099	1025.543327
		p-value	4.4951E-07	1.95E-06
N_D02	MSTLRO (1-Obj) 1010.203433	config	MSTLRO-CXP (2-Objs)	MSTLRO-CXP (2-Objs-MJV)
		mean	1008.658873	1006.694397
		p-value	0.178686113	1.74E-02
N_D03	MSTLRO (1-Obj) 977.3706252	config	MSTLRO-CXP (2-Objs)	MSTLRO-CXP (2-Objs-MJV)
		mean	994.6999819	997.9471185
		p-value	1.88473E-07	7.51E-08
N_D04	SHARO (1-Obj) 1006.428205	config	MSTLRO-CXP (2-Objs)	MSTLRO-CXP (2-Objs-MJV)
		mean	985.3861237	999.6338978
		p-value	0.007240543	2.20E-01

Table 4-18: Paired one-tailed T-Test for the comparison between the best single-objective method and the top two best multi-objective methods for daily trading of risk-adjusted approach, choosing the top two highest performances,

displaying means and p-values for the top two best multi-objective methods and highlighting with gray colour if p-values > 0.10 .

Period	The First Best MO Method	Compare with the Second Best MO Method using one-tailed paired T-Test	
D01	MSTLRO-CXP (2-Objs) 986.8262942	config	MSTLRO-CXP (2-Objs-MJV)
		mean	990.4949016
		p-value	0.077043965
D02	MSTLRO-CXP (2-Objs) 1000.708644	config	MSTLRO-CXP (2-Objs-MJV)
		mean	1003.128107
		p-value	0.024792397
D03	MSTLRO-CXP (2-Objs) 995.5443138	config	MSTLRO-CXP (2-Objs-MJV)
		mean	993.9195622
		p-value	0.13736473
D04	MSTLRO-CXP (2-Objs) 955.4950048	config	MSTLRO-CXP (2-Objs-MJV)
		mean	959.3966463
		p-value	0.309800201
N_D01	MSTLRO-CXP (2-Objs) 1029.820099	config	MSTLRO-CXP (2-Objs-MJV)
		mean	1025.543327
		p-value	0.127238866
N_D02	MSTLRO-CXP (2-Objs) 1008.658873	config	MSTLRO-CXP (2-Objs-MJV)
		mean	1006.694397
		p-value	0.009595877
N_D03	MSTLRO-CXP (2-Objs) 994.6999819	config	MSTLRO-CXP (2-Objs-MJV)
		mean	997.9471185
		p-value	4.48055E-05
N_D04	MSTLRO-CXP (2-Objs) 985.3861237	config	MSTLRO-CXP (2-Objs-MJV)
		mean	999.6338978
		p-value	0.008683807

Table 4-19: Paired one-tailed T-Test for the comparison of the top two best multi-objective methods for daily trading of risk-adjusted approach, displaying means and p-values for the second best multi-objective methods and highlighting with gray colour if p-values > 0.10 .

It is noticeable from Figure 4-11 and Figure 4-12 that the two-objective method of MSTLRO-CXP considerably outperforms the other MO methods, so it is worth taking a closer look at this method. The results from Table 4-20 point out that this method can beat buy and hold completely (at rate of 100% - which means that in 20 independent runs it can beat buy and hold every time) in 6 out of the 8 daily data splits. Moreover, the best method in the single-objective group is MSTLRO; as a result, when we compare MSTLRO-CXP with single-objective configurations, the comparison is

between MSTLRO-CXP and a very similar approach. This is why the performance of MSTLRO-CXP in Figure 4-11 (at 37.5%) does not seem impressive.

Daily Trading									
Objectives	Percentage to outperform buy-and-hold								Overall
	D01	D02	D03	D04	N_D01	N_D02	N_D03	N_D04	
Single Objective	Single Objective								
MR (1-Obj)	0	35	90	30	85	0	100	100	55.00%
PC_LK12 (1-Obj)	0	75	10	40	0	0	10	75	26.25%
PC_LK24 (1-Obj)	0	10	85	35	10	0	100	60	37.50%
SHARO (1-Obj)	5	35	100	35	70	0	100	100	55.63%
MSTLRO (1-Obj)	100	100	95	95	45	0	100	100	79.38%
Normal Trading	Normal Trading								
MSTLRO-CXP (2-Objs-S2)	100	100	100	100	25	0	100	100	78.13%
Majority Trading	Majority Trading								
MSTLRO-CXP (2-Objs-MJV-S2)	100	100	90	95	30	0	100	100	76.88%

Table 4-20: Performance of single-objective methods and MSTLRO-CXP method in comparison with the buy-and-hold strategy for daily trading splits.

4.5 Rule Analysis

In section 3.5 we analyzed the trading rules that emerged from single-objective experiments; now it is time for trading rules from the multi-objective approach to be analyzed in this section. Please see the description in section 3.5 for how we process the results to get the graphs and tables we show here.

4.5.1 Rule Analysis for Monthly Trading

4.5.1.1 Profit Driven Approach

From Figure 4-13 and Table 4-21, we see that the four most prominent technical indicators are MX-1, MX-2, MN-2 and MN-1 respectively; they share 71% of the whole proportion of indicators found in evolved rules. This means roughly 71% of the time the leaf nodes in the trading rule trees were from the local minima and local maxima group. Considering the Level2 structure (Figure 4-14 and Table 4-22), we found that “MN-2 < MX-1” and “MX-2 < MN-1” are the most common expressions used, accounting for 41.22% of the depth-2 subtrees found in evolved rules. Recall that there are 24 possible outcomes of the binary relational expression of the form of “TI1 *operator* TI2”, where TI1 and TI2 are indicators from the group of four (MN-1, MN-2, MX-1 and MX-2) and *operator* is either < or >.

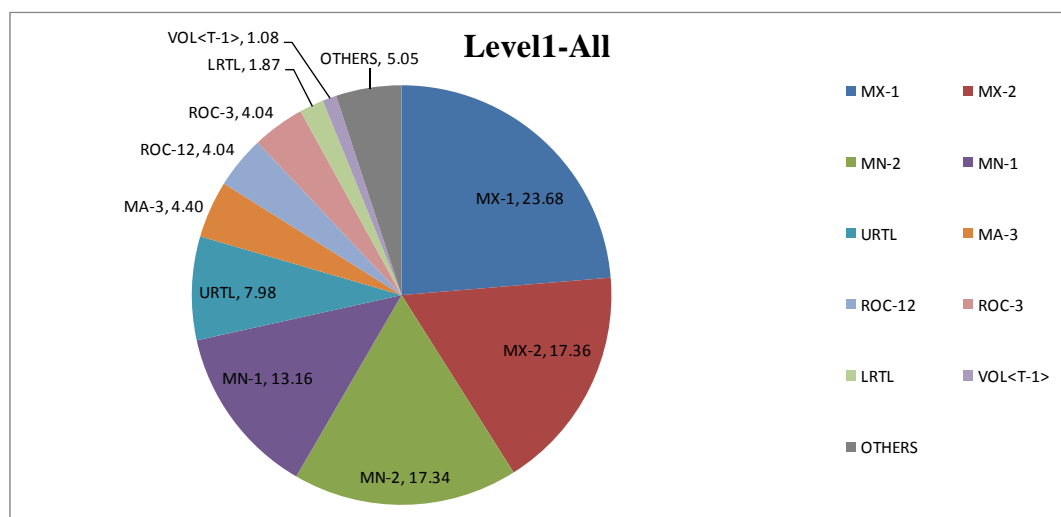


Figure 4-13: Level1 Monthly Proportion-Profit Driven Approach.

Level1-1					
#	All	2CSMR	2SMR_CXP	CMR_CPC-LK12	CMR_CPC-LK24
1	MX-1 (23.68)	MX-1 (24.09)	MX-1 (22.36)	MX-1 (29.93)	MX-1 (22.53)
2	MX-2 (17.36)	MX-2 (18.85)	MX-2 (16.82)	MN-2 (17.28)	MX-2 (21.72)
3	MN-2 (17.34)	MN-2 (17.00)	MN-2 (16.48)	MX-2 (14.45)	MN-2 (18.19)
4	MN-1 (13.16)	MN-1 (13.76)	MN-1 (13.40)	MN-1 (11.26)	MN-1 (15.97)
5	URL (7.98)	URL (8.56)	URL (7.59)	MA-3 (9.11)	URL (9.20)
6	MA-3 (4.40)	MA-3 (5.10)	MA-3 (5.13)	URL (7.74)	ROC-12 (2.76)
7	ROC-12 (4.04)	ROC-12 (3.09)	ROC-12 (4.43)	ROC-3 (2.57)	ROC-3 (2.76)
8	ROC-3 (4.04)	ROC-3 (3.09)	ROC-3 (4.43)	ROC-12 (2.57)	MA-3 (2.42)
9	LRTL (1.87)	LRTL (1.96)	LRTL (2.37)	LRTL (1.26)	LRTL (1.28)
10	VOL<T-1> (1.08)	MA-10 (1.21)	MA-6 (1.35)	MA-10 (1.08)	MA-6 (0.95)
OTHERS	OTHERS (5.05)	OTHERS (3.27)	OTHERS (5.64)	OTHERS (2.75)	OTHERS (2.24)
Level1-2					
#	MR_CXP	MR_PC-LK12_CXP	MR_PC-LK24_CXP	PC-LK12_CXP	PC-LK24_CXP
1	MX-1 (23.79)	MX-1 (28.17)	MX-1 (23.25)	MX-1 (18.28)	MX-2 (21.62)
2	MX-2 (20.42)	MN-2 (19.46)	MX-2 (21.19)	MN-2 (14.93)	MX-1 (17.79)
3	MN-2 (17.99)	MX-2 (13.61)	MN-2 (17.90)	ROC-3 (10.16)	MN-2 (15.62)
4	MN-1 (15.28)	MN-1 (10.54)	MN-1 (15.28)	ROC-12 (10.16)	MN-1 (15.34)
5	URL (8.07)	URL (7.86)	URL (8.14)	VOL<T> (8.29)	URL (7.03)
6	ROC-3 (3.36)	MA-3 (6.81)	ROC-12 (3.17)	VOL<T-1> (8.29)	ROC-12 (6.04)
7	ROC-12 (3.36)	ROC-3 (3.25)	ROC-3 (3.17)	URL (7.39)	ROC-3 (6.04)
8	MA-3 (3.23)	ROC-12 (3.25)	LRTL (2.43)	MN-1 (5.19)	LRTL (1.83)
9	LRTL (1.46)	LRTL (2.01)	MA-3 (2.18)	MA-3 (3.57)	CPRICE<T> (1.50)
10	MA-10 (0.69)	MA-10 (1.33)	MA-10 (0.69)	MX-2 (3.51)	MA-3 (1.12)
OTHERS	OTHERS (2.35)	OTHERS (3.72)	OTHERS (2.59)	OTHERS (10.22)	OTHERS (6.07)

Table 4-21: Level1 Monthly Table-Profit Driven Approach.

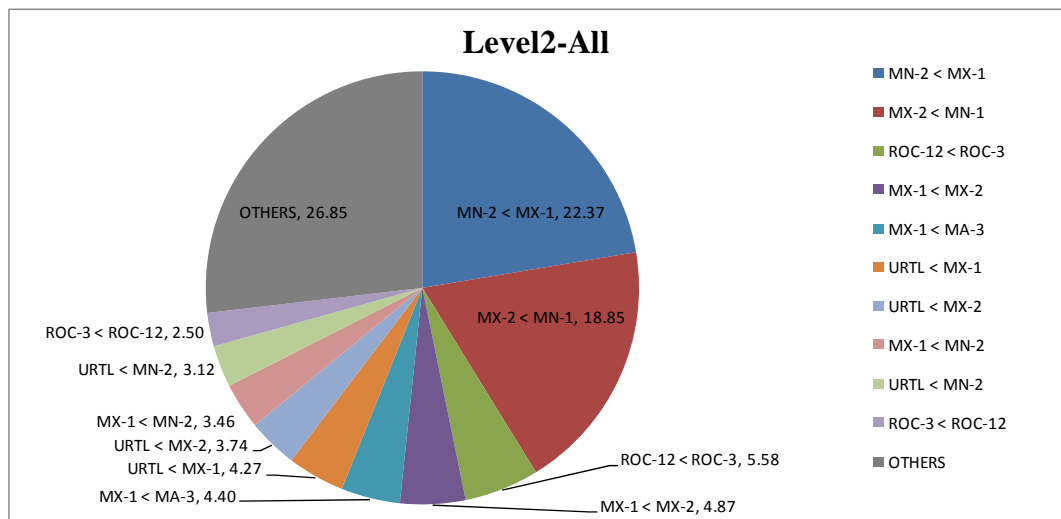


Figure 4-14: Level2 Monthly Proportion-Profit Driven Approach.

Level2-1					
#	All	2CSMR	2SMR_CXP	CMR_CPC-LK12	CMR_CPC-LK24
1	MN-2 < MX-1 (22.37)	MN-2 < MX-1 (23.04)	MN-2 < MX-1 (20.91)	MN-2 < MX-1 (24.21)	MX-2 < MN-1 (24.14)
2	MX-2 < MN-1 (18.85)	MX-2 < MN-1 (22.04)	MX-2 < MN-1 (18.34)	MX-2 < MN-1 (14.58)	MN-2 < MX-1 (23.48)
3	ROC-12 < ROC-3 (5.58)	MX-1 < MA-3 (5.19)	ROC-12 < ROC-3 (5.82)	MX-1 < MA-3 (8.56)	URL < MX-2 (6.32)
4	MX-1 < MX-2 (4.87)	URL < MX-2 (5.09)	MX-1 < MA-3 (5.12)	MA-3 < MX-1 (5.68)	MX-1 < MX-2 (6.21)
5	MX-1 < MA-3 (4.40)	ROC-12 < ROC-3 (4.63)	MX-1 < MX-2 (4.09)	MX-1 < MX-2 (5.21)	URL < MX-1 (5.41)
6	URL < MX-1 (4.27)	URL < MX-1 (4.54)	URL < MX-1 (3.68)	URL < MX-1 (4.72)	ROC-12 < ROC-3 (4.54)
7	URL < MX-2 (3.74)	MX-1 < MX-2 (3.86)	URL < MN-2 (3.38)	ROC-12 < ROC-3 (4.05)	URL < MN-2 (3.83)
8	MX-1 < MN-2 (3.46)	MA-3 < MX-1 (3.39)	URL < MX-2 (3.36)	MX-1 < MN-2 (3.98)	MX-1 < MA-3 (3.23)
9	URL < MN-2 (3.12)	MX-1 < MN-2 (3.03)	MA-3 < MX-1 (3.17)	MX-1 < URL (3.92)	MX-1 < MN-2 (3.05)
10	ROC-3 < ROC-12 (2.50)	URL < MN-2 (2.91)	ROC-3 < ROC-12 (3.04)	URL < MN-2 (2.70)	MN-1 < MX-2 (2.79)
OTHERS	OTHERS (26.85)	OTHERS (22.29)	OTHERS (29.09)	OTHERS (22.38)	OTHERS (16.99)
Level2-2					
#	MR_CXP	MR_PC-LK12_CXP	MR_PC-LK24_CXP	PC-LK12_CXP	PC-LK24_CXP
1	MX-2 < MN-1 (24.56)	MN-2 < MX-1 (26.24)	MN-2 < MX-1 (23.21)	MN-2 < MX-1 (15.32)	MX-2 < MN-1 (22.02)
2	MN-2 < MX-1 (23.62)	MX-2 < MN-1 (14.76)	MX-2 < MN-1 (22.86)	ROC-3 < ROC-12 (10.70)	MN-2 < MX-1 (17.98)
3	MX-1 < MX-2 (5.61)	MX-1 < MA-3 (6.54)	MX-1 < MX-2 (6.79)	ROC-12 < ROC-3 (9.61)	ROC-12 < ROC-3 (8.74)
4	URL < MX-1 (5.24)	ROC-12 < ROC-3 (5.13)	URL < MX-1 (5.69)	VOL<T> < VOL<T-1> (8.45)	MX-1 < MX-2 (6.67)
5	ROC-12 < ROC-3 (4.81)	MA-3 < MX-1 (4.60)	ROC-12 < ROC-3 (4.84)	VOL<T-1> < VOL<T> (8.13)	URL < MX-2 (4.96)
6	MX-1 < MA-3 (4.60)	URL < MN-2 (3.94)	URL < MX-2 (4.36)	MX-1 < MN-2 (5.85)	ROC-3 < ROC-12 (3.35)
7	URL < MN-2 (4.37)	MX-1 < MN-2 (3.88)	MX-1 < URL (3.37)	MX-1 < URL (5.17)	URL < MX-1 (3.20)
8	URL < MX-2 (3.65)	MX-1 < MX-2 (3.84)	MN-1 < MX-2 (3.33)	MA-3 < MN-1 (2.71)	MX-1 < MN-2 (2.49)
9	MX-1 < MN-2 (3.46)	URL < MX-1 (3.42)	URL < MN-2 (3.32)	URL < MN-2 (1.40)	MN-1 < MX-2 (2.15)
10	MN-1 < MX-2 (2.22)	MX-1 < URL (3.04)	MX-1 < MA-3 (2.55)	MX-1 < MA-3 (1.38)	MN-1 < MN-2 (1.82)
OTHERS	OTHERS (17.86)	OTHERS (24.62)	OTHERS (19.68)	OTHERS (31.26)	OTHERS (26.63)

Table 4-22: Level2 Monthly Table-Profit Driven Approach.

4.5.1.2 Risk-Adjusted Approach

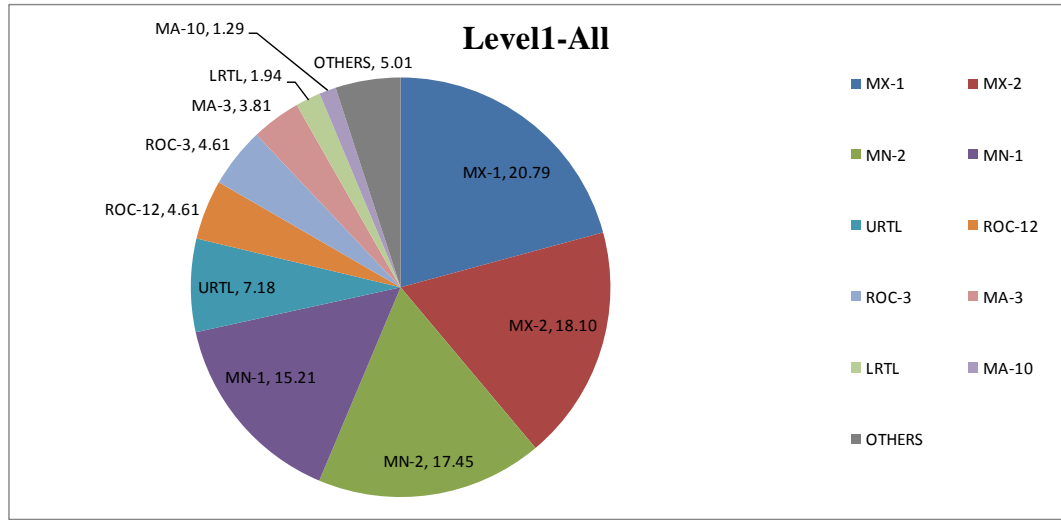


Figure 4-15: Level1 Monthly Proportion-Risk-Adjusted Approach.

Level1-1							
#	All	CMR CPC-LK12 MSTLRO	CMR CPC-LK12 SHARO	CMR CPC-LK24 MSTLRO	CMR CPC-LK24 SHARO	CMR MMDD	MR MMDD CXP
1	MX-1 (20.79)	MX-1 (23.58)	MX-1 (25.35)	MX-2 (20.37)	MX-2 (22.36)	MX-2 (23.63)	MX-2 (21.68)
2	MX-2 (18.10)	MN-2 (19.17)	MN-2 (17.72)	MX-1 (20.30)	MX-1 (21.52)	MN-1 (20.45)	MN-1 (18.84)
3	MN-2 (17.45)	MX-2 (16.37)	MX-2 (16.38)	MN-2 (18.60)	MN-2 (17.34)	MX-1 (17.96)	MX-1 (18.52)
4	MN-1 (15.21)	MN-1 (13.20)	MN-1 (13.10)	MN-1 (16.16)	MN-1 (15.85)	MN-2 (14.10)	MN-2 (16.46)
5	URTL (7.18)	URTL (8.77)	URTL (7.85)	URTL (7.77)	URTL (8.72)	URTL (8.16)	URTL (7.88)
6	ROC-12 (4.61)	ROC-12 (3.98)	MA-3 (6.59)	ROC-12 (4.01)	ROC-3 (3.31)	MA-3 (3.78)	ROC-3 (4.28)
7	ROC-3 (4.61)	ROC-3 (3.98)	ROC-12 (3.84)	ROC-3 (4.01)	ROC-12 (3.31)	ROC-12 (3.04)	ROC-12 (4.28)
8	MA-3 (3.81)	MA-3 (3.63)	ROC-3 (3.84)	MA-3 (1.99)	MA-3 (2.91)	ROC-3 (3.04)	MA-3 (2.51)
9	LRTL (1.94)	LRTL (1.89)	LRTL (1.19)	LRTL (1.76)	LRTL (1.81)	LRTL (1.71)	LRTL (1.47)
10	MA-10 (1.29)	MA-10 (1.37)	MA-6 (1.01)	MA-10 (1.22)	MA-6 (0.60)	MA-10 (0.80)	MA-10 (1.01)
OTHERS	OTHERS (5.01)	OTHERS (4.06)	OTHERS (3.13)	OTHERS (3.79)	OTHERS (2.29)	OTHERS (3.32)	OTHERS (3.07)
Level1-2							
#	MR PC-LK12 MSTLRO CXP	MR PC-LK12 SHARO CXP	MR PC-LK24 MSTLRO CXP	MR PC-LK24 SHARO CXP	MSTLRO CPC-LK12	MSTLRO CPC-LK24	MSTLRO CXP
1	MX-1 (22.24)	MX-1 (25.40)	MX-2 (20.10)	MX-2 (21.85)	MX-1 (22.64)	MN-2 (19.33)	MN-1 (16.68)
2	MN-2 (19.80)	MN-2 (17.20)	MX-1 (19.41)	MX-1 (21.38)	MN-2 (20.82)	MX-2 (17.09)	MX-2 (15.15)
3	MX-2 (14.44)	MX-2 (16.33)	MN-2 (17.82)	MN-1 (16.95)	MX-2 (13.84)	MX-1 (15.27)	ROC-3 (7.96)
4	MN-1 (12.80)	MN-1 (13.17)	MN-1 (15.73)	MN-2 (16.11)	MN-1 (13.57)	MN-1 (15.18)	ROC-12 (7.96)
5	URTL (7.36)	URTL (8.74)	URTL (8.86)	URTL (7.94)	URTL (7.39)	URTL (7.58)	MX-1 (7.33)
6	ROC-12 (5.05)	MA-3 (5.76)	ROC-12 (4.06)	ROC-12 (3.53)	ROC-12 (4.91)	ROC-3 (5.85)	MN-2 (7.14)
7	ROC-3 (5.05)	ROC-12 (4.52)	ROC-3 (4.06)	ROC-3 (3.53)	ROC-3 (4.91)	ROC-12 (5.85)	VOL<T-1> (5.76)
8	MA-3 (4.12)	ROC-3 (4.52)	LRTL (2.16)	MA-3 (3.30)	MA-3 (4.25)	LRTL (2.73)	VOL<T-1> (5.76)
9	LRTL (2.01)	LRTL (1.71)	MA-3 (2.01)	LRTL (1.80)	MA-10 (1.49)	MA-3 (2.00)	URTL (4.93)
10	MA-10 (1.81)	MA-10 (1.54)	MA-10 (1.04)	MA-6 (0.90)	LRTL (1.47)	VOL<T-1> (1.59)	CPRICE<T-1> (4.68)
OTHERS	OTHERS (5.33)	OTHERS (3.11)	OTHERS (4.75)	OTHERS (2.82)	OTHERS (4.71)	OTHERS (7.53)	OTHERS (16.08)
Level1-3							
#	MSTLRO PC-LK12 CXP	MSTLRO PC-LK24 CXP	SHARO CPC-LK12	SHARO CPC-LK24	SHARO CXP	SHARO PC-LK12 CXP	SHARO PC-LK24 CXP
1	MX-1 (21.49)	MN-2 (20.31)	MX-1 (25.71)	MX-1 (20.17)	MX-1 (20.35)	MX-1 (24.38)	MX-2 (21.15)
2	MN-2 (19.49)	MX-2 (18.18)	MN-2 (17.65)	MX-2 (20.13)	MX-2 (19.96)	MN-2 (17.80)	MX-1 (20.48)
3	MX-2 (14.19)	MX-1 (16.61)	MX-2 (15.24)	MN-2 (16.24)	MN-1 (18.05)	MX-2 (16.23)	MN-1 (17.20)
4	MN-1 (12.86)	MN-1 (16.00)	MN-1 (12.65)	MN-1 (15.21)	MN-2 (14.32)	MN-1 (14.19)	MN-2 (16.49)
5	URTL (7.25)	URTL (8.20)	MA-3 (6.21)	URTL (5.71)	MA-3 (5.78)	MA-3 (6.20)	URTL (7.18)
6	ROC-3 (5.26)	ROC-12 (4.25)	ROC-12 (5.22)	ROC-3 (5.63)	ROC-12 (5.27)	URTL (5.51)	ROC-3 (4.14)
7	ROC-12 (5.26)	ROC-3 (4.25)	ROC-3 (5.22)	ROC-12 (5.63)	ROC-3 (5.27)	ROC-12 (4.42)	ROC-12 (4.14)
8	MA-3 (3.83)	LRTL (2.44)	URTL (4.54)	MA-3 (3.36)	URTL (4.47)	ROC-3 (4.42)	MA-3 (2.88)
9	LRTL (2.03)	MA-3 (1.70)	MA-10 (1.60)	LRTL (2.03)	LRTL (1.34)	LRTL (1.48)	LRTL (2.16)
10	MA-10 (1.92)	CPRICE<T-1> (1.31)	LRTL (1.56)	CPRICE<T-1> (1.03)	CPRICE<T-1> (1.10)	MA-10 (1.15)	MA-10 (1.29)
OTHERS	OTHERS (6.42)	OTHERS (6.72)	OTHERS (4.41)	OTHERS (4.87)	OTHERS (4.09)	OTHERS (4.22)	OTHERS (2.91)

Table 4-23: Level1 Monthly Table- Risk-Adjusted Approach.

The most active technical indicator group from the risk-adjusted approach experiments are, again, the local minima and local maxima group with approximately 71%, as shown in Figure 4-15, and when turning to see the connection between those indicators in Figure 4-16, “MX-2 < MN-1” and “MN-2 < MX-1” accounted for 41.84% of the structures at Level2.

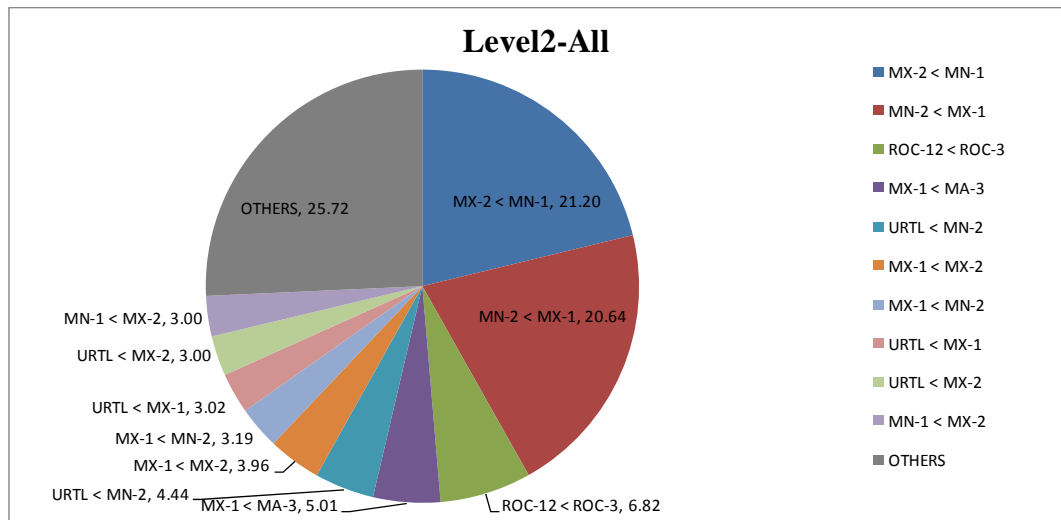


Figure 4-16: Level2 Monthly Proportion-Risk-Adjusted Approach.

Level2-1							
#	All	CMR CPC-LK12 MSTLRO	CMR CPC-LK12 SHARO	CMR CPC-LK24 MSTLRO	CMR CPC-LK24 SHARO	CMR MMDD	MR MMDD CXP
1	MX-2 < MN-1 (21.20)	MN-2 < MX-1 (25.27)	MN-2 < MX-1 (22.29)	MX-2 < MN-1 (23.94)	MX-2 < MN-1 (24.71)	MX-2 < MN-1 (30.74)	MX-2 < MN-1 (28.25)
2	MN-2 < MX-1 (20.64)	MX-2 < MN-1 (19.60)	MN-2 < MN-1 (18.54)	MN-2 < MX-1 (21.54)	MN-2 < MX-1 (21.93)	MN-2 < MX-1 (16.48)	MN-2 < MX-1 (19.37)
3	ROC-12 < ROC-3 (6.82)	ROC-12 < ROC-3 (6.61)	MX-1 < MA-3 (7.87)	ROC-12 < ROC-3 (5.45)	URTL < MX-2 (6.35)	MX-1 < MA-3 (6.51)	ROC-12 < ROC-3 (5.63)
4	MX-1 < MA-3 (5.01)	URTL < MN-2 (5.80)	ROC-12 < ROC-3 (6.79)	URTL < MN-2 (5.36)	MX-1 < MX-2 (5.46)	MN-1 < MX-2 (4.61)	URTL < MN-2 (5.49)
5	URTL < MN-2 (4.44)	MX-1 < MA-3 (5.39)	MX-1 < MX-2 (4.50)	MX-1 < MX-2 (5.23)	ROC-12 < ROC-3 (4.81)	URTL < MN-2 (4.41)	MX-1 < MX-2 (4.02)
6	MX-1 < MX-2 (3.96)	URTL < MX-1 (4.38)	URTL < MN-2 (4.26)	URTL < MX-2 (3.96)	URTL < MX-1 (4.45)	URTL < MX-2 (3.80)	MX-1 < MA-3 (3.99)
7	MX-1 < MN-2 (3.19)	MX-1 < MX-2 (3.92)	MX-1 < MN-2 (3.69)	MX-1 < MN-2 (3.89)	MX-1 < MA-3 (4.10)	MX-1 < MX-2 (3.75)	URTL < MX-2 (3.59)
8	URTL < MX-1 (3.02)	URTL < MX-2 (3.12)	URTL < MX-1 (3.58)	URTL < MX-1 (3.20)	URTL < MN-2 (3.52)	ROC-12 < ROC-3 (3.59)	URTL < MX-1 (3.16)
9	URTL < MX-2 (3.00)	MX-1 < MN-2 (2.32)	URTL < MX-2 (3.39)	MN-1 < MX-2 (3.16)	MX-1 < MN-2 (3.47)	URTL < MX-1 (3.42)	MN-1 < MX-2 (3.08)
10	MN-1 < MX-2 (3.00)	MN-1 < MX-2 (2.18)	MA-3 < MX-1 (2.76)	MX-1 < MA-3 (2.95)	MN-1 < MX-2 (2.76)	MX-1 < MN-2 (2.55)	ROC-3 < ROC-12 (2.93)
OTHERS	OTHERS (25.72)	OTHERS (21.41)	OTHERS (22.34)	OTHERS (21.31)	OTHERS (18.45)	OTHERS (20.12)	OTHERS (20.49)

Level2-2							
#	MR PC-LK12 MSTLRO CXP	MR PC-LK12 SHARO CXP	MR PC-LK24 MSTLRO CXP	MR PC-LK24 SHARO CXP	MSTLRO CPC-LK12	MSTLRO CPC-LK24	MSTLRO CXP
1	MN-2 < MX-1 (23.61)	MN-2 < MX-1 (22.79)	MX-2 < MN-1 (21.95)	MX-2 < MN-1 (24.66)	MN-2 < MX-1 (24.15)	MX-2 < MN-1 (20.76)	MX-2 < MN-1 (17.38)
2	MX-2 < MN-1 (17.90)	MX-2 < MN-1 (18.09)	MN-2 < MX-1 (21.07)	MN-2 < MX-1 (19.10)	MX-2 < MN-1 (18.05)	MN-2 < MX-1 (18.50)	ROC-12 < ROC-3 (8.19)
3	ROC-12 < ROC-3 (8.17)	ROC-12 < ROC-3 (7.63)	ROC-12 < ROC-3 (5.34)	ROC-12 < ROC-3 (5.59)	ROC-12 < ROC-3 (7.60)	ROC-12 < ROC-3 (8.49)	ROC-3 < ROC-12 (7.73)
4	URTL < MN-2 (6.10)	MX-1 < MA-3 (6.45)	URTL < MX-2 (4.72)	MX-1 < MX-2 (5.55)	URTL < MN-2 (6.38)	URTL < MN-2 (6.56)	VOL<T> < VOL<T-1> (5.90)
5	MX-1 < MA-3 (4.89)	MX-1 < MX-2 (4.88)	URTL < MX-1 (4.70)	URTL < MX-2 (5.11)	MX-1 < MA-3 (5.37)	MX-1 < MN-2 (3.46)	VOL<T> < VOL<T-1> (5.62)
6	MX-1 < MX-2 (2.92)	URTL < MX-1 (4.46)	URTL < MN-2 (4.67)	URTL < MX-1 (4.91)	MX-1 < MN-2 (3.21)	MN-1 < MX-2 (3.38)	MX-1 < MA-3 (4.56)
7	MX-1 < MN-2 (2.86)	MX-1 < MN-2 (2.93)	MX-1 < MX-2 (4.44)	MX-1 < MA-3 (4.90)	URTL < MX-1 (2.78)	ROC-3 < ROC-12 (3.21)	MN-1 < MX-2 (4.36)
8	URTL < MX-1 (2.63)	MA-3 < MX-1 (2.64)	MN-1 < MX-2 (3.02)	MX-1 < MN-2 (4.06)	MX-1 < MX-2 (2.33)	URTL < MX-2 (2.88)	MN-2 < MX-1 (3.97)
9	MN-1 < MX-2 (1.99)	URTL < MN-2 (2.56)	MX-1 < MN-2 (2.97)	MN-1 < MX-2 (3.27)	MN-1 < MX-2 (2.28)	MX-1 < MX-2 (2.42)	URTL < MN-2 (2.29)
10	ROC-3 < ROC-12 (1.93)	URTL < MX-2 (2.09)	ROC-3 < ROC-12 (2.77)	URTL < MN-2 (2.93)	ROC-3 < ROC-12 (2.22)	MX-1 < MA-3 (2.42)	LRIL < MN-1 (1.50)
OTHERS	OTHERS (27.01)	OTHERS (25.49)	OTHERS (24.35)	OTHERS (19.92)	OTHERS (25.64)	OTHERS (27.91)	OTHERS (38.50)

Level2-3							
#	MSTLRO PC-LK12 CXP	MSTLRO PC-LK24 CXP	SHARO CPC-LK12	SHARO CPC-LK24	SHARO CXP	SHARO PC-LK12 CXP	SHARO PC-LK24 CXP
1	MN-2 < MX-1 (23.74)	MX-2 < MN-1 (21.70)	MN-2 < MX-1 (22.63)	MX-2 < MN-1 (20.88)	MX-2 < MN-1 (25.57)	MN-2 < MX-1 (22.77)	MX-2 < MN-1 (25.01)
2	MX-2 < MN-1 (16.84)	MN-2 < MX-1 (21.05)	MX-2 < MN-1 (15.88)	MN-2 < MX-1 (18.87)	MN-2 < MX-1 (16.32)	MX-2 < MN-1 (17.97)	MN-2 < MX-1 (19.99)
3	ROC-12 < ROC-3 (7.97)	URTL < MN-2 (8.06)	ROC-12 < ROC-3 (7.39)	ROC-12 < ROC-3 (8.64)	MX-1 < MA-3 (9.55)	MX-1 < MA-3 (7.36)	ROC-12 < ROC-3 (6.41)
4	URTL < MN-2 (6.33)	ROC-12 < ROC-3 (6.64)	MX-1 < MA-3 (6.50)	MX-1 < MA-3 (4.70)	ROC-12 < ROC-3 (8.14)	ROC-12 < ROC-3 (6.22)	URTL < MX-2 (5.37)
5	MX-1 < MA-3 (4.97)	MN-1 < MX-2 (3.79)	MX-1 < MN-2 (4.68)	MX-1 < MX-2 (4.46)	MX-1 < MX-2 (4.43)	MX-1 < MX-2 (4.48)	MX-1 < MX-2 (4.60)
6	ROC-3 < ROC-12 (2.54)	MX-1 < MX-2 (3.09)	MX-1 < MX-2 (4.68)	MN-1 < MX-2 (4.10)	MX-1 < MN-2 (3.96)	MN-1 < MX-2 (3.68)	MX-1 < MA-3 (4.42)
7	MX-1 < MN-2 (2.52)	MX-1 < MX-2 (2.50)	MN-1 < MX-2 (3.30)	URTL < MX-2 (3.80)	MN-1 < MX-2 (3.95)	URTL < MN-2 (3.64)	URTL < MX-1 (3.71)
8	MX-1 < MX-2 (2.45)	URTL < MX-2 (2.43)	ROC-3 < ROC-12 (3.05)	MX-1 < MN-2 (3.70)	URTL < MN-2 (2.45)	MX-1 < MN-2 (3.45)	MX-1 < MN-2 (3.64)
9	URTL < MX-1 (2.45)	MN-1 < MN-2 (2.37)	MA-3 < MX-1 (2.58)	MX-2 < MX-1 (2.68)	ROC-3 < ROC-12 (2.39)	ROC-3 < ROC-12 (2.61)	URTL < MN-2 (2.70)
10	MN-1 < MX-2 (1.99)	MX-1 < MA-3 (1.95)	URTL < MN-2 (2.11)	ROC-3 < ROC-12 (2.62)	URTL < MX-1 (1.93)	MA-3 < MX-1 (2.07)	MN-1 < MX-2 (2.49)
OTHERS	OTHERS (28.19)	OTHERS (26.43)	OTHERS (27.19)	OTHERS (25.54)	OTHERS (21.30)	OTHERS (25.74)	OTHERS (21.66)

Table 4-24: Level2 Monthly Table- Risk-Adjusted Approach.

4.5.2 Rule Analysis for Weekly Trading

4.5.2.1 Profit Driven Approach

What are the structures that appear in the more frequent environment of weekly trading?

Figure 4-17 tells us that the local minima and local maxima group remain prominent with about 60% representation among the leaves of the selected trading rules, but now

we also find that the trend line indicators (LRTL and URTL), which make use of the local minima and local maxima, have more active roles in the case of weekly trading, accounting together for 18.20%. These findings are reflected in the Level2 structures; the first and second most used expressions ($MX-2 < MN-1$ ” and “ $MN2 < MX-1$ ”) account for 28.85% of depth-2 subtrees, while “ $MX-1 < MA-3$ ” (5.81%), “ $MX-1 < UTRL$ ” (4.67%) and “ $UTRL < MN-2$ ” (4.29%) are the third, fourth and fifth ranked expressions respectively.

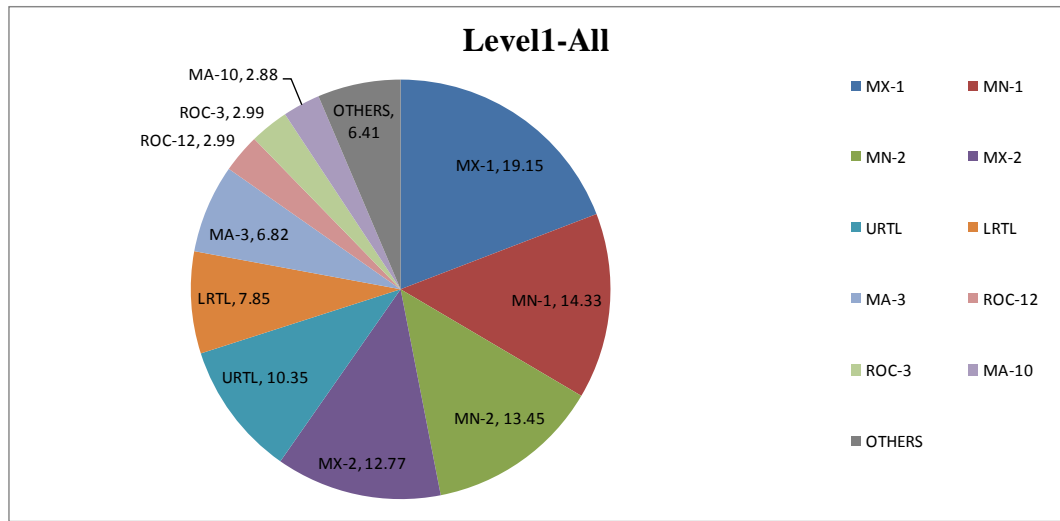


Figure 4-17: Level1 Weekly Proportion-Profit Driven Approach.

Level1-1					
#	All	2CSMR	2SMR_CXP	CMR CPC-LK12	CMR CPC-LK24
1	MX-1 (19.15)	MX-1 (18.75)	MX-1 (17.24)	MX-1 (21.57)	MX-1 (20.94)
2	MN-1 (14.33)	MN-1 (16.20)	MN-1 (15.74)	MN-1 (15.00)	MN-2 (14.46)
3	MN-2 (13.45)	MX-2 (14.60)	MX-2 (13.47)	MN-2 (14.43)	MN-1 (14.11)
4	MX-2 (12.77)	MN-2 (14.42)	MN-2 (12.54)	MX-2 (13.05)	MX-2 (12.58)
5	URTL (10.35)	URTL (9.82)	URTL (9.82)	URTL (10.15)	URTL (11.23)
6	LRTL (7.85)	LRTL (7.98)	LRTL (8.72)	MA-3 (7.78)	MA-3 (8.35)
7	MA-3 (6.82)	MA-3 (7.17)	MA-3 (6.53)	LRTL (7.38)	LRTL (8.16)
8	ROC-12 (2.99)	MA-10 (3.24)	MA-10 (3.19)	MA-10 (2.93)	MA-10 (3.10)
9	ROC-3 (2.99)	MA-6 (1.76)	ROC-3 (2.91)	ROC-3 (1.77)	ROC-12 (1.27)
10	MA-10 (2.88)	ROC-12 (1.72)	ROC-12 (2.91)	ROC-12 (1.77)	ROC-3 (1.27)
OTHERS	OTHERS (6.41)	OTHERS (4.33)	OTHERS (6.94)	OTHERS (4.17)	OTHERS (4.53)
Level1-2					
#	MR_CXP	MR_PC-LK12_CXP	MR_PC-LK24_CXP	PC-LK12_CXP	PC-LK24_CXP
1	MX-1 (19.17)	MX-1 (20.69)	MX-1 (20.70)	MX-1 (17.24)	MX-1 (13.85)
2	MN-1 (15.48)	MN-1 (13.98)	MN-1 (14.17)	MN-1 (12.16)	URTL (12.68)
3	MX-2 (14.68)	MN-2 (13.78)	MN-2 (13.51)	MN-2 (11.90)	MN-2 (12.17)
4	MN-2 (13.26)	MX-2 (12.20)	MX-2 (13.19)	MX-2 (10.53)	MN-1 (10.80)
5	URTL (9.69)	URTL (9.44)	URTL (11.44)	URTL (9.30)	MX-2 (9.34)
6	LRTL (8.27)	LRTL (8.31)	LRTL (8.18)	LRTL (7.57)	ROC-12 (8.01)
7	MA-3 (6.75)	MA-3 (7.63)	MA-3 (6.63)	ROC-12 (7.09)	ROC-3 (8.01)
8	MA-10 (3.46)	MA-10 (3.00)	MA-10 (3.22)	ROC-3 (7.09)	LRTL (5.10)
9	ROC-3 (2.11)	ROC-3 (2.74)	ROC-12 (1.72)	MA-3 (4.56)	MA-3 (4.83)
10	ROC-12 (2.11)	ROC-12 (2.74)	ROC-3 (1.72)	MA-10 (2.11)	VOL<T-1> (2.61)
OTHERS	OTHERS (5.03)	OTHERS (5.50)	OTHERS (5.51)	OTHERS (10.46)	OTHERS (12.62)

Table 4-25: Level1 Weekly Table-Profit Driven Approach.

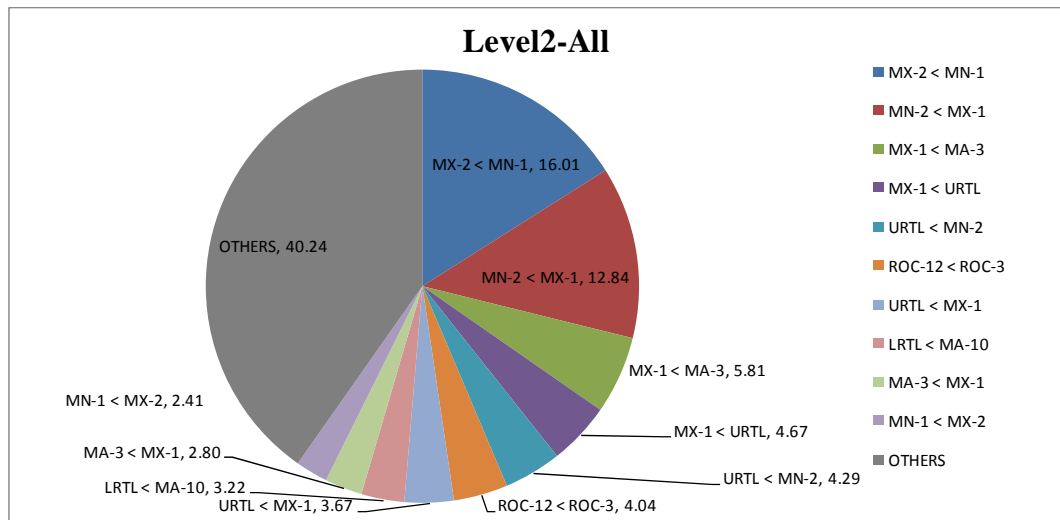


Figure 4-18: Level2 Weekly Proportion-Profit Driven Approach.

Level2-1					
#	All	2CSMR	2SMR_CXP	CMR_CPC-LK12	CMR_CPC-LK24
1	MX-2 < MN-1 (16.01)	MX-2 < MN-1 (19.48)	MX-2 < MN-1 (16.31)	MX-2 < MN-1 (17.81)	MX-2 < MN-1 (16.27)
2	MN-2 < MX-1 (12.84)	MN-2 < MX-1 (13.56)	MN-2 < MX-1 (11.61)	MN-2 < MX-1 (14.87)	MN-2 < MX-1 (15.30)
3	MX-1 < MA-3 (5.81)	MX-1 < MA-3 (6.52)	MX-1 < MA-3 (5.83)	MX-1 < MA-3 (6.90)	MX-1 < MA-3 (7.08)
4	MX-1 < URTL (4.67)	URTL < MN-2 (4.98)	URTL < MN-2 (3.80)	MX-1 < URTL (5.38)	URTL < MN-2 (6.23)
5	URTL < MN-2 (4.29)	URTL < MX-1 (3.48)	URTL < MX-1 (3.45)	URTL < MN-2 (5.20)	MX-1 < URTL (4.35)
6	ROC-12 < ROC-3 (4.04)	MA-3 < MX-1 (3.36)	ROC-12 < ROC-3 (3.34)	LRTL < MA-10 (4.41)	LRTL < MA-10 (3.95)
7	URTL < MX-1 (3.67)	LRTL < MA-10 (3.29)	MX-1 < URTL (3.02)	URTL < MX-1 (3.95)	URTL < MX-1 (3.44)
8	LRTL < MA-10 (3.22)	MX-1 < URTL (2.84)	MN-1 < MX-2 (2.92)	MA-3 < MX-1 (3.08)	MA-3 < MX-1 (3.39)
9	MA-3 < MX-1 (2.80)	MN-1 < MX-2 (2.76)	MA-3 < MX-1 (2.67)	MX-1 < MN-2 (2.69)	LRTL < MX-1 (2.93)
10	MN-1 < MX-2 (2.41)	MX-1 < MN-2 (2.37)	LRTL < MA-10 (2.65)	ROC-12 < ROC-3 (2.54)	MN-1 < MX-2 (2.63)
OTHERS	OTHERS (40.24)	OTHERS (37.35)	OTHERS (44.40)	OTHERS (33.15)	OTHERS (34.43)
Level2-2					
#	MR_CXP	MR_PC-LK12_CXP	MR_PC-LK24_CXP	PC-LK12_CXP	PC-LK24_CXP
1	MX-2 < MN-1 (18.83)	MX-2 < MN-1 (15.86)	MX-2 < MN-1 (16.55)	MX-2 < MN-1 (12.17)	ROC-12 < ROC-3 (11.12)
2	MN-2 < MX-1 (13.30)	MN-2 < MX-1 (14.54)	MN-2 < MX-1 (12.55)	ROC-12 < ROC-3 (10.73)	MX-2 < MN-1 (7.96)
3	MX-1 < MA-3 (6.10)	MX-1 < MA-3 (7.02)	MX-1 < MA-3 (6.22)	MN-2 < MX-1 (10.68)	MX-1 < URTL (7.82)
4	URTL < MN-2 (4.89)	MX-1 < URTL (4.59)	URTL < MN-2 (4.92)	MX-1 < URTL (6.62)	MN-2 < MX-1 (6.93)
5	LRTL < MA-10 (4.11)	URTL < MN-2 (3.85)	MX-1 < URTL (4.90)	ROC-3 < ROC-12 (3.45)	ROC-3 < ROC-12 (4.90)
6	URTL < MX-1 (3.82)	ROC-12 < ROC-3 (3.75)	LRTL < MA-10 (4.52)	MX-1 < MN-2 (3.05)	URTL < MX-1 (3.87)
7	MX-1 < URTL (3.81)	URTL < MX-1 (3.52)	URTL < MX-1 (4.33)	URTL < MX-1 (3.01)	URTL < MN-2 (3.12)
8	MN-1 < MX-2 (2.51)	MA-3 < MX-1 (3.34)	MX-1 < MN-2 (3.22)	MX-1 < MA-3 (2.92)	VOL<T-1> < VOL<T> (2.85)
9	ROC-12 < ROC-3 (2.49)	LRTL < MA-10 (3.26)	MA-3 < MX-1 (3.14)	MN-1 < MX-2 (2.22)	VOL<T> < VOL<T-1> (2.36)
10	MA-3 < MX-1 (2.19)	MN-1 < MX-2 (2.14)	LRTL < MX-1 (2.56)	LRTL < URTL (2.19)	MN-1 < MX-2 (1.99)
OTHERS	OTHERS (37.95)	OTHERS (38.13)	OTHERS (37.09)	OTHERS (42.96)	OTHERS (47.10)

Table 4-26: Level2 Weekly Table-Profit Driven Approach.

4.5.2.2 Risk-Adjusted Approach

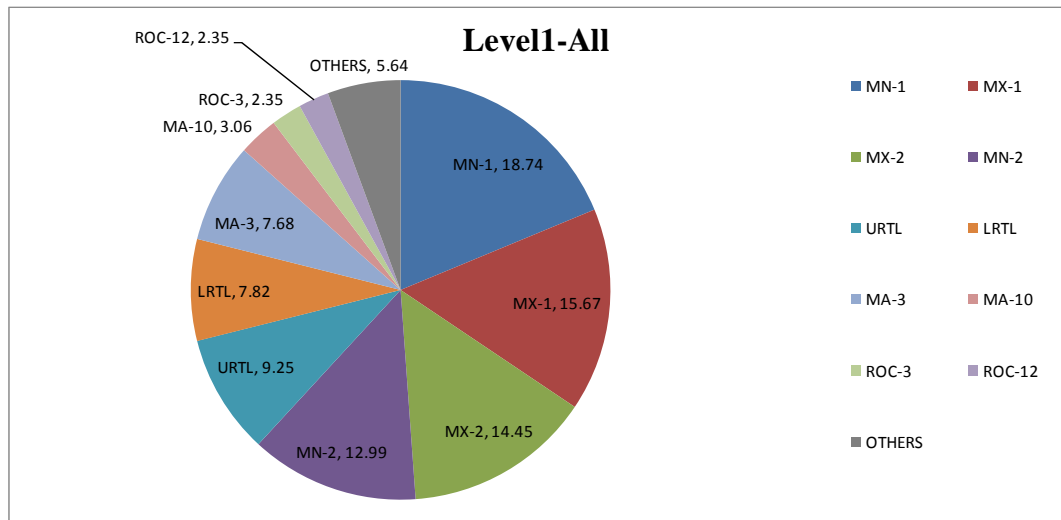


Figure 4-19: Level1 Weekly Proportion-Risk-Adjusted Approach.

Level1-1								
#	All	CMR CPC-LK12 MSTLRO	CMR CPC-LK12 SHARO	CMR CPC-LK24 MSTLRO	CMR CPC-LK24 SHARO	CMR MMDD	MR MMDD CXP	
1	MN-1 (18.74)	MN-1 (19.31)	MN-1 (18.50)	MN-1 (18.94)	MN-1 (18.46)	MN-1 (21.82)	MN-1 (19.28)	
2	MX-1 (15.67)	MX-1 (16.37)	MX-1 (17.12)	MX-1 (16.68)	MX-1 (17.31)	MX-2 (17.45)	MX-1 (15.95)	
3	MX-2 (14.45)	MN-2 (13.47)	MN-2 (15.28)	MX-2 (14.03)	MX-2 (14.37)	MX-1 (15.23)	MX-2 (15.76)	
4	MN-2 (12.99)	MX-2 (13.15)	MX-2 (14.69)	MN-2 (13.63)	MN-2 (13.66)	MN-2 (10.41)	MN-2 (11.88)	
5	URTL (9.25)	URTL (9.10)	URTL (8.80)	MA-3 (9.50)	URTL (9.93)	URTL (9.88)	URTL (9.19)	
6	LRTL (7.82)	MA-3 (8.78)	LRTL (8.08)	URTL (8.81)	LRTL (8.66)	MA-3 (8.03)	LRTL (8.57)	
7	MA-3 (7.68)	LRTL (8.85)	MA-3 (7.40)	LRTL (6.99)	MA-3 (7.89)	LRTL (7.70)	MA-3 (7.68)	
8	MA-10 (3.06)	MA-10 (3.15)	MA-10 (2.87)	MA-10 (2.87)	MA-10 (3.34)	MA-10 (3.38)	MA-10 (4.20)	
9	ROC-3 (2.35)	ROC-3 (2.36)	ROC-12 (1.32)	MA-6 (1.98)	ROC-3 (1.45)	MA-6 (1.60)	ROC-12 (1.75)	
10	ROC-12 (2.35)	ROC-12 (2.36)	ROC-3 (1.92)	ROC-12 (1.63)	ROC-3 (1.45)	ROC-3 (1.48)	ROC-3 (1.75)	
OTHERS	OTHERS (5.64)	OTHERS (5.08)	OTHERS (3.41)	OTHERS (4.97)	OTHERS (3.48)	OTHERS (3.03)	OTHERS (4.00)	
Level1-2								
#	MR PC-LK12 MSTLRO CXP	MR PC-LK12 SHARO CXP	MR PC-LK24 MSTLRO CXP	MR PC-LK24 SHARO CXP	MSTLRO CPC-LK12	MSTLRO CPC-LK24	MSTLRO CXP	
1	MN-1 (17.78)	MX-1 (17.67)	MN-1 (18.27)	MN-1 (17.33)	MN-1 (19.28)	MN-1 (18.47)	MN-1 (18.32)	
2	MX-1 (16.42)	MN-1 (16.39)	MX-1 (15.12)	MX-1 (17.07)	MX-1 (15.01)	MN-2 (13.85)	MX-2 (15.38)	
3	MX-2 (13.69)	MX-2 (14.67)	MN-2 (14.20)	MX-2 (14.62)	MX-2 (12.81)	MX-1 (13.56)	MX-1 (9.25)	
4	MN-2 (12.25)	MN-2 (13.45)	MX-2 (13.82)	MN-2 (13.52)	MN-2 (12.74)	MX-2 (12.70)	URTL (7.78)	
5	URTL (8.67)	URTL (9.66)	URTL (9.42)	URTL (9.60)	URTL (8.95)	URTL (10.05)	ROC-12 (7.20)	
6	MA-3 (8.05)	LRTL (8.75)	MA-3 (7.60)	LRTL (8.16)	MA-3 (8.58)	LRTL (8.04)	ROC-3 (7.20)	
7	LRTL (7.38)	MA-3 (7.10)	LRTL (6.39)	MA-3 (7.21)	LRTL (7.04)	MA-3 (7.85)	MN-2 (6.60)	
8	MA-10 (3.26)	MA-10 (2.99)	MA-10 (3.35)	MA-10 (3.78)	ROC-3 (3.02)	MA-6 (2.72)	MA-3 (5.63)	
9	ROC-3 (2.97)	ROC-12 (2.30)	ROC-12 (2.39)	ROC-3 (1.86)	ROC-12 (3.02)	MA-10 (2.58)	LRTL (4.74)	
10	ROC-12 (2.97)	ROC-3 (2.30)	ROC-3 (2.39)	ROC-12 (1.86)	MA-10 (2.70)	ROC-3 (2.51)	VOL-T-1> (3.96)	
OTHERS	OTHERS (6.57)	OTHERS (4.13)	OTHERS (7.06)	OTHERS (5.00)	OTHERS (6.86)	OTHERS (7.66)	OTHERS (13.64)	
Level1-3								
#	MSTLRO PC-LK12 CXP	MSTLRO PC-LK24 CXP	SHARO CPC-LK12	SHARO CPC-LK24	SHARO CXP	SHARO PC-LK12 CXP	SHARO PC-LK24 CXP	
1	MN-1 (19.62)	MN-1 (19.62)	MX-1 (17.84)	MN-1 (19.53)	MN-1 (22.88)	MN-1 (18.25)	MN-1 (18.24)	
2	MX-1 (15.12)	MX-1 (14.06)	MN-1 (17.05)	MX-1 (15.46)	MX-2 (19.71)	MX-1 (16.57)	MX-1 (16.15)	
3	MX-2 (13.25)	MN-2 (13.09)	MN-2 (14.42)	MN-1 (12.03)	MX-2 (15.06)	MX-2 (14.57)	MX-2 (14.81)	
4	MN-2 (12.28)	MX-2 (12.92)	MX-2 (13.90)	MN-2 (12.98)	MN-2 (10.59)	MN-2 (14.11)	MN-2 (14.05)	
5	URTL (8.83)	URTL (9.97)	URTL (9.02)	URTL (9.29)	URTL (8.55)	URTL (9.13)	URTL (9.71)	
6	MA-3 (8.24)	MA-3 (8.43)	LRTL (8.42)	LRTL (9.26)	LRTL (7.66)	LRTL (8.46)	LRTL (9.27)	
7	LRTL (7.41)	LRTL (7.56)	MA-3 (7.32)	MA-3 (7.25)	MA-3 (6.36)	MA-3 (7.38)	MA-3 (6.41)	
8	ROC-3 (3.01)	MA-10 (2.84)	MA-10 (3.47)	MA-10 (3.28)	ROC-12 (2.32)	MA-10 (2.99)	MA-10 (2.96)	
9	ROC-12 (3.01)	ROC-3 (2.47)	ROC-12 (2.39)	ROC-12 (1.59)	ROC-3 (2.32)	ROC-3 (1.90)	ROC-3 (1.91)	
10	MA-10 (2.78)	ROC-12 (2.47)	ROC-3 (2.39)	ROC-3 (1.59)	MA-10 (1.80)	ROC-12 (1.90)	ROC-12 (1.91)	
OTHERS	OTHERS (7.47)	OTHERS (6.58)	OTHERS (3.81)	OTHERS (4.73)	OTHERS (5.80)	OTHERS (4.76)	OTHERS (4.59)	

Table 4-27: Level1 Weekly Table- Risk-Adjusted Approach.

The technical indicators from the local minima and local maxima group re-appear as the most active in the risk-adjusted multi-objective approaches, as we see from Figure 4-19, accounting for 61.85% of leaves in rules, (59.70% in the profit driven approaches). The trend line indicators, LRTL and URTL, were involved in the trading rules at almost the same rate as in profit driven configuration, 17.07% for risk-adjusted configurations

versus 18.20% for profit driven approach. Moving to Level2, Figure 4-20 tells us that the most prominent expression was “MX-2 < MN-1” at 20.20%; this component was similarly prominent in the profit driven approaches at 16.01%. However, “MN-2 < MX-1”, the second most prominent expression, was a little less highlighted at 10.46% (12.84% in the profit driven approaches).

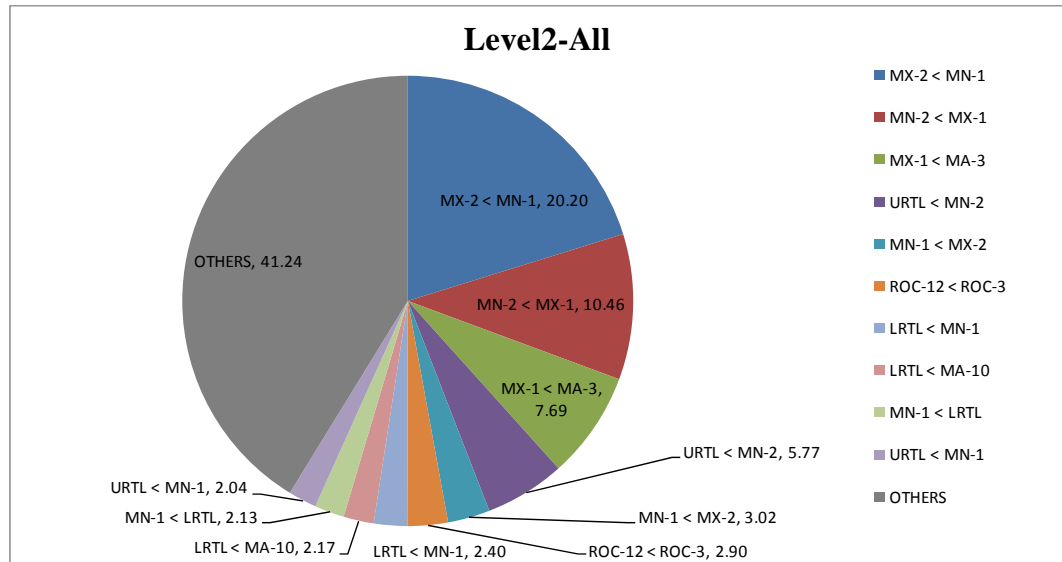


Figure 4-20: Level2 Weekly Proportion-Risk-Adjusted Approach.

Level2-1							
#	All	CMR CPC-LK12_MSTLRO	CMR CPC-LK12_SHARO	CMR CPC-LK24_MSTLRO	CMR CPC-LK24_SHARO	CMR MMDD	MR MMDD_CXP
1	MX2 < MN-1 (20.20)	MX2 < MN-1 (18.91)	MX2 < MN-1 (20.64)	MX2 < MN-1 (21.21)	MX2 < MN-1 (19.18)	MX2 < MN-1 (24.90)	MX2 < MN-1 (22.82)
2	MN-2 < MX-1 (10.46)	MN-2 < MX-1 (11.95)	MN-2 < MX-1 (12.74)	MN-2 < MX-1 (11.36)	MN-2 < MX-1 (10.97)	MN-2 < MX-1 (8.78)	MN-2 < MX-1 (10.44)
3	MX-1 < MA-3 (7.69)	MX-1 < MA-3 (9.07)	MX-1 < MA-3 (7.63)	MX-1 < MA-3 (9.72)	MX-1 < MA-3 (7.15)	MX-1 < MA-3 (7.61)	MX-1 < MA-3 (6.71)
4	URTL < MN-2 (5.77)	URTL < MN-2 (5.54)	URTL < MN-2 (7.36)	URTL < MN-2 (5.83)	URTL < MN-2 (7.01)	URTL < MN-2 (5.09)	URTL < MN-2 (4.92)
5	MN-1 < MX-2 (3.02)	ROC-12 < ROC-3 (3.37)	LRTL < MA-10 (3.15)	LRTL < MN-1 (2.92)	MN-1 < MX-2 (3.04)	MN-1 < MX-2 (3.67)	LRTL < MA-10 (3.48)
6	ROC-12 < ROC-3 (2.90)	MN-1 < MX-2 (3.34)	ROC-12 < ROC-3 (2.91)	MA-3 < MX-1 (2.46)	LRTL < MA-10 (2.74)	URTL < MN-1 (3.11)	MA-3 < MX-1 (3.02)
7	LRTL < MN-1 (2.40)	LRTL < MN-1 (2.87)	MX-1 < MN-2 (2.81)	MN-1 < MX-2 (2.16)	LRTL < MX-1 (2.66)	MA-3 < MX-1 (2.85)	LRTL < MX-1 (2.58)
8	LRTL < MA-10 (2.17)	MA-3 < MN-1 (2.39)	MN-1 < MX-2 (2.71)	ROC-12 < ROC-3 (2.06)	MA-3 < MX-1 (2.56)	LRTL < MA-10 (2.84)	MN-1 < LRTL (2.41)
9	MN-1 < LRTL (2.13)	MN-1 < LRTL (2.30)	MN-1 < LRTL (2.68)	MA-3 < MN-1 (2.05)	MN-1 < LRTL (2.45)	MN-1 < LRTL (2.68)	MN-1 < MX-2 (2.20)
10	URTL < MN-1 (2.04)	MA-3 < MX-1 (2.20)	LRTL < MN-1 (2.41)	MN-1 < LRTL (1.98)	LRTL < MN-1 (2.34)	LRTL < MX-1 (2.12)	ROC-12 < ROC-3 (2.06)
OTHERS	OTHERS (41.24)	OTHERS (38.06)	OTHERS (34.97)	OTHERS (38.24)	OTHERS (39.90)	OTHERS (36.34)	OTHERS (39.36)
Level2-2							
#	MR_PC-LK12_MSTLRO_CXP	MR_PC-LK12_SHARO_CXP	MR_PC-LK24_MSTLRO_CXP	MR_PC-LK24_SHARO_CXP	MSTLRO_CPC-LK12	MSTLRO_CPC-LK24	MSTLRO_CXP
1	MX2 < MN-1 (18.84)	MX2 < MN-1 (19.75)	MX2 < MN-1 (19.66)	MX2 < MN-1 (20.40)	MX2 < MN-1 (18.01)	MX2 < MN-1 (17.67)	MX2 < MN-1 (17.61)
2	MN-2 < MX-1 (10.39)	MN-2 < MX-1 (12.40)	MN-2 < MX-1 (10.95)	MN-2 < MX-1 (11.92)	MN-2 < MX-1 (10.20)	MN-2 < MX-1 (9.79)	MX-1 < MA-3 (7.71)
3	MX-1 < MA-3 (7.87)	MX-1 < MA-3 (7.51)	MX-1 < MA-3 (8.03)	MX-1 < MA-3 (6.82)	MX-1 < MA-3 (7.96)	MX-1 < MA-3 (6.73)	ROC-12 < ROC-3 (7.61)
4	URTL < MN-2 (4.48)	URTL < MN-2 (5.16)	URTL < MN-2 (6.89)	URTL < MN-2 (6.02)	URTL < MN-2 (5.10)	URTL < MN-2 (5.83)	ROC-3 < ROC-12 (6.79)
5	ROC-12 < ROC-3 (3.95)	ROC-12 < ROC-3 (3.10)	LRTL < MN-1 (2.82)	LRTL < MA-10 (3.47)	ROC-12 < ROC-3 (3.85)	LRTL < MN-1 (3.17)	MN-1 < MX-2 (4.93)
6	MN-1 < MX-2 (2.86)	MN-1 < MX-2 (2.73)	ROC-12 < ROC-3 (2.51)	LRTL < MX-1 (2.62)	MN-1 < MX-2 (3.45)	MN-1 < MX-2 (3.07)	URTL < MN-1 (4.70)
7	LRTL < MN-1 (2.52)	LRTL < MA-10 (2.71)	MN-1 < MX-2 (2.45)	ROC-12 < ROC-3 (2.55)	LRTL < MN-1 (3.20)	ROC-12 < ROC-3 (2.61)	VOL<T> < VOL<T-1> (3.98)
8	MA-3 < MX-1 (2.43)	MX-1 < URTL (2.66)	ROC-3 < ROC-12 (2.27)	MA-3 < MX-1 (2.49)	MA-3 < MN-1 (2.95)	ROC-3 < ROC-12 (2.41)	VOL<T-1> < VOL<T-2> (3.94)
9	LRTL < MX-1 (2.22)	URTL < MX-1 (2.36)	URTL < MN-1 (2.17)	MN-1 < MX-2 (2.45)	ROC-3 < ROC-12 (2.18)	MX-1 < MN-2 (1.92)	MN-2 < MX-1 (2.68)
10	URTL < MN-1 (2.16)	MA-3 < MX-1 (2.24)	MA-3 < MX-1 (2.01)	URTL < MX-1 (2.17)	URTL < MN-1 (2.04)	URTL < MA-10 (1.85)	URTL < MN-2 (2.29)
OTHERS	OTHERS (42.28)	OTHERS (39.37)	OTHERS (40.23)	OTHERS (39.10)	OTHERS (41.04)	OTHERS (44.94)	OTHERS (37.76)
Level2-3							
#	MSTLRO_PC-LK12_CXP	MSTLRO_PC-LK24_CXP	SHARO_CPC-LK12	SHARO_CPC-LK24	SHARO_CXP	SHARO_PC-LK12_CXP	SHARO_PC-LK24_CXP
1	MX2 < MN-1 (18.96)	MX2 < MN-1 (18.49)	MX2 < MN-1 (19.33)	MX2 < MN-1 (20.73)	MX2 < MN-1 (27.82)	MX2 < MN-1 (19.99)	MX2 < MN-1 (21.69)
2	MN-2 < MX-1 (10.37)	MN-2 < MX-1 (8.88)	MN-2 < MX-1 (12.47)	MN-2 < MX-1 (9.35)	MX-1 < MA-3 (8.77)	MN-2 < MX-1 (12.04)	MN-2 < MX-1 (11.01)
3	MX-1 < MA-3 (8.23)	MX-1 < MA-3 (7.63)	MX-1 < MA-3 (7.64)	MX-1 < MA-3 (7.34)	URTL < MN-2 (7.02)	MX-1 < MA-3 (7.24)	URTL < MN-2 (7.02)
4	URTL < MN-2 (4.82)	URTL < MN-2 (5.52)	URTL < MN-2 (6.37)	URTL < MN-2 (6.51)	MN-2 < MX-1 (6.00)	URTL < MN-2 (5.71)	MX-1 < MA-3 (6.74)
5	ROC-12 < ROC-3 (3.54)	MN-1 < MX-2 (3.38)	LRTL < MA-10 (3.36)	LRTL < MA-10 (3.36)	MN-1 < MX-2 (5.61)	MN-1 < MX-2 (3.31)	LRTL < MA-10 (3.23)
6	LRTL < MN-1 (2.79)	LRTL < MN-1 (3.14)	ROC-12 < ROC-3 (3.30)	MN-1 < MX-2 (2.96)	MN-1 < LRTL (4.14)	MX-1 < URTL (2.72)	LRTL < MN-1 (2.77)
7	ROC-3 < ROC-12 (2.46)	ROC-12 < ROC-3 (2.71)	MN-1 < MX-2 (2.85)	ROC-12 < MN-1 (2.73)	ROC-3 < ROC-12 (2.46)	ROC-12 < ROC-3 (2.67)	MX-1 < URTL (2.75)
8	URTL < MN-1 (2.38)	URTL < MN-1 (2.61)	MX-1 < MN-2 (2.64)	MN-1 < LRTL (2.44)	LRTL < MA-10 (2.42)	LRTL < MA-10 (2.57)	MN-1 < MX-2 (2.52)
9	MN-1 < MX-2 (2.34)	ROC-3 < ROC-12 (2.23)	MX-1 < URTL (2.59)	LRTL < MX-1 (2.38)	ROC-12 < ROC-3 (2.17)	MN-1 < LRTL (2.50)	LRTL < MX-1 (2.34)
10	MN-1 < LRTL (2.23)	MN-1 < LRTL (2.10)	MA-3 < MX-1 (2.33)	MX-1 < MN-2 (2.07)	URTL < MN-1 (2.17)	MX-1 < MN-2 (2.17)	ROC-12 < ROC-3 (2.33)
OTHERS	OTHERS (41.86)	OTHERS (43.32)	OTHERS (37.12)	OTHERS (40.14)	OTHERS (31.42)	OTHERS (39.09)	OTHERS (37.60)

Table 4-28: Level2 Weekly Table- Risk-Adjusted Approach.

4.5.3 Rule Analysis for Daily Trading

4.5.3.1 Profit Driven Approach

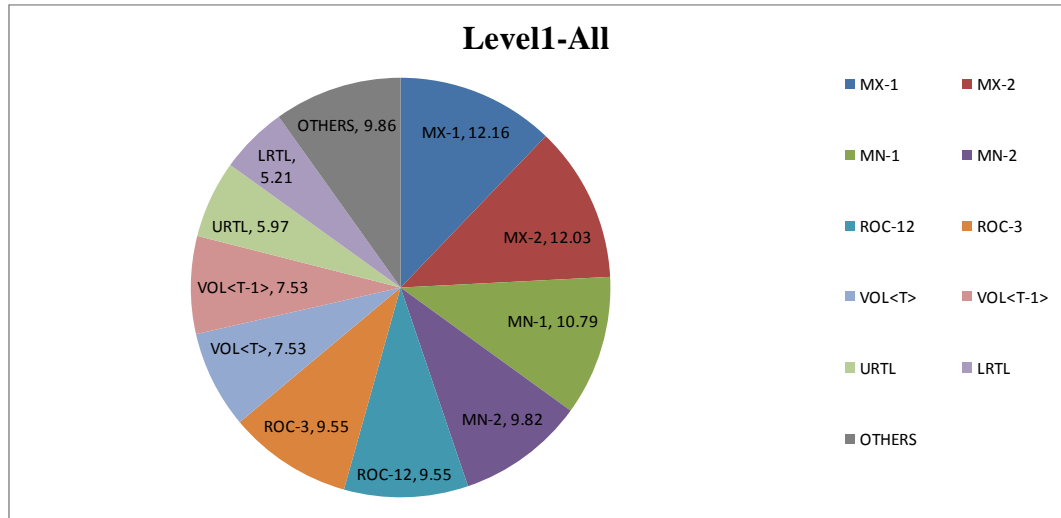


Figure 4-21: Level1 Daily Proportion-Profit Driven Approach.

Level1-1					
#	All	2CSMR	2SMR_CXP	CMR_CPC-LK12	CMR_CPC-LK24
1	MX1 (12.16)	MX2 (12.69)	MX2 (12.37)	MX2 (14.40)	MX1 (13.62)
2	MX2 (12.03)	MX1 (11.94)	MN-1 (11.66)	MX1 (13.26)	MX2 (11.88)
3	MN-1 (10.79)	MN-1 (11.76)	MX-1 (11.09)	MN-1 (11.62)	MN-1 (10.96)
4	MN-2 (9.82)	MN-2 (10.84)	MN-2 (10.35)	MN-2 (10.54)	MN-2 (10.16)
5	ROC-12 (9.55)	ROC-12 (8.08)	ROC-12 (8.66)	ROC-12 (8.48)	ROC-3 (9.06)
6	ROC-3 (9.55)	ROC-3 (8.08)	ROC-3 (8.66)	ROC-3 (8.48)	ROC-12 (9.06)
7	VOL<T> (7.53)	VOL<T> (6.34)	VOL<T-1> (6.45)	VOL<T> (6.60)	URTL (7.83)
8	VOL<T-1> (7.53)	VOL<T-1> (6.34)	VOL<T> (6.45)	VOL<T-1> (6.60)	VOL<T> (7.02)
9	URTL (5.97)	LRTL (6.30)	LRTL (5.99)	LRTL (5.88)	VOL<T-1> (7.02)
10	LRTL (5.21)	URTL (5.77)	URTL (5.43)	URTL (5.23)	LRTL (4.37)
OTHERS	OTHERS (9.86)	OTHERS (11.86)	OTHERS (12.91)	OTHERS (8.90)	OTHERS (9.03)
Level1-2					
#	MR_CXP	MR_PC-LK12_CXP	MR_PC-LK24_CXP	PC-LK12_CXP	PC-LK24_CXP
1	MX1 (12.66)	MX2 (14.60)	MX1 (14.50)	ROC-3 (12.04)	ROC-3 (14.49)
2	MX2 (12.48)	MX-1 (13.56)	MX-2 (12.10)	ROC-12 (12.04)	ROC-12 (14.49)
3	MN-1 (11.41)	MN-1 (11.63)	MN-1 (10.75)	VOL<T-1> (9.99)	VOL<T> (10.78)
4	MN-2 (10.43)	MN-2 (10.64)	MN-2 (9.76)	VOL<T> (9.99)	VOL<T-1> (10.78)
5	ROC-3 (8.82)	ROC-12 (8.29)	ROC-12 (8.99)	MX-2 (9.46)	MX-1 (9.44)
6	ROC-12 (8.82)	ROC-3 (8.29)	ROC-3 (8.99)	MX-1 (8.73)	MN-1 (8.05)
7	VOL<T> (7.93)	VOL<T-1> (6.38)	URTL (7.13)	MN-1 (8.68)	MX-2 (7.45)
8	VOL<T-1> (7.93)	VOL<T> (6.38)	VOL<T> (7.05)	MN-2 (8.44)	URTL (6.92)
9	LRTL (4.94)	URTL (5.93)	VOL<T-1> (7.05)	LRTL (6.32)	MN-2 (6.70)
10	URTL (4.56)	LRTL (5.25)	LRTL (4.58)	URTL (4.79)	LRTL (3.09)
OTHERS	OTHERS (10.00)	OTHERS (9.04)	OTHERS (9.09)	OTHERS (9.53)	OTHERS (7.83)

Table 4-29: Level1 Daily Table-Profit Driven Approach.

As we also saw in the case of single-objective daily trading, Figure 4-21 and Table 4-29 reveal that technical indicators in the trading rules for multi-objective trading were spread across various kinds of indicators, with no particular indicator dominating. The top six indicators from Figure 4-22 vary in proportion from 9.85% down to 7.34%. Together they account for around 51% of leaves in rules. Furthermore, the rate of change group of indicators constituted 19.1% of Level2 structures with “ROC-12 <

ROC-3” and “ROC-3 < ROC-12”, while the local minima and local maxima group made up 17.46% with “MX-2 < MN-1” and “MN-2 < MX-1”.

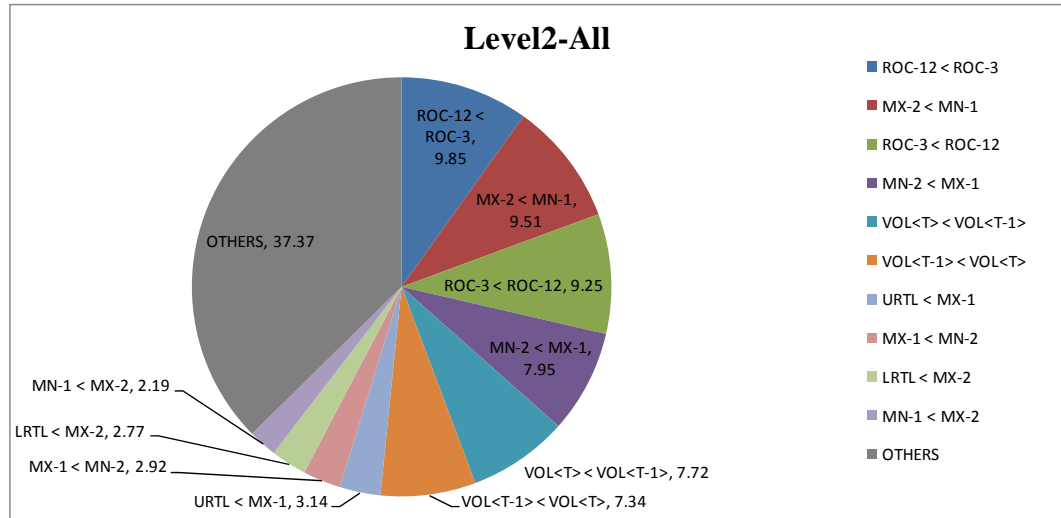


Figure 4-22: Level2 Daily Proportion-Profit Driven Approach.

Level2-1					
#	All	2CSMR	2SMR_CXP	CMR_CPC-LK12	CMR_CPC-LK24
1	ROC-12 < ROC-3 (9.85)	MX-2 < MN-1 (10.32)	MX-2 < MN-1 (10.07)	MX-2 < MN-1 (10.71)	MX-2 < MN-1 (10.34)
2	MX-2 < MN-1 (9.51)	ROC-3 < ROC-12 (8.84)	ROC-3 < ROC-12 (9.57)	ROC-12 < ROC-3 (9.09)	ROC-12 < ROC-3 (10.18)
3	ROC-3 < ROC-12 (9.25)	MN-2 < MX-1 (7.81)	MN-2 < MX-1 (7.90)	MN-2 < MX-1 (8.32)	ROC-3 < ROC-12 (7.94)
4	MN-2 < MX-1 (7.95)	ROC-12 < ROC-3 (7.32)	ROC-12 < ROC-3 (7.74)	ROC-3 < ROC-12 (7.88)	MN-2 < MX-1 (7.48)
5	VOL<T> < VOL<T-1> (7.72)	VOL<T> < VOL<T-1> (6.94)	VOL<T-1> < VOL<T> (6.65)	VOL<T> < VOL<T-1> (6.97)	VOL<T> < VOL<T-1> (7.47)
6	VOL<T-1> < VOL<T> (7.34)	VOL<T-1> < VOL<T> (5.75)	VOL<T> < VOL<T-1> (6.24)	VOL<T-1> < VOL<T> (6.23)	VOL<T-1> < VOL<T> (6.56)
7	URTL < MX-1 (3.14)	LRTL < MX-2 (3.21)	LRTL < MX-2 (2.83)	LRTL < MX-2 (4.04)	URTL < MX-1 (5.12)
8	MX-1 < MN-2 (2.92)	MN-1 < MX-2 (3.00)	MX-1 < MN-2 (2.75)	MX-1 < MN-2 (3.75)	MX-1 < MN-2 (3.24)
9	LRTL < MX-2 (2.77)	MX-1 < MN-2 (2.95)	MN-1 < MX-2 (2.39)	MN-1 < MX-1 (2.72)	URTL < MX-2 (2.51)
10	MN-1 < MX-2 (2.19)	URTL < MX-1 (2.45)	URTL < MX-1 (1.46)	URTL < MX-1 (2.71)	MN-1 < MX-1 (2.41)
OTHERS	OTHERS (37.37)	OTHERS (41.42)	OTHERS (42.38)	OTHERS (37.59)	OTHERS (36.76)

Level2-2					
#	MR_CXP	MR_PC-LK12_CXP	MR_PC-LK24_CXP	PC-LK12_CXP	PC-LK24_CXP
1	MX-2 < MN-1 (12.21)	MX-2 < MN-1 (11.39)	MX-2 < MN-1 (9.50)	ROC-12 < ROC-3 (13.49)	ROC-12 < ROC-3 (16.71)
2	MN-2 < MX-1 (10.31)	MN-2 < MX-1 (8.97)	ROC-12 < ROC-3 (9.41)	ROC-3 < ROC-12 (10.58)	ROC-3 < ROC-12 (12.26)
3	ROC-3 < ROC-12 (10.23)	ROC-12 < ROC-3 (8.64)	ROC-3 < ROC-12 (8.58)	VOL<T> < VOL<T-1> (10.06)	VOL<T> < VOL<T-1> (11.53)
4	VOL<T-1> < VOL<T> (8.35)	ROC-3 < ROC-12 (7.93)	MN-2 < MX-1 (8.38)	VOL<T-1> < VOL<T> (9.92)	VOL<T-1> < VOL<T> (10.03)
5	VOL<T> < VOL<T-1> (7.52)	VOL<T> < VOL<T-1> (6.41)	VOL<T> < VOL<T-1> (7.21)	MN-2 < MX-1 (7.72)	URTL < MX-1 (5.47)
6	ROC-12 < ROC-3 (7.42)	VOL<T-1> < VOL<T> (6.35)	VOL<T-1> < VOL<T> (6.89)	MX-2 < MN-1 (5.35)	MX-2 < MN-1 (4.75)
7	MX-1 < MN-2 (2.99)	MX-1 < MN-2 (3.92)	URTL < MX-1 (4.99)	LRTL < MX-2 (4.51)	MN-2 < MX-1 (4.28)
8	LRTL < MX-2 (2.38)	LRTL < MX-2 (3.52)	MN-1 < MX-1 (2.80)	MA-3 < MN-1 (2.42)	MN-1 < MX-2 (1.78)
9	MN-1 < MX-1 (2.31)	URTL < MX-1 (3.35)	MX-1 < MN-2 (2.80)	MN-1 < MX-2 (2.16)	MX-1 < MN-2 (1.70)
10	URTL < MX-1 (2.12)	URTL < MX-2 (2.65)	URTL < MX-2 (2.66)	MX-1 < MN-2 (1.97)	MN-2 < MX-2 (1.20)
OTHERS	OTHERS (34.16)	OTHERS (36.86)	OTHERS (36.79)	OTHERS (31.82)	OTHERS (30.30)

Table 4-30: Level2 Daily Table-Profit Driven Approach.

4.5.3.2 Risk-Adjusted Approach

Looking at the structures found in rules for the risk-adjusted multi-objective approaches in the daily trading environment, we again find the diversity that we have found before in the daily trading context. In this case, the top six technical indicators cover 67% of the whole proportion (see Figure 4-23), and in Figure 4-24 we see that the local minima and local maxima group, “MX-2 < MN-1” and “MN-2 < MX-1”, account together for 21.3% of Level2 structures, whereas the rate of change group, “ROC-12 < ROC-3” and “ROC-3 < ROC-12”, accounted for 15.85%.

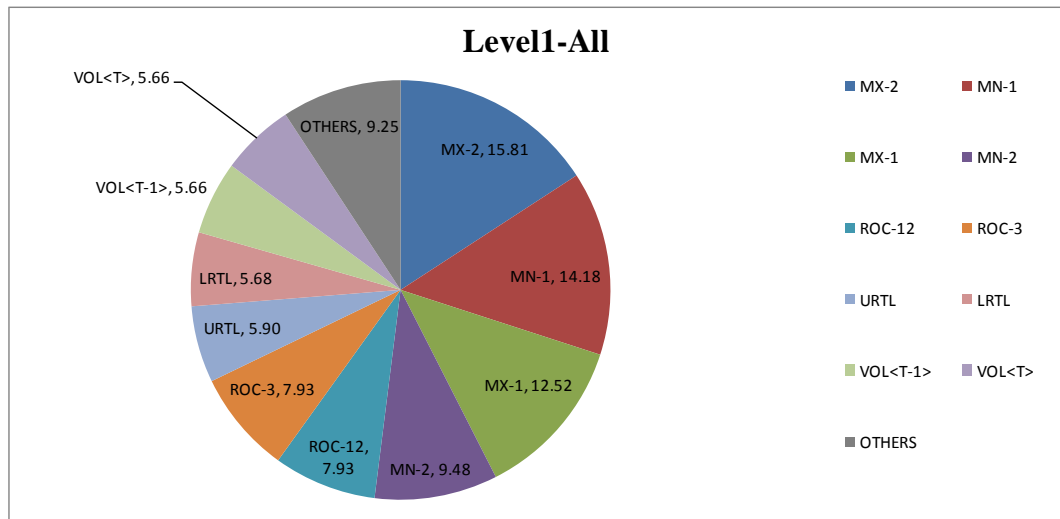


Figure 4-23: Level1 Daily Proportion-Risk-Adjusted Approach.

Level1-1							
#	All	CMR_CPC-LK12_MSTLRO	CMR_CPC-LK12_SHARO	CMR_CPC-LK24_MSTLRO	CMR_CPC-LK24_SHARO	CMR_MMDD	MR_MMDD_CXP
1	MX-2 (15.81)	MX-2 (19.14)	MX-2 (15.10)	MX-2 (17.63)	MX-1 (14.53)	MX-2 (14.89)	MX-2 (13.29)
2	MN-1 (14.18)	MN-1 (17.16)	MX-1 (14.35)	MN-1 (15.25)	MX-2 (13.18)	MX-1 (13.46)	MN-1 (13.19)
3	MX-1 (12.52)	MX-1 (12.54)	MN-1 (13.70)	MX-1 (12.97)	MN-1 (11.97)	MN-1 (13.17)	MX-1 (11.71)
4	MN-2 (9.48)	MN-2 (9.60)	MN-2 (10.38)	MN-2 (8.96)	MN-2 (9.99)	MN-2 (9.78)	MN-2 (9.62)
5	ROC-12 (7.93)	ROC-3 (6.27)	ROC-12 (8.07)	ROC-3 (7.24)	ROC-3 (8.62)	ROC-3 (7.92)	ROC-3 (9.31)
6	ROC-3 (7.93)	ROC-12 (6.27)	ROC-3 (8.07)	ROC-12 (7.24)	ROC-12 (8.62)	ROC-12 (7.92)	ROC-12 (9.31)
7	URTL (5.90)	LRTL (6.38)	LRTL (5.66)	URTL (6.77)	URTL (6.59)	LRTL (6.38)	VOL<T> (7.06)
8	LRTL (5.68)	URTL (5.72)	URTL (5.51)	LRTL (6.57)	VOL<T> (6.51)	VOL<T-1> (5.88)	VOL<T-1> (7.06)
9	VOL<T-1> (5.66)	VOL<T> (3.70)	VOL<T-1> (4.97)	VOL<T-1> (4.06)	VOL<T-1> (6.51)	VOL<T> (5.88)	LRTL (5.34)
10	VOL<T> (5.66)	VOL<T-1> (3.70)	VOL<T> (4.97)	VOL<T> (4.06)	LRTL (4.45)	URTL (4.09)	URTL (3.62)
OTHERS	OTHERS (9.25)	OTHERS (9.81)	OTHERS (9.22)	OTHERS (9.25)	OTHERS (9.03)	OTHERS (10.93)	OTHERS (10.49)
Level1-2							
#	MR_PC-LK12_MSTLRO_CXP	MR_PC-LK12_SHARO_CXP	MR_PC-LK24_MSTLRO_CXP	MR_PC-LK24_SHARO_CXP	MSTLRO_CPC-LK12	MSTLRO_CPC-LK24	MSTLRO_CXP
1	MX-2 (18.18)	MX-2 (15.10)	MX-2 (17.03)	MX-1 (13.87)	MX-2 (19.68)	MX-2 (16.17)	MX-2 (13.39)
2	MN-1 (16.69)	MX-1 (13.59)	MN-1 (15.41)	MX-2 (12.14)	MN-1 (16.98)	MN-1 (15.44)	MN-1 (17.44)
3	MX-1 (13.27)	MN-1 (13.43)	MX-1 (13.27)	MN-1 (11.15)	MX-1 (12.43)	MX-1 (11.65)	ROC-12 (8.90)
4	MN-2 (10.37)	MN-2 (11.30)	MN-2 (9.83)	MN-2 (9.50)	MN-2 (9.99)	MN-2 (8.95)	ROC-3 (8.90)
5	URTL (7.07)	ROC-12 (7.19)	ROC-12 (6.84)	ROC-12 (8.80)	LRTL (7.10)	ROC-3 (7.89)	ROC-3 (8.23)
6	LRTL (6.10)	ROC-12 (7.19)	ROC-3 (6.84)	ROC-3 (8.80)	ROC-12 (6.68)	ROC-12 (7.89)	MN-2 (7.08)
7	ROC-3 (6.01)	URTL (6.18)	URTL (6.41)	URTL (7.57)	ROC-3 (6.68)	URTL (6.40)	VOL<T> (7.04)
8	ROC-12 (6.01)	LRTL (5.54)	LRTL (6.34)	VOL<T> (6.88)	URTL (5.37)	LRTL (6.30)	VOL<T-1> (7.04)
9	VOL<T> (3.64)	VOL<T-1> (5.42)	VOL<T-1> (3.97)	VOL<T-1> (6.88)	VOL<T> (3.78)	VOL<T-1> (5.53)	URTL (4.41)
10	VOL<T-1> (3.64)	VOL<T> (5.42)	VOL<T> (3.97)	LRTL (4.99)	VOL<T-1> (3.78)	VOL<T> (5.53)	URTL (3.61)
OTHERS	OTHERS (9.02)	OTHERS (9.63)	OTHERS (10.06)	OTHERS (9.40)	OTHERS (7.53)	OTHERS (8.30)	OTHERS (9.80)
Level1-3							
#	MSTLRO_PC-LK12_CXP	MSTLRO_PC-LK24_CXP	SHARO_CPC-LK12	SHARO_CPC-LK24	SHARO_CXP	SHARO_PC-LK12_CXP	SHARO_PC-LK24_CXP
1	MX-2 (19.55)	MX-2 (14.48)	MX-2 (13.34)	MX-2 (13.20)	MX-2 (13.20)	MX-2 (13.20)	MX-2 (13.39)
2	MN-1 (18.25)	MN-1 (16.12)	MX-1 (12.32)	MX-1 (12.33)	MN-1 (12.52)	MX-1 (12.11)	MX-1 (13.28)
3	MX-1 (11.67)	MX-1 (12.46)	MN-1 (12.23)	MN-1 (11.06)	VOL<T> (9.34)	MN-1 (12.10)	MN-1 (10.42)
4	MN-2 (9.54)	MN-2 (9.05)	MN-2 (9.52)	ROC-12 (9.87)	VOL<T-1> (9.34)	MN-2 (9.63)	ROC-3 (9.35)
5	ROC-3 (6.75)	ROC-12 (7.22)	ROC-12 (8.48)	ROC-3 (8.87)	ROC-12 (9.29)	ROC-3 (8.57)	ROC-12 (9.35)
6	ROC-12 (6.75)	ROC-3 (7.22)	ROC-3 (8.48)	MN-2 (8.62)	ROC-3 (9.29)	ROC-12 (8.57)	MN-2 (9.07)
7	LRTL (6.55)	URTL (6.74)	VOL<T-1> (6.65)	VOL<T> (7.78)	MX-1 (9.03)	VOL<T> (6.75)	VOL<T> (7.62)
8	URTL (5.55)	LRTL (6.68)	VOL<T> (6.65)	VOL<T-1> (7.78)	MN-2 (8.21)	VOL<T-1> (6.75)	VOL<T-1> (7.62)
9	VOL<T-1> (3.51)	VOL<T> (4.17)	LRTL (6.04)	URTL (6.70)	LRTL (5.64)	URTL (5.43)	URTL (6.60)
10	VOL<T> (3.51)	VOL<T-1> (4.17)	URTL (5.67)	URTL (4.05)	URTL (4.43)	URTL (5.37)	LRTL (4.63)
OTHERS	OTHERS (8.36)	OTHERS (8.34)	OTHERS (9.47)	OTHERS (8.60)	OTHERS (9.70)	OTHERS (9.84)	OTHERS (8.67)

Table 4-31: Level1 Daily Table- Risk-Adjusted Approach.

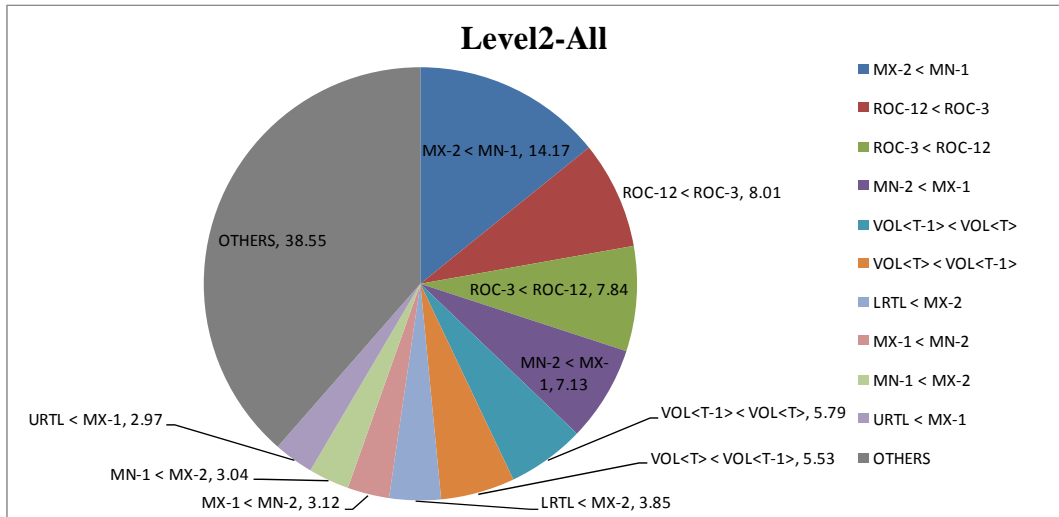


Figure 4-24: Level2 Daily Proportion-Risk-Adjusted Approach.

Level2-1							
#	All	CMR CPC-LK12 MSTLRO	CMR CPC-LK12 SHARO	CMR CPC-LK24 MSTLRO	CMR CPC-LK24 SHARO	CMR MMDD	MR MMDD CXP
1	MX-2 < MN-1 (14.17)	MX-2 < MN-1 (17.79)	MX-2 < MN-1 (11.72)	MX-2 < MN-1 (15.05)	MX-2 < MN-1 (11.46)	MX-2 < MN-1 (13.99)	MX-2 < MN-1 (12.11)
2	ROC-12 < ROC-3 (8.01)	MN-2 < MX-1 (7.52)	MN-2 < MX-1 (8.74)	ROC-12 < ROC-3 (7.78)	ROC-12 < ROC-3 (9.57)	ROC-3 < ROC-12 (9.38)	ROC-3 < ROC-12 (10.14)
3	ROC-3 < ROC-12 (7.84)	ROC-3 < ROC-12 (6.42)	ROC-3 < ROC-12 (8.26)	ROC-3 < ROC-12 (6.71)	MN-2 < MX-1 (8.26)	MN-2 < MX-1 (7.99)	ROC-12 < ROC-3 (8.48)
4	MN-2 < MX-1 (7.13)	ROC-12 < ROC-3 (6.12)	ROC-12 < ROC-3 (7.88)	MN-2 < MX-1 (6.35)	ROC-3 < ROC-12 (7.67)	ROC-12 < ROC-3 (6.45)	VOL<T-1> < VOL<T> (7.24)
5	VOL<T-1> < VOL<T> (5.79)	LRTL < MX-2 (4.35)	VOL<T-1> < VOL<T> (5.21)	LRTL < MX-2 (4.52)	VOL<T-1> < VOL<T> (6.54)	VOL<T-1> < VOL<T> (5.89)	VOL<T-1> < VOL<T> (6.88)
6	VOL<T> < VOL<T-1> (5.53)	VOL<T-1> < VOL<T> (3.89)	VOL<T-1> < VOL<T> (4.73)	URTL < MX-1 (4.52)	VOL<T-1> < VOL<T> (6.49)	VOL<T-1> < VOL<T> (6.87)	MN-2 < MX-1 (6.74)
7	LRTL < MX-2 (3.85)	VOL<T-1> < VOL<T> (3.52)	MN-1 < MX-1 (4.21)	VOL<T-1> < VOL<T> (4.31)	URTL < MX-1 (4.07)	MX-1 < MN-2 (4.69)	MX-1 < MN-2 (4.08)
8	MX-1 < MN-2 (3.12)	MN-1 < MX-2 (3.23)	LRTL < MX-2 (3.88)	MN-1 < MX-2 (4.16)	MN-1 < MX-1 (3.14)	LRTL < MX-2 (3.71)	MN-1 < MX-2 (3.52)
9	MN-1 < MX-2 (3.04)	MX-1 < MN-2 (3.04)	MX-1 < MN-2 (3.32)	VOL<T-1> < VOL<T> (3.80)	MX-1 < MN-2 (2.95)	MN-1 < MX-2 (3.30)	LRTL < MX-2 (3.50)
10	URTL < MX-1 (2.97)	MN-1 < MX-1 (2.55)	MN-1 < MX-2 (2.98)	MN-1 < MX-1 (3.13)	LRTL < MX-2 (2.53)	URTL < MX-1 (2.04)	MN-1 < MX-1 (2.11)
OTHERS	OTHERS (38.55)	OTHERS (41.58)	OTHERS (39.08)	OTHERS (39.67)	OTHERS (37.33)	OTHERS (36.63)	OTHERS (35.19)
Level2-2							
#	MR PC-LK12 MSTLRO CXP	MR PC-LK12 SHARO CXP	MR PC-LK24 MSTLRO CXP	MR PC-LK24 SHARO CXP	MSTLRO CPC-LK12	MSTLRO CPC-LK24	MSTLRO CXP
1	MX-2 < MN-1 (16.60)	MX-2 < MN-1 (11.52)	MX-2 < MN-1 (15.36)	MX-2 < MN-1 (10.08)	MX-2 < MN-1 (17.91)	MX-2 < MN-1 (15.57)	MX-2 < MN-1 (20.98)
2	MN-2 < MX-1 (7.92)	MN-2 < MX-1 (9.50)	ROC-12 < ROC-3 (7.34)	ROC-12 < ROC-3 (9.25)	MN-2 < MX-1 (8.29)	ROC-3 < ROC-12 (7.94)	ROC-3 < ROC-12 (9.94)
3	ROC-3 < ROC-12 (6.09)	ROC-12 < ROC-3 (7.25)	MN-2 < MX-1 (6.82)	ROC-12 < ROC-3 (8.36)	ROC-12 < ROC-3 (6.90)	ROC-12 < ROC-3 (7.85)	ROC-12 < ROC-3 (7.85)
4	ROC-12 < ROC-3 (5.92)	ROC-3 < ROC-12 (7.14)	ROC-3 < ROC-12 (6.35)	MN-2 < MX-1 (7.16)	ROC-3 < ROC-12 (6.46)	MN-2 < MX-1 (6.21)	VOL<T-1> < VOL<T> (7.81)
5	LRTL < MX-2 (4.96)	VOL<T-1> < VOL<T> (5.56)	LRTL < MX-2 (4.89)	VOL<T-1> < VOL<T> (7.09)	LRTL < MX-2 (6.05)	VOL<T-1> < VOL<T> (5.56)	VOL<T-1> < VOL<T> (6.28)
6	VOL<T-1> < VOL<T> (3.72)	VOL<T-1> < VOL<T> (5.29)	VOL<T-1> < VOL<T> (4.21)	VOL<T-1> < VOL<T> (6.68)	MN-1 < MX-2 (4.57)	VOL<T-1> < VOL<T> (5.49)	MN-1 < MX-2 (4.71)
7	MN-1 < MX-2 (3.71)	LRTL < MX-2 (4.49)	VOL<T-1> < VOL<T> (3.74)	URTL < MX-1 (4.39)	VOL<T-1> < VOL<T> (3.89)	LRTL < MX-2 (3.92)	MX-1 < MN-2 (3.23)
8	MX-1 < MN-2 (3.66)	MN-1 < MX-1 (3.27)	URTL < MX-1 (3.62)	MX-1 < MN-2 (3.05)	VOL<T-1> < VOL<T> (3.66)	MN-1 < MX-2 (3.64)	MN-2 < MX-1 (2.81)
9	VOL<T> < VOL<T-1> (3.55)	MX-1 < MN-2 (3.25)	MN-1 < MX-1 (3.32)	MN-1 < MX-2 (2.77)	MX-1 < MN-2 (3.57)	URTL < MX-1 (3.15)	MX-1 < MA-3 (1.97)
10	MN-1 < MX-1 (3.35)	URTL < MX-1 (2.69)	MX-1 < MN-2 (3.13)	URTL < MX-2 (2.50)	MN-1 < MX-1 (2.57)	MX-1 < MN-2 (3.02)	MX-2 < MX-2 (1.69)
OTHERS	OTHERS (40.51)	OTHERS (40.04)	OTHERS (41.23)	OTHERS (38.67)	OTHERS (36.12)	OTHERS (37.65)	OTHERS (32.72)
Level2-3							
#	MSTLRO PC-LK12 CXP	MSTLRO PC-LK24 CXP	SHARO CPC-LK12	SHARO CPC-LK24	SHARO CXP	SHARO PC-LK12 CXP	SHARO PC-LK24 CXP
1	MX-2 < MN-1 (18.77)	MX-2 < MN-1 (17.83)	MX-2 < MN-1 (11.77)	MX-2 < MN-1 (11.23)	MX-2 < MN-1 (12.09)	MX-2 < MN-1 (11.49)	MX-2 < MN-1 (10.93)
2	ROC-12 < ROC-3 (6.85)	ROC-12 < ROC-3 (8.19)	ROC-12 < ROC-3 (9.19)	ROC-12 < ROC-3 (10.44)	ROC-3 < ROC-12 (10.29)	ROC-12 < ROC-3 (8.80)	ROC-12 < ROC-3 (9.98)
3	MN-2 < MX-1 (6.85)	MN-2 < MX-1 (6.70)	ROC-3 < ROC-12 (7.77)	ROC-3 < ROC-12 (9.30)	VOL<T-1> < VOL<T> (9.69)	MN-2 < MX-1 (8.57)	ROC-3 < ROC-12 (8.73)
4	ROC-3 < ROC-12 (6.66)	ROC-3 < ROC-12 (6.25)	MN-2 < MX-1 (7.38)	VOL<T-1> < VOL<T> (8.07)	VOL<T-1> < VOL<T> (8.99)	ROC-3 < ROC-12 (8.33)	VOL<T-1> < VOL<T> (7.81)
5	LRTL < MX-2 (5.66)	LRTL < MX-2 (5.35)	VOL<T-1> < VOL<T> (6.89)	VOL<T-1> < VOL<T> (7.49)	ROC-12 < ROC-3 (8.29)	VOL<T-1> < VOL<T> (6.81)	VOL<T-1> < VOL<T> (7.42)
6	MN-1 < MX-2 (4.38)	VOL<T-1> < VOL<T> (4.33)	VOL<T-1> < VOL<T> (6.41)	MN-2 < MX-1 (5.98)	MN-2 < MX-1 (4.61)	VOL<T-1> < VOL<T> (6.68)	MN-2 < MX-1 (7.07)
7	VOL<T-1> < VOL<T> (3.66)	URTL < MX-1 (4.10)	LRTL < MX-2 (3.83)	URTL < MX-1 (4.57)	MN-1 < MX-2 (3.45)	LRTL < MX-2 (4.66)	URTL < MX-1 (4.27)
8	MX-1 < MN-2 (3.58)	VOL<T-1> < VOL<T> (4.01)	MX-1 < MN-2 (2.71)	MX-1 < MN-2 (3.21)	LRTL < MX-2 (3.18)	URTL < MX-1 (2.29)	MX-1 < MN-2 (2.72)
9	VOL<T> < VOL<T-1> (3.36)	MN-1 < MX-2 (3.56)	URTL < MX-1 (2.56)	URTL < MX-2 (2.68)	URTL < MX-1 (2.23)	MN-1 < MX-1 (2.20)	URTL < MX-2 (2.72)
10	MN-1 < MX-1 (3.00)	MX-1 < MN-2 (2.67)	URTL < MX-2 (2.37)	MX-1 < MX-2 (1.96)	MN-1 < MX-1 (2.00)	MX-1 < MN-2 (2.18)	MX-1 < MX-2 (2.14)
OTHERS	OTHERS (37.25)	OTHERS (37.01)	OTHERS (39.11)	OTHERS (35.06)	OTHERS (35.16)	OTHERS (37.99)	OTHERS (36.20)

Table 4-32: Level2 Daily Table- Risk-Adjusted Approach.

4.6 Concluding Discussion

In this chapter we have investigated the use of several configurations of multi-objective GP algorithms for developing trading rules for trading on the stock market, specifically in association with the S&P 500 portfolio of companies. This continues a thread of research that dates back to Allen and Karjalainen's work [2] which showed that GP could produce profitable rules, and has since been supplemented by many efforts, increasingly aimed at understanding how to configure GP to provide robust and convincing performance (in terms of outperforming buy and hold and risk-free return strategies). Building on earlier work in the previous chapter that indicated robust and convincing performance could be achieved in the context of monthly trading, but less so for weekly trading (and rarely for daily trading), we show that multi-objective approaches, especially when using a majority voting approach, are able to generally outperform single-objective approaches, as well as robustly outperform buy and hold and risk-free return. This is the case for monthly trading, and even more clearly so for weekly trading, in which improving on the performance shown for single-objective approaches in [60] and in Chapter three, we see good evidence that multi-objective GP for evolving trading rules seems viable in the context of weekly trading. These results are to some extent underpinned in their robustness by the fact that they emerge after considering 8 different data splits that cover a range of different trading environments.

Similarly, we find that, in the monthly and weekly scenarios, the approach in which trading was done via majority vote over a set of non-dominated rules was more effective than the approach in which performance was effectively averaged over the rules in the set. Again, these observations were reversed in the case of daily trading, which we speculate is due to the increased volatility in that scenario, confounding any clear measures of risk. However it could also be that our approach to evolving rules for daily

trading was compromised by forcing the rules to reason over only short-timescale moving average and similar indicators.

In comparing each of nine different characterizations of the problem into multiple objectives using profit-driven approaches, we find that the worst performance (again, considering only the monthly and weekly cases) came from the configurations within the ‘basic group’, which comprised the most straightforward multi-objective versions of our single-objective approaches. This was also true for the risk-adjusted approaches in both the monthly and weekly environments. It is quite obvious that methods in either the profit-driven basic group or the risk basic group could not deliver effective results compared with single-objective methods. Having said that, when the corresponding single objective methods are themselves part of a multi-objective configuration, such as MR-PC_LK24-CXP (3 objectives), this seems to result in a robust and effective configuration. Comparing profit-driven and risk-adjusted approaches, the results showed that considering profit only can perform better in both monthly and weekly trading, while integrating risk resulted in improvement in the more challenging case of daily trading.

In our analyses to learn the ingredients of evolved trading rules in both single-objective and multi-objective approaches, we discovered two facts in common between those two approaches. First, we found that technical indicators from the group of local minima and maxima indicators dominated other indicators in all cases of monthly and weekly trading, however in weekly trading the trend-line group was also highly active. When we considered depth-2 subtrees, the most active expressions were “MN-2 < MX-1” and “MX-2 < MN-1” in both monthly and weekly trading rules. Finally, in rules generated for daily trading, technical indicators were found from various groups with no indicators being particularly prominent.

5. A New Approach for Grammatical Evolution

5.1 Introduction

Grammatical Evolution (GE) [77] is a recent member in the family of Evolutionary Algorithms. It uses binary genomes and production rules in Backus Naur Form (BNF) format to map from genotype to phenotype, i.e. a key element of GE is the *genotype-to-phenotype mapping process*. This feature allows GE to generate programs in any language by using the binary strings to select BNF production rules defined by users. This brings about freedom to GE users and makes GE itself highly mutable, involving a mapping layer to the GE system and thereby allowing users to change the grammar independently without needing to change other aspects of the GE system. The mapping layer transforms a binary genome to a phenotype based on the underlying user-defined grammar. Given its promising and interesting aspects, we decided to study the GE algorithm and aim to apply it to solve trading problems as we did with GP in [60, 61] and in Chapters 3 and 4. Our first attempt to understand the GE algorithm was done by testing the GE system (JAVA version) downloaded from [41, 67]. However, performance of this system was poor when compared with the results on the same problems mentioned in [77]. Thereafter, our approach was to replicate the GE algorithm originally proposed in [72, 73, 77, 86] with help of JAVA open source code from [42, 70]. We then tested our replicated GE with problems mentioned in [77], but the results indicated low performance for some problems compared with those in [77]. We explain the differences by noting the incomplete specifications of the GE algorithm in some of the cited publications, especially in terms of the configurations used to generate certain results and, in general, the details of the genome mapping process used. Following on from this experience, we developed our own approaches to GE and the work in this chapter represents our new GE approach and provides a comprehensive evaluation of it,

compared with the results of GP, for each of several standard test problems. Subsequently, we have also carried out experiments for a more challenging problem in the real world – financial trading.

The following describes the structure of the remaining sections in this chapter.

- Section 5.2 defines three problem domains, which are used to evaluate the performance of algorithms, for both GE and GP.
- Section 5.3 provides details of our GE Methodology, including overview concept, fitness calculation, mapping process and initialization process.
- Section 5.4 presents concepts concerning repairing invalid genomes, which sometimes result from an incomplete mapping process, via three different strategies: Valid Map Method1 (MAP01), Valid Map Method2 (MAP02) and Valid Map Method3 (MAP03), and section 5.5 shows the results of using GE to solve six test problems defined in section 5.2 for each of the valid map methods, and the results of using GP to solve the same set of problems, being used as a baseline comparison.
- Section 5.6 gives details of standard genetic operators for GE: one-point crossover, two-point crossover and flip mutation, while section 5.7 proposes the new approach of genetic operators for GE: typed genetic operators – one typed crossover operator and six typed mutation operators. To see the performance of the new genetic operators, comprehensive tests on six problems from section 5.2 has been done, and the results are reported in section 5.8.
- In section 5.9 we test our GE approach on the more challenging problem of financial trading. We compare GP and our new GE approach (with new valid map method and typed genetic operators) on each of monthly, weekly and daily data.

- Section 5.10 is the summary section for this chapter giving an overview summary, and some discussion on interesting issues and potential future developments.

5.2 Test Problems for Comparing GE and GP

In this section we state and describe three problem domains that we will use to test the performance of GE, in comparison with GP. These domains are: symbolic regression problem, symbolic integration problem and the Santa Fe ant trail. A description of each problem is as follows.

5.2.1 Symbolic Regression (SR)

Providing a given set of input and output pairs (data points) in the form (*input*, *output*), this problem aims to find the target mathematical function that generates the outputs from the corresponding inputs in each pair. To calculate a fitness value for a candidate solution to this problem, it is calculated by finding the error between the evolved and target functions for each pair of 20 data points and summing those errors as a fitness value. In total, we have five different target functions mixed with various types of components; they are $x^4 + x^3 + x^2 + x$ (SR1), $\sin(x) + x + x^2$ (SR2), $\log x + e^x + x^2 + x$ (SR3), $\sin(x) + e^x + x^3 + x^2 + x$ (SR4) and $\sin(x) + x^4 + x^3 + x^2 + x$ (SR5).

A BNF grammar used by GE for the symbolic regression problem and also for symbolic integration is defined in Figure 5-1.

$N = \{ \langle \text{Expr} \rangle, \langle \text{Op} \rangle, \langle \text{Pre-op} \rangle \}$ $T = \{ \text{Sin}, \text{Cos}, \text{Exp}, \text{Log}, +, -, *, /, X \}$ $S = \langle \text{Expr} \rangle$			
(A)	$\langle \text{Expr} \rangle ::=$	$\langle \text{Expr} \rangle \langle \text{Op} \rangle \langle \text{Expr} \rangle$ $\langle \text{Pre-op} \rangle (\langle \text{Expr} \rangle)$ $\langle \text{Var} \rangle$	(A.0) (A.1) (A.2)
(B)	$\langle \text{Op} \rangle ::=$	$+$ $-$ $*$ $/$	(B.0) (B.1) (B.2) (B.3)
(C)	$\langle \text{Pre-op} \rangle ::=$	Sin Cos Exp Log	(C.0) (C.1) (C.2) (C.3)
(D)	$\langle \text{Var} \rangle ::=$	X	(D.0)

Figure 5-1: BNF Grammar for both symbolic regression problem and symbolic integration problem

Configurations for symbolic regression problems are given in Table 5-1 and Table 5-2 for GE and GP respectively.

Symbolic Regression Configuration for GE	
Objective	Find a target function that fits a given 20 data points in the form of (x_i, y_i) within interval $[-1, 1]$ for SR1, interval $[0, 2\pi]$ for SR2, SR4 and SR5 and interval $(0, 2]$ for SR3.
Terminal Operands	The independent variable: x
Terminal Operators	The arithmetic operators: $+$, $-$, $*$, $/$ The unary operators: Sin , Cos , Exp , Log
Fitness Function	The sum of the error over 20 data points
Parameters	Model: GE Steady State Population Size: 500 Generation Gap: 0.5 Generation Size: 50 Mutation Probability.: 0.01 Crossover Probability: 0.9

Table 5-1: Symbolic regression configuration for GE

Symbolic Regression Configuration for GP	
Objective	Find a target function that fits a given 20 data points in the form of (x_i, y_i) within interval $[-1, 1]$ for SR1, interval $[0, 2\pi]$ for SR2, SR4 and SR5 and interval $(0, 2]$ for SR3.
GP Terminal Set	The independent variable: x
GP Function Set	+, -, *, /, Sin, Cos, Exp, Log
Fitness Function	The sum of the error over 20 data points
Parameters	Model: GP Steady State Population Size: 500 Generation Gap: 0.5 Generation Size: 50 Mutation Probability.: 0.01 Crossover Probability: 0.9

Table 5-2: Symbolic regression configuration for GP

5.2.2 Symbolic Integration (SI)

The goal of this problem is to find a function that is the integral of the given curve represented by a given set of data points, and this type of problem can be reduced to symbolic regression. The same symbolic integration function as in [77] is used for our experiments, which is

$$f(x) = \cos(x) + 2x + 1$$

eq. 5-1

And, the target integral curve for the above function is

$$f(x) = \sin(x) + x + x^2$$

eq. 5-2

Similarly to the approach in Koza [54], this particular symbolic integration problem can be reduced to the symbolic regression problem of finding the target function on the

target integral curve (eq. 5-2) instead and this is the same SR2 problem as in the previous section. The system is fed with a set of 20 input and output pairs in the form (x_i, y_i) , and the fitness function is calculated in the same way. Please refer to the previous section for more detail.

5.2.3 Santa Fe Ant Trail (ANT)

The objective of the Santa Fe ant trail, a standard test problem for GP, is to obtain a computer program to control an artificial ant to discover 89 pieces of food on a 32×32 toroidal grid, and the ant has limited energy (615 energy units) to search in the grid. Moreover, the ant spends one energy unit on each of the following actions: turn left, turn right, move one position forward; it may also look ahead one block in the direction it is facing to determine whether that block contains food or not without losing energy. The fitness measure for this problem is simply the number of food pieces picked up by the artificial ant. A BNF grammar used by GE for the Santa Fe ant trail is defined in Figure 5-2.

N = { <Code>, <Line>, <If-Else>, <Op> }		
T = { TurnLeft(), TurnRight(), Move(), FoodAhead() }		
S = <Code>		
(A) <Code> ::=	<Line>	(A.0)
	<Code> <Line>	(A.1)
(B) <Line> ::=	If (FoodAhead()) {<Line>} Else {<Line>}	(B.0)
	<Op>	(B.1)
(C) <Op> ::=	TurnLeft()	(C.0)
	TurnRight()	(C.1)
	Move()	(C.2)

Figure 5-2: BNF Grammar for Santa Fe ant Trail.

Configurations for Santa Fe ant trial are given in Table 5-3 and Table 5-4 for GE and GP respectively.

Santa Fe Ant Configuration for GE	
Objective	Find a computer program to control an artificial ant to search all 89 pieces of food spread through out a 32x32 toroidal grid.
Terminal Operands	None
Terminal Operators	TurnLeft(), TurnRight(), Move(), FoodAhead()
Fitness Function	Number of food pieces picked up by the artificial ant before it run out of energy.
Parameters	Model: GE Steady State Population Size: 500 Generation Gap: 0.5 Generation Size: 50 Mutation Probability.: 0.01 Crossover Probability: 0.9

Table 5-3: Santa Fe ant configuration for GE

Santa Fe Ant Configuration for GP	
Objective	Find a computer program to control an artificial ant to search all 89 pieces of food spread through out a 32x32 toroidal grid.
GP Terminal Set	TurnLeft(), TurnRight(), Move(), FoodAhead()
GP Function Set	<If-Else>, <Code>, <Line>
Fitness Function	Number of food pieces picked up by the artificial ant before it run out of energy.
Parameters	Model: GP Steady State Population Size: 500 Generation Gap: 0.5 Generation Size: 50 Mutation Probability.: 0.01 Crossover Probability: 0.9

Table 5-4: Santa Fe ant configuration for GP

Summary notes on all above problems are provided in Table 5-5. Please notice that problem #1, #2 and #3 are the same standard problems tested in [77].

Problem	Full Name	Abbr.	Function
#1	Santa Fe Ant	ANT	Computer program to control an artificial ant to find 89 pieces of foods on a 32x32 toroidal grid
#2	Symbolic Regression1	SR1	$x^4 + x^3 + x^2 + x$
#3	Symbolic Regression2 (Symbolic Integration)	SR2	$\sin(x) + x^2 + x$
#4	Symbolic Regression3	SR3	$\log x + e^x + x^2 + x$
#5	Symbolic Regression4	SR4	$\sin(x) + e^x + x^3 + x^2 + x$
#6	Symbolic Regression5	SR5	$\sin(x) + x^4 + x^3 + x^2 + x$

Table 5-5: Summary table for six testing problems.

5.3 GE Methodology

5.3.1 Overview

GE is an evolutionary algorithm (EA) that can evolve complete programs by using a genotype-to-phenotype mapping process in which the genotype is a variable-length binary or integer string. GE uses the Backus Naur Form (BNF) grammar in its mapping process (the translation phase), and it uses the binary or integer genome to select production rules from the grammar, eventually to derive, from the start symbol, a complete program which contains only terminal symbols at the end of the mapping process. The overall process corresponds somewhat to that in the biological process of

generating a protein from the combination of genetic material present in the DNA of an organism as shown in Figure 5-3 [77].

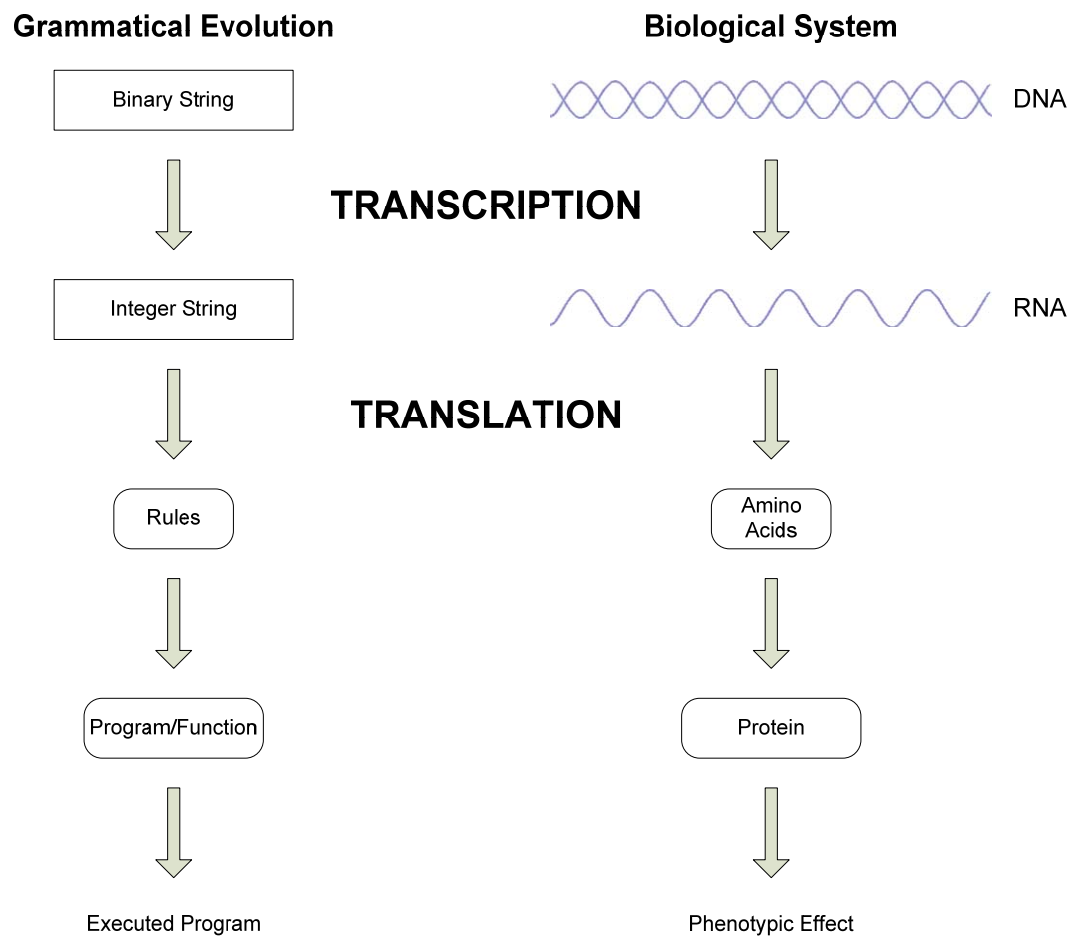


Figure 5-3: A mapping between the grammatical evolution (GE) system and a biological genetic system.

Using a grammar in the GE mapping process can give benefits in two ways. First, one need not worry about the semantics of the phenotype, as its valid semantics are assured and encoded already in the grammar. In addition, this also brings about independence in choosing the languages to build the complete program as one can use any valid grammar to map from genotype to phenotype. Last, the simple binary or integer genome enables the use of standard GA genetic operators.

5.3.2 GE Fitness Calculation and Mapping Process

In Figure 5-4 is shown the GE fitness calculation process. The mapping process turns the integer genome into a function tree, via the selection and application of production rules from the grammar. It is important to realise that this can often lead to an invalid tree, e.g. perhaps because it does not all of the function nodes have terminals assigned. To begin with, if the genome hasn't been mapped yet, then we start the mapping process. Eventually, a valid genome is evaluated and the fitness value is returned as a result. In the case of an invalid genome, and if we are using a 'Valid Map' method, the invalid genome is repaired by a specific 'valid map method', which guarantees the validity of the mapping result, and then proceeds to next step of evaluation process. In the case we are not using a valid map method, a default fitness value based on the problem is used, (the lowest possible fitness value for a maximisation problem, or highest possible fitness value for a minimisation problem). The GE mapping process is illustrated by an example we present next. One can find complete detail in [72, 73, 77, 86] and one can also get the GE java open source code from [42, 70]. Moreover, there are GE open source codes available for downloading from Bangor with Java implementation [49] and from the BDS Group at the University of Limerick with C++ implementation [5].

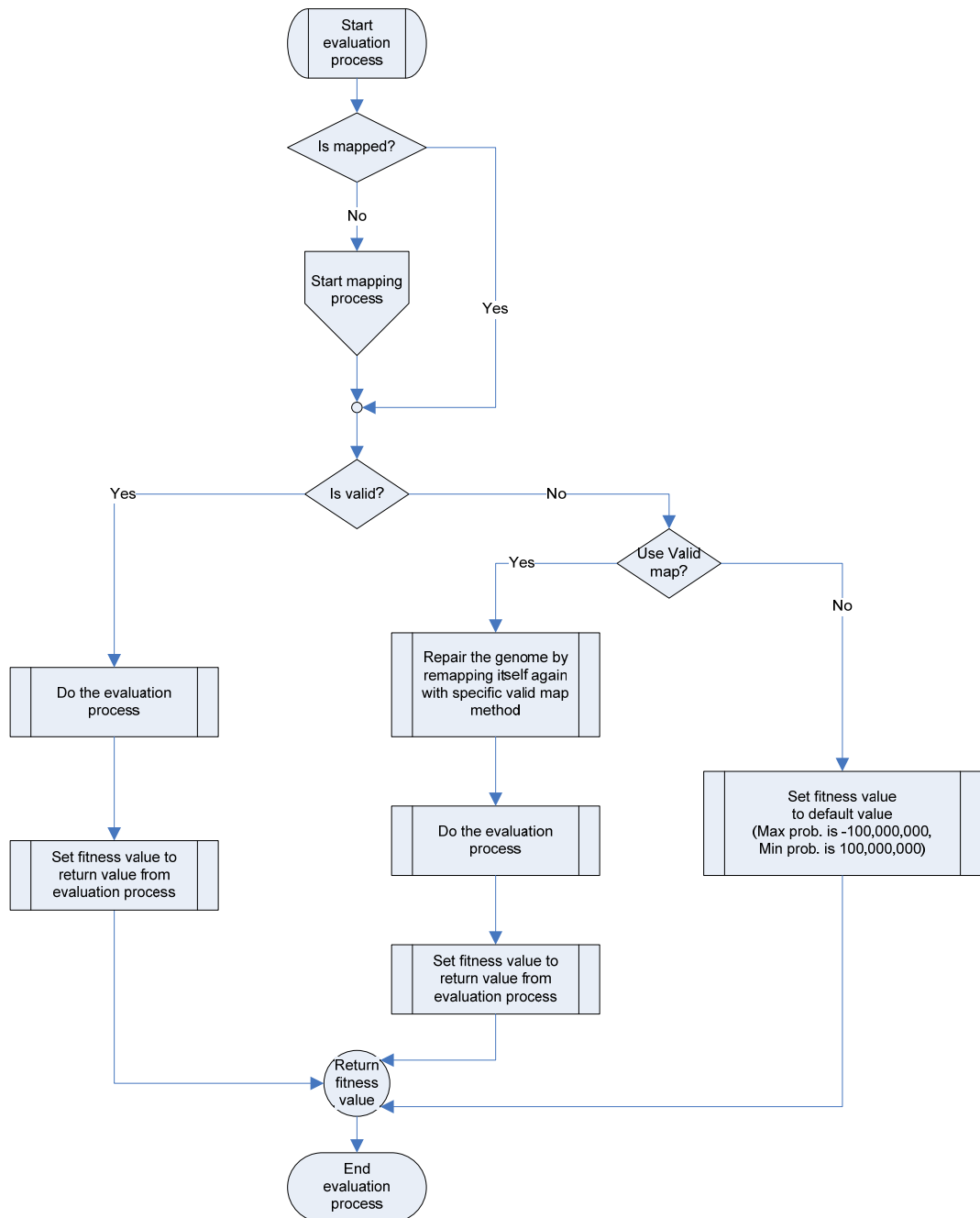


Figure 5-4: GE fitness calculation

Table 5-6 summarizes the grammar rules and the number of choices associated with each one of the grammar in Figure 5-1.

Grammar Rule Name	Number of Choices
A	3
B	4
C	4
D	1

Table 5-6: The number of available choices for each grammar rule.

In GE, the gene values tend to be called *codons*. Assuming that we have the genome in Figure 5-5, we will use these codons below to illustrate the mapping procedure.

7	8	2	5	2	4
0	1	2	3	4	5

Figure 5-5: An example of codons used to map the rules.

The mapping process begins with the start symbol which is $\langle \text{Expr} \rangle$ in this case. By looking up the grammar the grammar rule for this symbol is rule (A). Considering rule (A), we have 3 choices of production rules that can act (see Table 5-6), and we will decide which of these to use by using the following mapping function:

$$\text{Rule} = (\text{Codon integer value}) \text{ MOD } (\text{Number of production rules for the current non-terminal})$$

where MOD stands for the modulo operator. The first codon is 7, then

$$7 \text{ Mod } 3 = 1$$

gives rise to selecting production rule (A.1) $\langle \text{Expr} \rangle ::= \langle \text{Pre-op} \rangle (\langle \text{Expr} \rangle)$. As a result, $\langle \text{Expr} \rangle$ is now replaced as

$$\langle \text{Pre-op} \rangle (\langle \text{Expr} \rangle)$$

By always starting at the left hand side of the expression, we then get the first non-terminal symbol $\langle \text{Pre-op} \rangle$. We now continue the mapping process with the second codon, which is 8; so we have $8 \text{ Mod } 4 = 0$, meaning we have to take the first production rule (C.0) which is $\langle \text{Pre-op} \rangle := \text{Sin}$. Therefore, the expression will be

$$\text{Sin} (\langle \text{Expr} \rangle)$$

After that, we continue with the third codon with the non-terminal symbol $\langle \text{Expr} \rangle$, and this time we get $2 \bmod 3 = 2$ which leads us to select rule (A.2). In consequence, the expression will become

$$\text{Sin} (\langle \text{Var} \rangle)$$

Afterwards, please note that the non-terminal $\langle \text{Var} \rangle$ has only one production rule (D.0) which means there is no need to use the mapping function to select the rule. For this reason, the final expression will be

$$\text{Sin} (\langle \text{X} \rangle)$$

Hence, beginning from the left hand side of the expression, codon integer values are generated from a binary string and used to select production rules from the BNF grammar, until one of the following situations arise:

1. A complete program is generated. In this case all the non-terminal symbols in the expression can be mapped into terminal symbols, so the expression only consists of terminal symbols.
2. The end of the binary (integer) string is reached, which means we have run out of codons. If a wrapping feature is in use, then we are going to reuse these codons by returning to the beginning. Reuse may continue a number of times, repeatedly reusing the codons a number of times. We specify maximum amount of reuse (a maximum amount of ‘wrapping’). Hence, reuse will continue if necessary, until this maximum is reached.
3. When we reach the end of the string (and/or maximum reuse), and the individual is still incompletely mapped, we have two options:
 - a. The mapping process is halted, and the individual is assigned the worst possible fitness value.
 - b. Use a repair strategy to fix this invalid genome [110].

5.3.3 GE Initialization

The last issue that we would like to have a short discussion about is the issue of initialization methods. Normally a GE system generates the initial population in a random manner (random strings), where individuals, more specifically binary (integer) strings, are generated randomly bit by bit; in consequence. This means there is no guarantee that the individuals are able to map completely and become valid genomes. On the other hand, GP usually uses the ramped half-and-half initialization method [54], the most common initialization method, combining full and grow initialization methods to create a range of tree genomes in a diversity of shapes and depths. In [77] can be found a sensible initialization method for GE which is analogous to the ramped half-and-half method in the sense that both methods intend to make the initial population as general and diverse as possible. Also, in this chapter we have proposed an initialization method for GE which is analogous to the main method used in GP, and we simply call it the ramped half-and-half method for GE.

5.4 Repair Strategies

When a genome can only be incompletely mapped, we can sort this problem out by repairing the genome with any of various strategies, which we call ‘valid map’ methods. We describe and experiment with three alternative valid map methods. Any invalid genomes arising from incomplete mapping will be repaired by using one of these valid map methods. Otherwise, if repair is not in use, we will assign the fitness values of the invalid genomes to default values depending on the problem.

Let us suppose that we have the codons in figure 5.6 for the current genome, and these produce the invalid genome (incomplete mapping) with a maximum number of allowed wrappings set to 5 (`MAX_NO_OF_WRAPPING = 5`).

2	1	3	5	2	1
0	1	2	3	4	5

codon_length = 6

Figure 5-6: An example of codons generated invalid genome.

If we assume that the wrapping count (how many times that wrapping has been done) of 3 was used in the mapping process and the last codon index was 3 (zero-based index), then the unwrapped codons will be:

2	1	3	5	2	1	2	1	3	5	2	1	2	1	3	5	2	1	2	1	3
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

codon_length = 21

Figure 5-7: Unwrapped codons after reusing codons in Figure 5-6 3 times

We will repair the genome using these unwrapped codons in the next step of the valid mapping process. The details for each of our three valid map methods, when applied in the situation above, are given next as we explain the three valid map methods.

5.4.1 Valid Map Method1 (MAP01)

This method uses unwrapped codons in Figure 5-7 directly without modifying them.

5.4.2 Valid Map Method2 (MAP02)

We begin with randomly selecting the start index (`start_index`) from the range of $[0, \text{codon_length} - 1]$ on the unwrapped codons, and then randomly pick the number of codons (`cut_count`) from the range of $[1, \text{codon_length} - \text{start_index}]$ to cut from that start index. Suppose that the random `start_index` is 8 and `cut_count` is 3, then the codons used in the valid mapping process will be as in figure below.

2	1	3	5	2	1	2	1	3	5	2	1	2	1	3	5	2	1	2	1	3
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

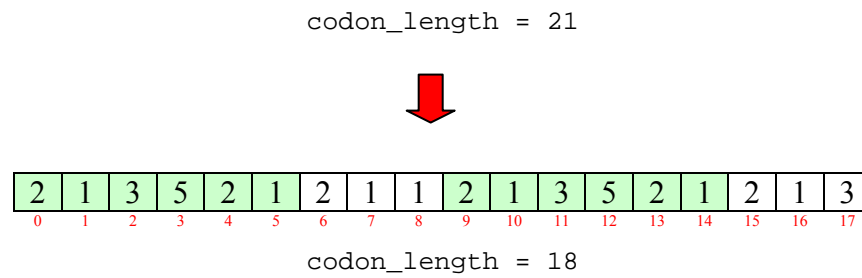


Figure 5-8: Unwrapped codons after fixing with valid map method2

5.4.3 Valid Map Method3 (MAP03)

This method works in the same way as MAP02, but after we have the random start index we cut all codons from that point to the end. Suppose that random `start_index` is 8, and then the codons used in the valid mapping process will be as in Figure 5-9.

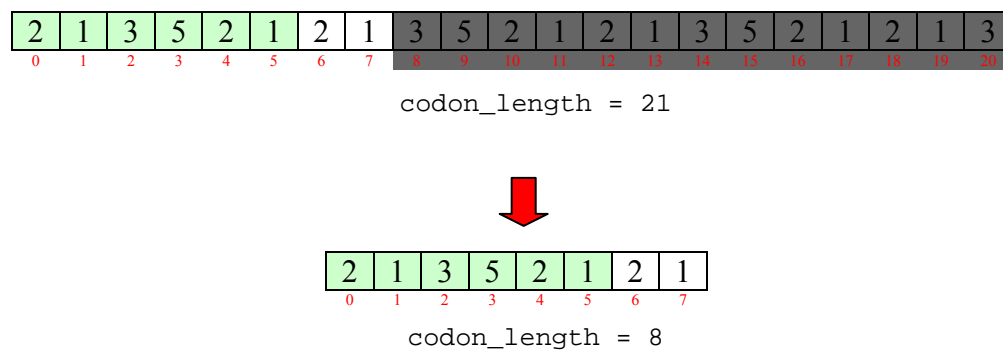


Figure 5-9: Unwrapped codons after fixing with valid map method3

After we've got the codons by using any of the valid map methods above, the next step is using them in the mapping process. In order to generate a valid genome, when we create the list of possible production rules for mapping the derivation trees with the selected production rule from that list, we select only the production rules which will not make the current tree exceed the desired depth. However, it is possible for the genome to run out of codons (remember that we cannot use wrapping with the unwrapped codons here), and in this case we randomly select the production rule from the possible list of production rules and add the index for that rule at the end of the codons, so in this circumstance the final list of codons will be longer than the original

one. In the next section we present the results of using GE to solve problems defined in section 5.2, for each of the three valid map methods.

5.5 Performance Comparison of Valid Map Methods

Table 5-7 summarises the results of both GP and GE on solving six problems, measuring performance as the success rate of finding the solution over 100 runs; each row provides the success rates for each of six problems for a particular configuration, beginning with standard GP, standard GE, and then GE with valid map methods. Please note that rows 1.2 and 2.3 are the results of GP and GE respectively reported in O'Neill and Ryan's work [77], which is available for only the first three problems: ANT, SR1 and SR2. To make a general summary of comparative performance easier, each row of Table 5-7 is compared with the results of standard GP as a baseline (row 1.1, highlighted in green), and a blue highlight in a cell indicates that the performance in the highlighted cell was better or equal to standard GP in the corresponding column.

#	Method	ANT	SR1	SR2	SR3	SR4	SR5
1	GP - Standard						
1.1	GP (Standard)	0.38	0.46	1.00	0.87	0.22	0.13
1.2	GP (O'Neill and Ryan's Work)	0.15	0.99	0.37	N/A	N/A	N/A
2	GE - Standard						
2.1	GE (Standard)	0.89	0.15	0.22	0.27	0.06	0.03
2.2	GE (Ramped-half-and-half Initialization)	0.73	0.14	0.86	0.49	0.29	0.08
2.3	GE (O'Neill and Ryan's Work)	0.90	0.98	0.80	N/A	N/A	N/A
3	GE - With Valid Map Method						
3.1	Random Initialization (INIT1)						
3.1.1.1	GE-MAP01-INIT1	0.89	0.50	0.43	0.25	0.09	0.03
3.1.1.2	GE-MAP02-INIT1	0.91	0.47	0.41	0.31	0.13	0.07
3.1.1.3	GE-MAP03-INIT1	0.94	0.41	0.44	0.42	0.13	0.06
3.2	Ramped-half-and-half Initialization (INIT2)						
3.2.1.1	GE-MAP01-INIT2	0.78	0.41	0.94	0.63	0.21	0.06
3.2.1.2	GE-MAP02-INIT2	0.81	0.28	0.92	0.70	0.13	0.10
3.2.1.3	GE-MAP03-INIT2	0.81	0.37	0.94	0.68	0.18	0.07

Table 5-7: Results of standard GP, standard GE and GE with valid map methods on six problems, showing success rate of finding the solution over 100 runs (e.g. 0.87 means that in 87 out of 100 runs it succeeded to find the solution).

It should be noted that the results of our GP implementation and the GP system in O'Neill and Ryan's work [77] were quite different, since we do not know the precise details of the GP implementation used in their study, and we were not able to set our GP configuration to achieve closely similar results. Comparing O'Neill and Ryan's GP and our GP implementations, we found that our GP got better results on the Santa Fe ant

problem and perfect results (100% of success rate) on symbolic integration (SR2), while the success rate of O’Neill and Ryan’s GP on symbolic regression – SR1 was considerably higher than that of our GP. In the case of GE, our ‘standard GE’ is our attempt to replicate the original GE by using the descriptions in [11, 70, 72, 73, 77, 86] and using available open source code from [5, 42, 49] as the guideline for our implementation. However, when we test our standard GE (row 2.1), its results were deeply disappointing on SR1 and SR2, but satisfactory on the ANT problem. Moving to comparison of GE with GP: standard GE performed impressively on the Santa Fe ant problem with a success rate of 89%, but showed relatively poor performance on the remaining test problems.

These results of standard GE provided constant surprises for us and provided the inspiration to improve GE performance with the idea of valid map methods and typed genetic operators. Let us consider the effect of a valid map repair method in use with GE. The results of using the valid map methods, with either of the two initialization methods, reveal improvements over our standard GE implementation. When using the random initialization method, the best valid map method seems to be MAP03, and for ramped-half-and-half initialization, MAP03 was still able to keep consistent performance, with significant improvement on problems SR2 and SR3. As a result of this, we have decided to choose MAP03 as our standard valid map repair method for the remaining experiments reported in this chapter. However it should be noted that even with valid map methods we could not bring about noticeable improvement in GE on all six test problems; it was not able to outperform standard GP especially on SR2-SR5.

5.6 Standard GE Genetic Operators

As we have already mentioned above, standard GE uses standard genetic operators for binary or integer strings. The following subsections give details of GE's standard one-point crossover, two-point crossover and flip mutation.

5.6.1 One-point Crossover Operator

The single crossover point, r , is randomly chosen in the range of $[1, l - 1]$ (l is length of the binary string) excluding before the first position ($r = 0$) and after the last position ($r = l$) for each parent, and then create the two children by exchanging the tails of both parents (see Figure 5-10).

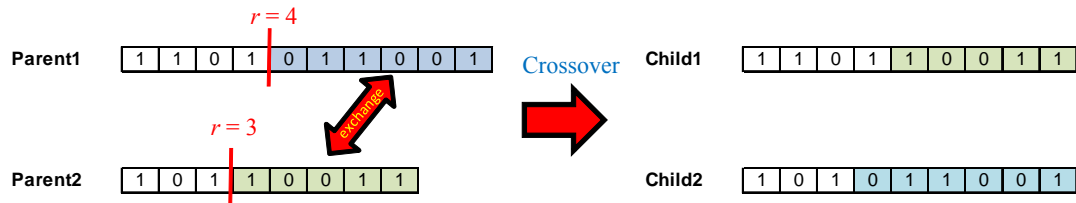


Figure 5-10: One-point crossover.

5.6.2 Two-point Crossover Operator

Both parents are separated into three segments of contiguous genes for each parent by randomly choosing two crossover point, r_1 and r_2 . Following that, two children are built by taking head and tail segments from one parent and middle segments from the other (see Figure 5-11).

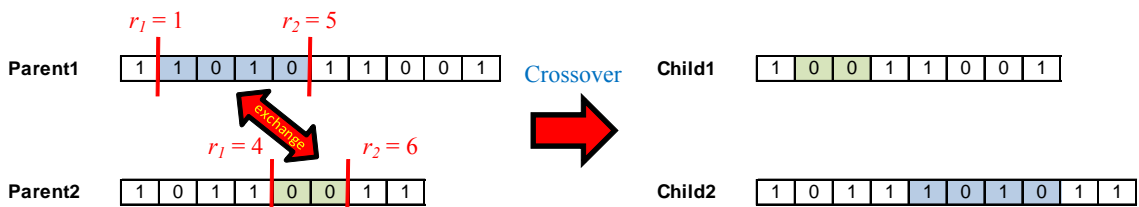


Figure 5-11: Two-point crossover.

5.6.3 Flip Mutation Operator

Each bit is considered separately to flip (i.e., from 1 to 0 or 0 to 1) with a small probability p_m . In Figure 5-12 is illustrated an example of a bit string, where the third, sixth, eighth and ninth bits are flipped.

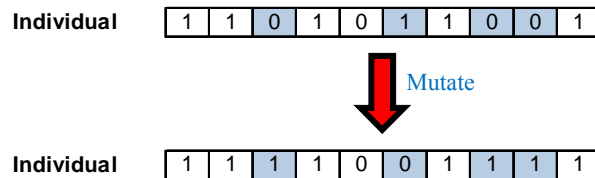


Figure 5-12: Flip mutation.

5.7 Typed Genetic Operators

Because the genome in standard GE is a binary string, it is natural to use the same standard crossover operators (one-point crossover/two-point crossover) and standard mutation operator (flip mutation) of the genetic algorithm (GA). By default, standard GE employs a standard one-point crossover operator inspired by GAs, and its behaviour, which is an inherent property of GE, was termed *ripple crossover* (first described in [51]), in which the tree is left with a *spine* and several *ripple sites* from which one or more sub-trees, dubbed *ripple trees*, are removed. (Please see [71] for more details). GE has been criticized for this seemingly destructive crossover operator, which is common with GP systems. However, in [75, 76] the behaviour of GE's one-point crossover is considered in a positive manner due to the *intrinsic polymorphism* [51] inherent in the way it distributes codons, since they can be used in any part of the grammar. In addition, studies in [84] have indicated that a high locality within representations is necessary for efficient evolutionary search, and the results of [84] revealed that the mapping used in GE has low locality, leading to poor performance when using standard mutation operators. Previously, there was an attempt [74] to

propose a new form of GE operator called homologous crossover, inspired by the molecular biological crossover process, and other approaches to design novel homologous crossover operators were taken in [39, 56], motivated by reducing the phenomenon of *bloat*. Nevertheless, the results of these attempts indicated that they seemed no better than the original standard GE one-point crossover, derived from GAs.

Moving to consider the process of standard GE crossover and mutation, crossover sites of two parents in the standard crossover process are arbitrarily selected, and the bits to flip in flip mutation are also arbitrarily assigned based on the mutation probability. In the GE mapping context, choosing crossover points and mutation points randomly can give rise to more invalid offspring. Consequently, our attempts to improve GE centred on finding a way to reduce invalid offspring, and this leads to a form of *typed* genetic operators for GE. The idea of typed genetic operators is about improving the quality of offspring after applying genetic operators to genomes by choosing better crossover sites for the crossover operation and better mutation points for the mutation process. This can be done by integrating some constraints to incorporate semantic awareness in the process of choosing crossover and mutation points in order to improve the performance over standard genetic operators, which maintain only syntactic correctness. Moreover, the concept of semantic awareness is related to work in [101] that proposed a new semantically-based schema for implementing crossover in GP, extending from previous work of Semantic Aware Crossover (SAC) for GP.

During the mapping process, we save extra information for each codon, for use in the process of selecting crossover points and mutation points in our new operators. The following are the data that we record about each codon:

- Grammar rule ID corresponding with non-terminal symbol which is currently mapped with the codon;

- type of production rule designated by the codon;
- reference data for each recursive grammar rule.

For the type of the production rule, we have investigated two approaches for typed operators: A Recursive-Nonrecursive approach and an Internal-Leaf approach. Regarding the first one, we classify a production rules into 3 types:

- Recursive-productive (RP): if the grammar rule is recursive, and the production rule is recursive;
- Recursive-non-productive (RN): if the grammar rule is recursive, but the production rule is non-recursive;
- Non-recursive (NR): if the grammar rule is non-recursive.

In the second approach, we classify production rules into 2 types: Internal-node (I) and Leaf-node (L), and the production rules are assigned to which type based on the type of corresponding derivation node in the derivation tree derived from those production rules. In addition, each production rule has two more fields associated with it: Dereference Count (Deref. Count) and Reference Count (Ref. Count). Deref. Count is always -1 , while Ref. Count depends on how many LHS of the current production rule appear in the RHS of that rule, for instance, rule A.0 has Ref. Count at 2 since `<Expr>` appears twice in its right hand side and rule A.2 has Ref. Count 0 since no `<Expr>` occurs in its right hand side. These two fields are used in the process of constructing a reference table for the codons. Let us reconsider the BNF Grammar for symbolic problems, now with more information as described above; this is presented in Table 5-8.

Rule ID	LHS	Production Rules	Prod. ID	Recursive	Rule Type	Deref. Count	Ref. Count
A	<code><Expr></code> <code>::=</code>	<code><Expr> <Op> <Expr></code>	(A.0)	True	RP/I	-1	2
		<code><Pre-op> (<Expr>)</code>	(A.1)		RP/I	-1	1
		<code><Var></code>	(A.2)		RN/L	-1	0
Rule ID	LHS	Production Rules	Prod. ID	Recursive	Rule Type	Deref. Count	Ref. Count
B	<code><Op></code>	<code>+</code>	(B.0)	False	NR/I	-1	0

	::=	-	(B.1)		NR/I	-1	0
		*	(B.2)		NR/I	-1	0
		/	(B.3)		NR/I	-1	0

Rule ID	LHS	Production Rules	Prod. ID	Recursive	Rule Type	Deref. Count	Ref. Count
C	<Pre-op> ::=	Sin	(C.0)	False	NR/I	-1	0
		Cos	(C.1)		NR/I	-1	0
		Exp	(C.2)		NR/I	-1	0
		Log	(C.3)		NR/I	-1	0

Rule ID	LHS	Production Rules	Prod. ID	Recursive	Rule Type	Deref. Count	Ref. Count
D	<Var> ::=	X	(D.0)	False	-	-	-

Table 5-8: Symbolic BNF grammar with Dereference Count (Deref. Count), Reference Count (Ref. Count) and Rule Type for each production rule.

To demonstrate the process of typed genetic operators, Table 5-9 provides an example showing the mapping of an integer string in Figure 5-13 with a symbolic grammar, presenting the expression after mapping each codon for each mapping state and reference table of each rule. The integer string in Figure 5-13 comprises 15 integer values (15 codons), and these values will be fetched one by one in the mapping process. Consequently, there will be 15 mapping states for these codons. In Table 5-9-a, the top row is the starting symbol <Expr>, and we refer to <Expr> as E, <Op> as O and <Pre-op> as Pre. In addition, each cell in the table shows the current symbol associated with the corresponding codon. Moving to Table 5-9-b, its first row displays the modulus value of the current codon in each column, calculated by dividing the current codon value with the number of choices of production rules associated with that codon. For instance, if a codon value is 20 (Table 5-9-b at row 3) and the current symbol associated with it is E (Table 5-9-a at row 3, column 1), then by looking up the symbolic grammar from Table 5-8 we found that E is the LHS of rule A. As a result, the modulus value for this codon must be 2 (20 modulo 3), the remainder after dividing the codon value (20) by the number of production rules of rule A (3). Moreover, “6:A” in the first column, for example, means that the codon value is 6 and the rule currently linked with it is rule A. The last table, Table 5-9-c – reference table, presents the reference data for each rule for

each state of mapping. There are two values associated with each rule for each state: the pre-reference count (P.1) and post-reference count (P.2). The first value, P.1, is computed by following equation.

$$\begin{aligned}
 & \textit{Pre} - \textit{Reference Count (P.1)} \\
 &= \textit{Previous Post} - \textit{Reference Count} \\
 &+ \textit{Deref.Count of Current Produciton Rule}
 \end{aligned}$$

eq. 5-3

And, the P.2 is calculated by the equation below.

$$\begin{aligned}
 & \textit{Post} - \textit{Reference Count (P.2)} \\
 &= \textit{Current Pre} - \textit{Reference Count} \\
 &+ \textit{Ref.Count of Current Produciton Rule}
 \end{aligned}$$

eq. 5-4

The first initial values for P.1 and P.2 of all rules are zeros (the first data column with gray background colour in Table 5-9-c). It should be noted that there is no reference data for rule D in the reference table because rule D has only one choice and needs no codon for choosing the production rule, so reference data is not necessary. Table 5-9 will be referred to again in the sections about typed crossover operators and typed mutation operators.

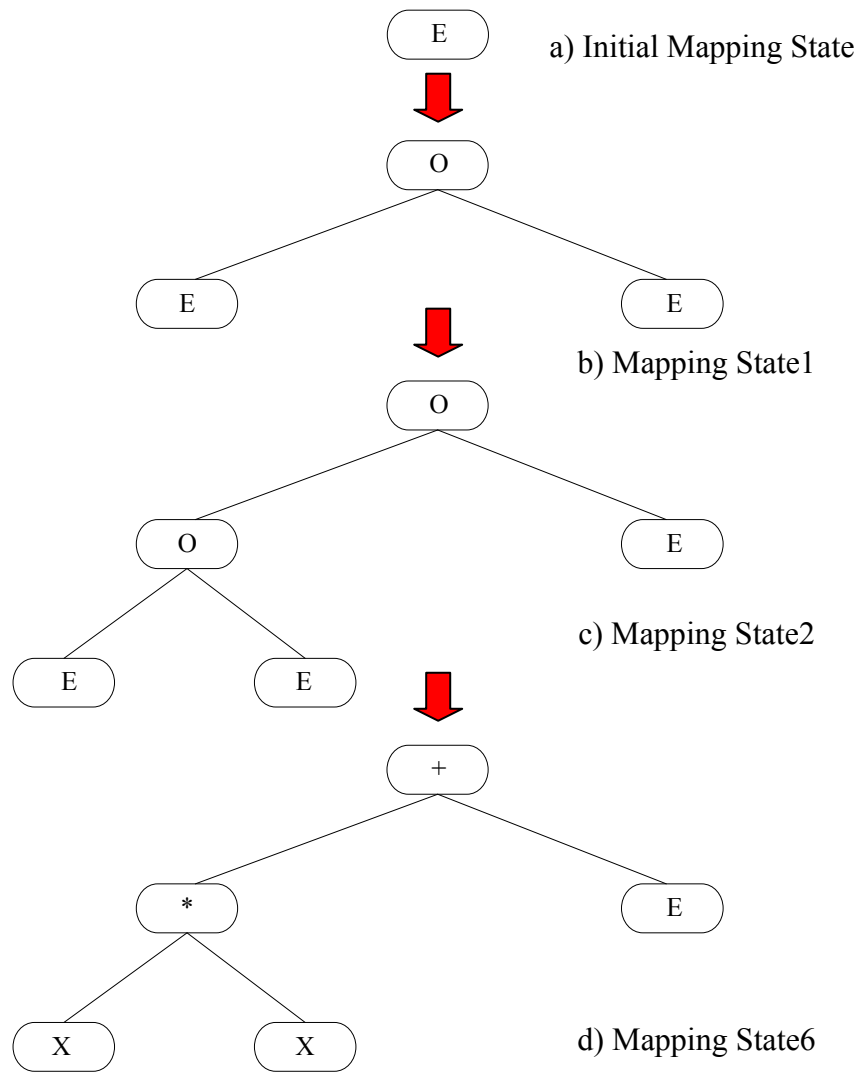


Figure 5-14: Derivation tree of codons in Figure 5-13 for initial mapping state, state1, state2 and state6.

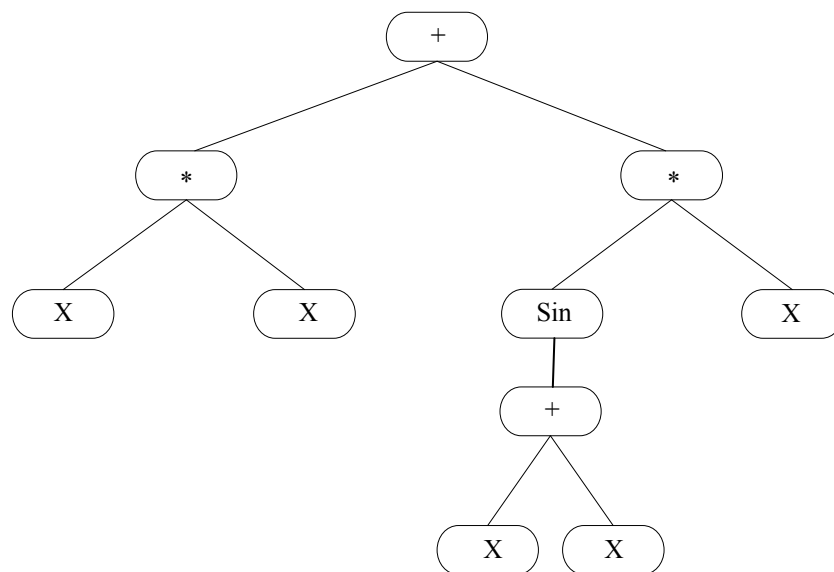


Figure 5-15: Derivation tree of codons in Figure 5-13 for final mapping state.

5.7.1 Two-point Typed Crossover Operator

Two-point crossover is selected as a standard typed crossover operator. The process of typed crossover begins with selecting two crossover sites for the first parent. We randomly choose the first crossover point and then the second crossover point is selected by looking up the reference data binding with the current codon starting from the first point. This forms the token (starting from the first point and ending at the second point), and our objective is to build a *valid token*, containing a subset of codons that can construct a complete subtree of the derivation tree of the whole set of codons. If the reference data of the current codon cannot make a valid token, then we proceed to the next codon and so on until we reach a suitable position that can construct a valid token. Thereafter, the first crossover point for the second parent is chosen from the list of codons with the same grammar ID as the first crossover point in the first parent. Next, the second crossover point is selected in same way as in the first parent. After we have all crossover points in both parents we then can exchange codons (genetic material) between two parents as in GA two-point crossover. The following three examples demonstrate the process of this two-point typed crossover. The first example shows the process of choosing the genetic material of the first parent, the second example displays the same process for the second parent, and the last example presents the process of swapping the genetic material between those two parents.

Initially, the type of production rules associated with the codons for the first parent is chosen with a given bias (`Crossover_Bias_Frequency`), allowing us to prefer recursive rules when using the Recursive-Nonrecursive approach, or internal-node rules for the Internal-Leaf approach, and then the first crossover point (site) among the codons in Figure 5-16 is randomly selected.

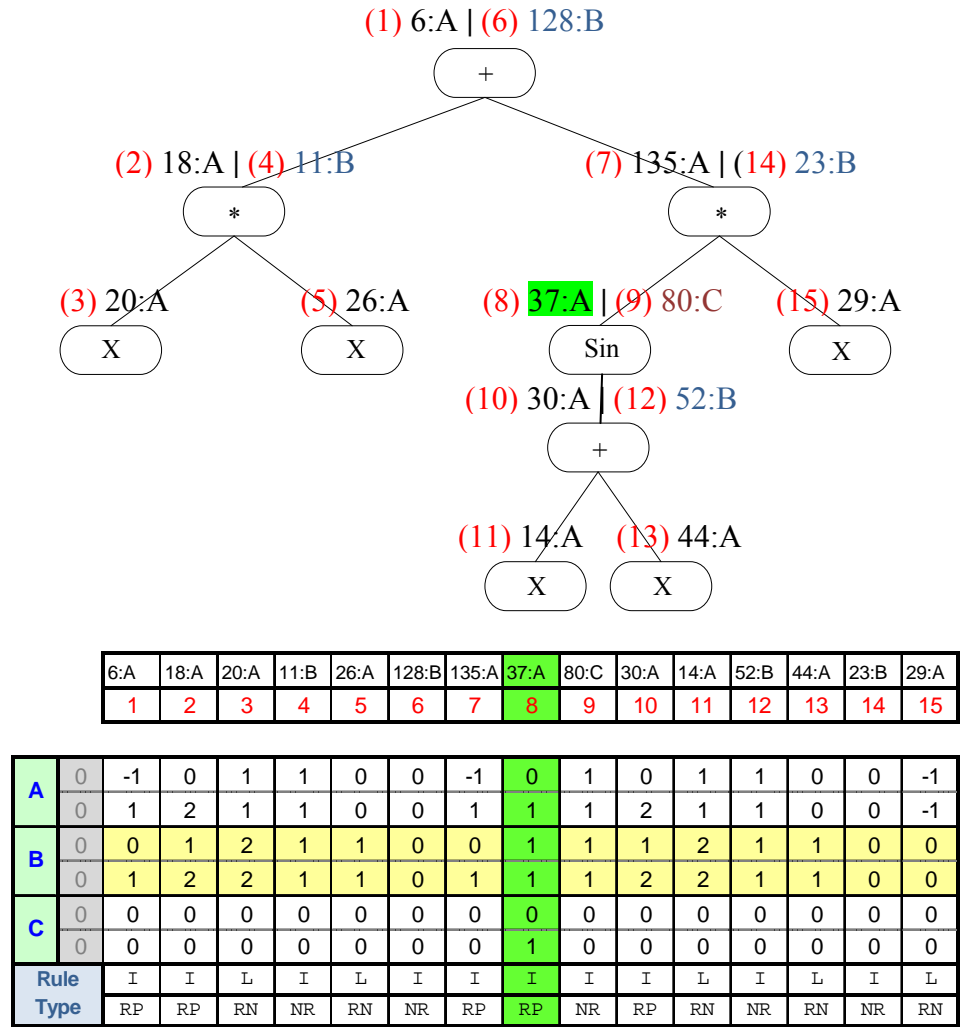


Figure 5-16: Derivation tree of the first parent with the first crossover point at position 8 highlighted with bright green colour.

Let us suppose that position 8 was selected as the first crossover point (the start point) with codon value of 37, and the associated production rule is rule A, and it is highlighted with bright green in both the derivation tree and reference table. The next step is to find the second crossover point (the end point) to construct the valid token. This can be done with the help of the reference table. The valid condition for an end point to make a *valid token* is that the P.1 value of the start point has to be equal to the P.1 value and P.2 value of the end point for each production rule in the reference table, and searching for the end point is begun from the start point (including itself). Starting from position 8, the codon that meets the validity condition is at position 13, as

indicated in Figure 5-17. In Figure 5-17 the valid token is highlighted with bright green, and the underlying subtree is surrounded with a bright green box.

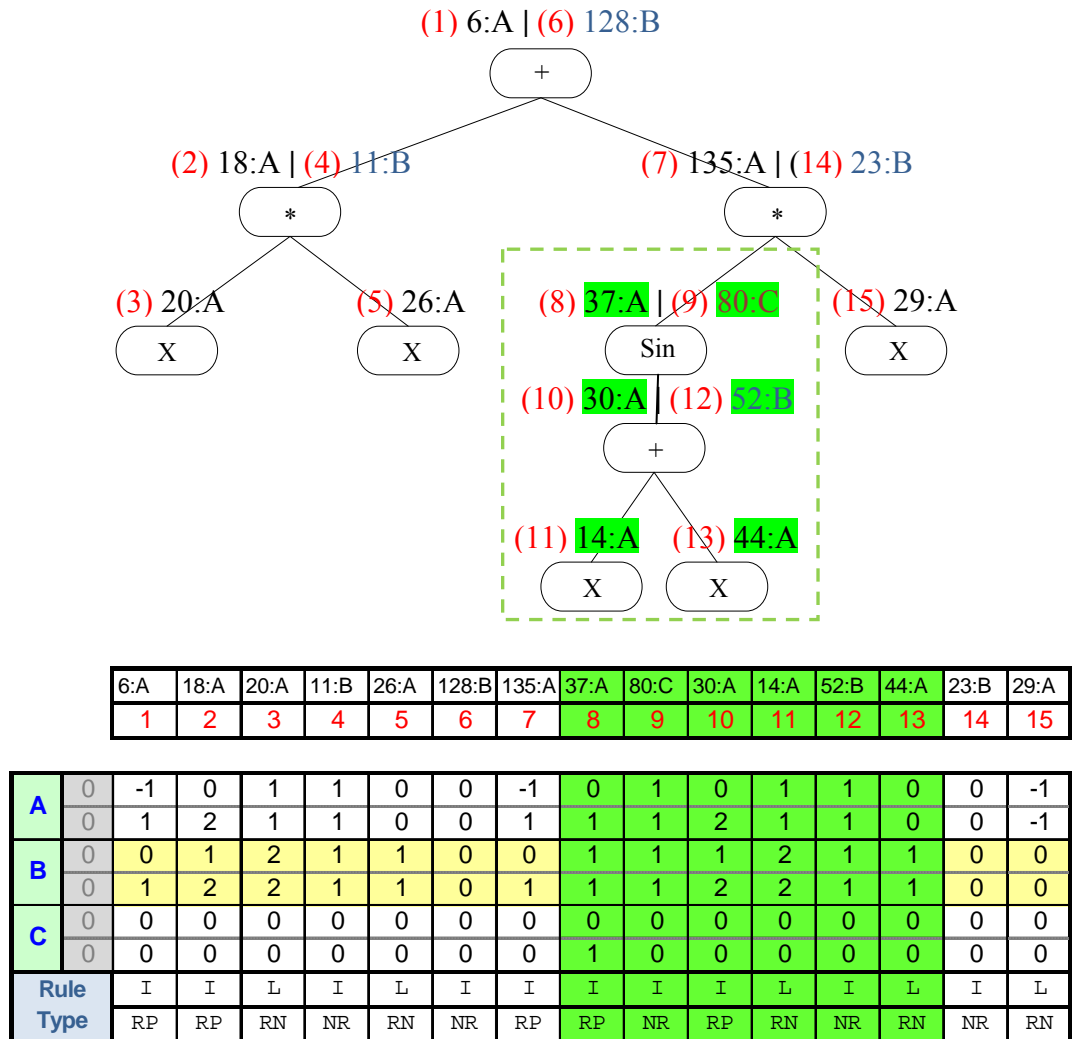


Figure 5-17: Derivation tree of the first parent with valid token starting at position 8 and ending at position 13 highlighted with bright green colour.

Now we have genetic material for the first parent, it is time for the second parent. Following the same procedure, the randomly selected first crossover point for the second parent is at position 2 and the end point is located at position 5 as shown in Figure 5-18, highlighted with bright blue for both the valid token and underlying subtree.

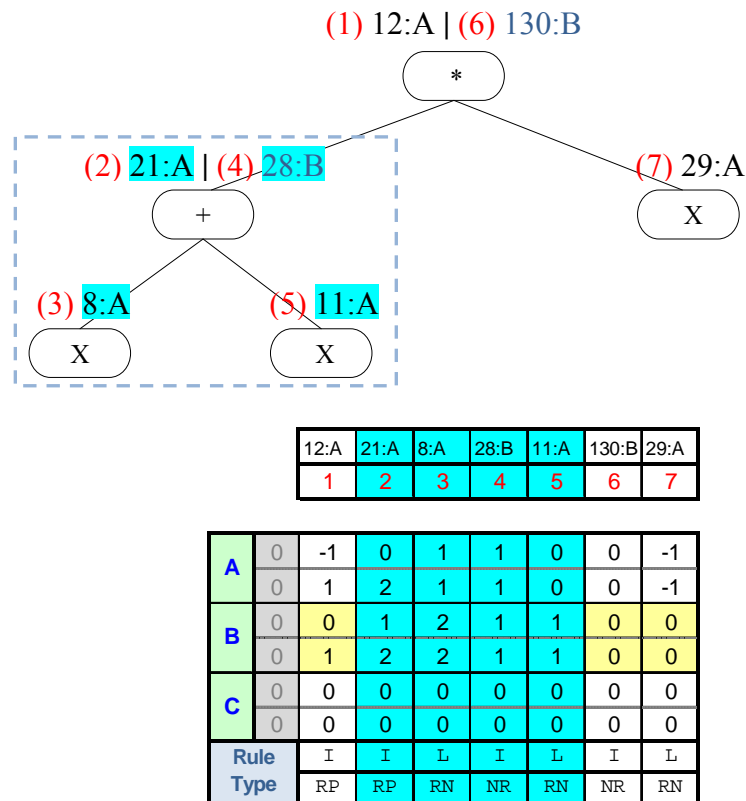


Figure 5-18: Derivation tree of the second parent with valid token starting at position 2 and ending at position 5 highlighted with bright blue.

The last step is the process of exchanging genetic material (valid tokens) between two parents, and this process results in producing two children (offspring). The first child is actually the first parent but containing some genetic material from the second parent, while in the second child the exchanged material is from the first parent. The derivation trees and codons for the first child and the second child are displayed in Figure 5-19 and Figure 5-20 respectively. Please notice that the valid token from the second parent is highlighted with bright blue in the first figure, and bright green is used for the valid token from the first parent in the second figure.

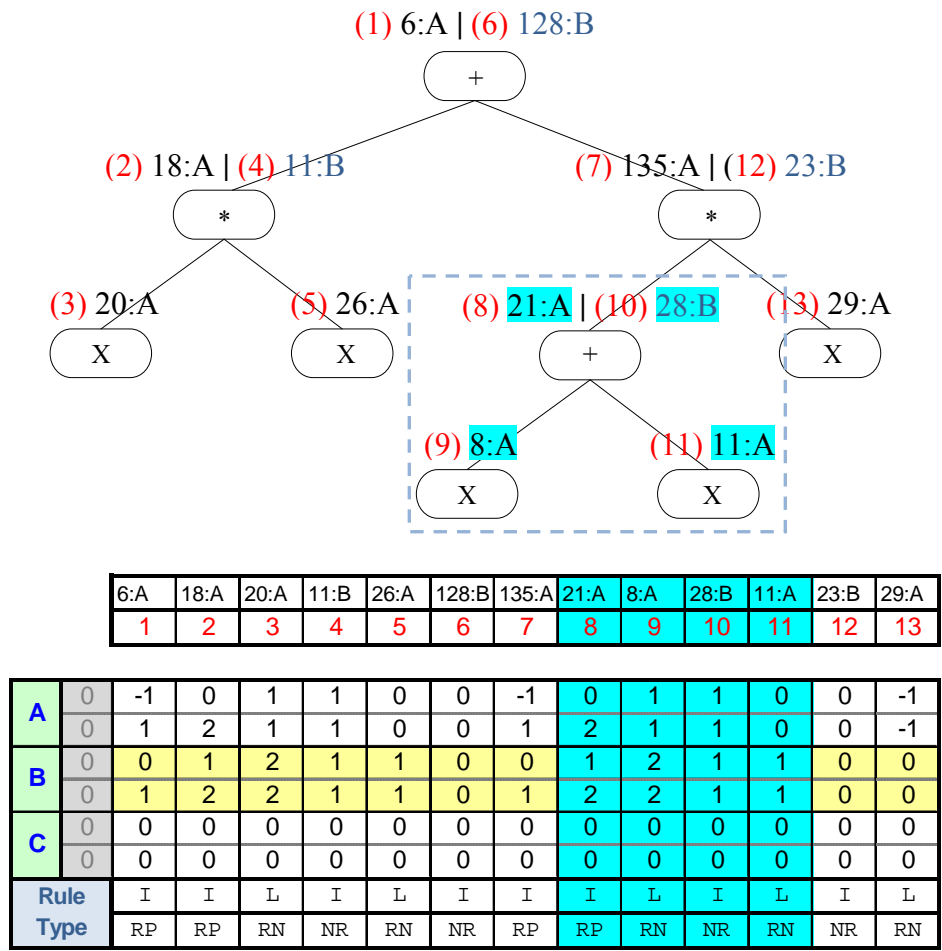
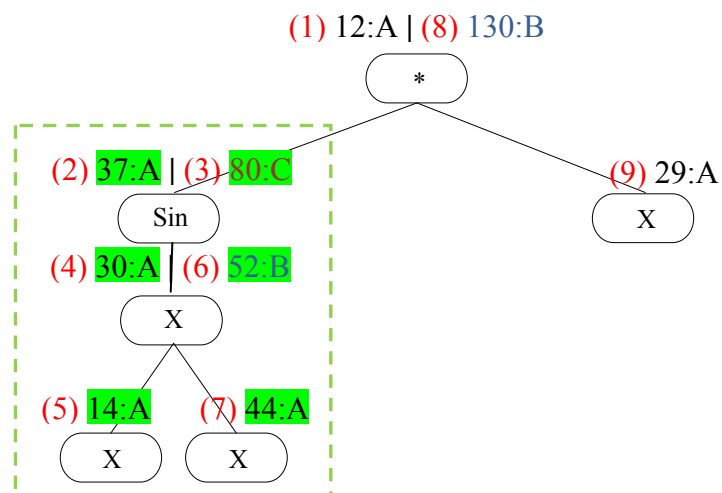


Figure 5-19: Derivation tree of the first child with the valid token from the second parent highlighted with bright blue colour.



12:A	37:A	80:C	30:A	14:A	52:B	44:A	130:B	29:A
1	2	3	4	5	6	7	8	9

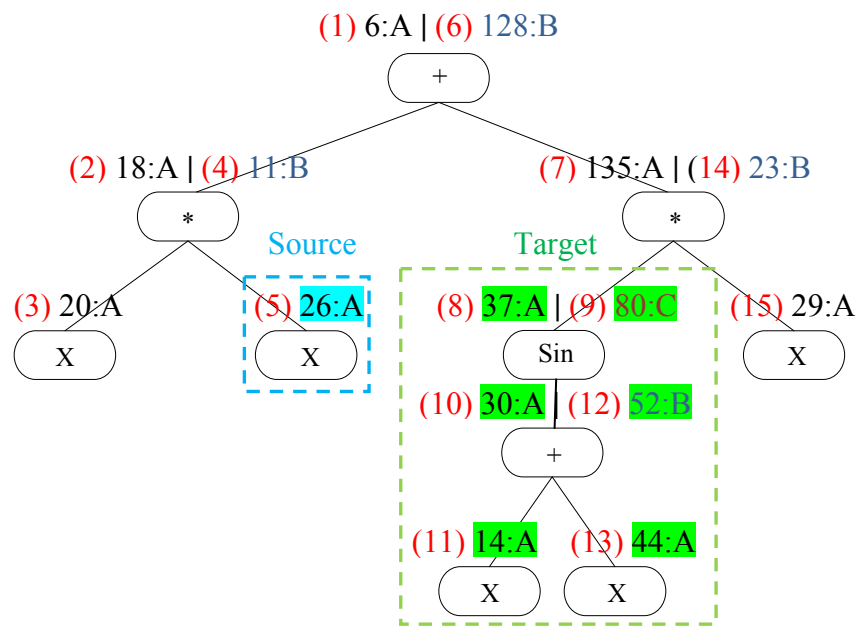
A	0	-1	0	1	0	1	1	0	0	-1
	0	1	1	1	2	1	1	0	0	-1
B	0	0	1	1	1	2	1	1	0	0
	0	1	1	1	2	2	1	1	0	0
C	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0
Rule Type	I	I	I	I	L	I	L	I	L	
	RP	RP	NR	RP	RN	NR	RN	NR	RN	

Figure 5-20: Derivation tree of the second child with the valid token from the first parent highlighted with bright green colour.

5.7.2 Typed Mutation Operator

The concept of typed mutation is based on the same idea of dealing with valid tokens, constructed with information from the reference table. Beginning in the same way as in the typed crossover process, the type of production rules associated with codons is chosen first with a bias (*Mutation_Bias_Frequency*), to prefer the recursive rules when we are using the Recursive-Nonrecursive approach or the internal-node rules in the case of the Internal-Leaf approach, and then the mutation point(s) (site(s)) of codons are randomly selected. After we have the mutation point(s), we have six choices of mutation operator. The details of each choice are given below.

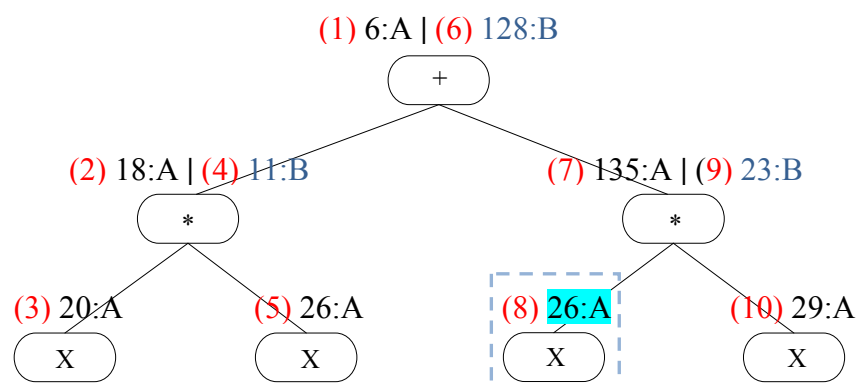
5.7.2.1 Subcodon Mutation – we construct the valid token by using the first mutation point (target) as its start point, and then replace it with another valid token whose start point is the second mutation point (source) with the same rule ID of the first mutation point (see Figure 5-21).



6:A	18:A	20:A	11:B	26:A	128:B	135:A	37:A	80:C	30:A	14:A	52:B	44:A	23:B	29:A
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

A	0	-1	0	1	1	0	0	-1	0	1	0	1	1	0	0	-1
	0	1	2	1	1	0	0	1	1	1	2	1	1	0	0	-1
B	0	0	1	2	1	1	0	0	1	1	1	2	1	1	0	0
	0	1	2	2	1	1	0	1	1	1	2	2	1	1	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Rule Type	I	I	L	I	L	I	I	I	I	I	I	L	I	L	I	L
	RP	RP	RN	NR	RN	NR	RP	RP	NR	RP	RN	NR	RN	NR	RN	

a) Derivation tree of the individual with the first valid token starting at the first mutation point (target) at position 8 and the second valid token starting at the second mutation point (source) at position 5 highlighted with bright green colour and bright blue colour respectively.



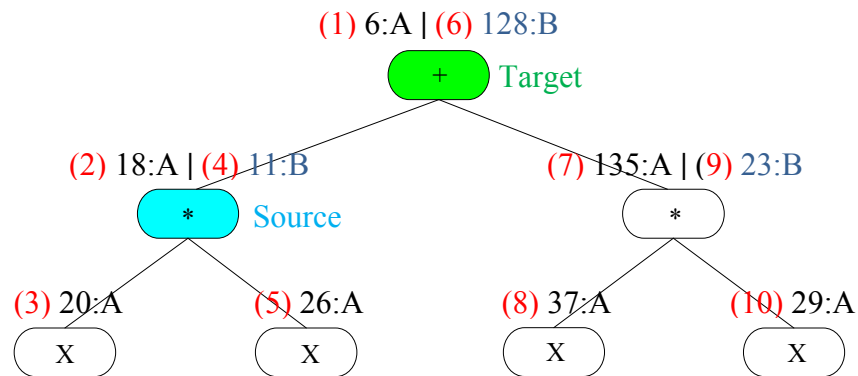
6:A	18:A	20:A	11:B	26:A	128:B	135:A	26:A	23:B	29:A
1	2	3	4	5	6	7	8	9	10

A	0	-1	0	1	1	0	0	-1	0	0	-1
	0	1	2	1	1	0	0	1	0	0	-1
B	0	0	1	2	1	1	0	0	1	0	0
	0	1	2	2	1	1	0	1	1	0	0
C	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
Rule Type	I	I	L	I	L	I	I	L	I	L	
	RP	RP	RN	NR	RN	NR	RP	RN	NR	RN	

b) Derivation tree of the individual after replacing the first valid token with the second one.

Figure 5-21: Subcodon Mutation

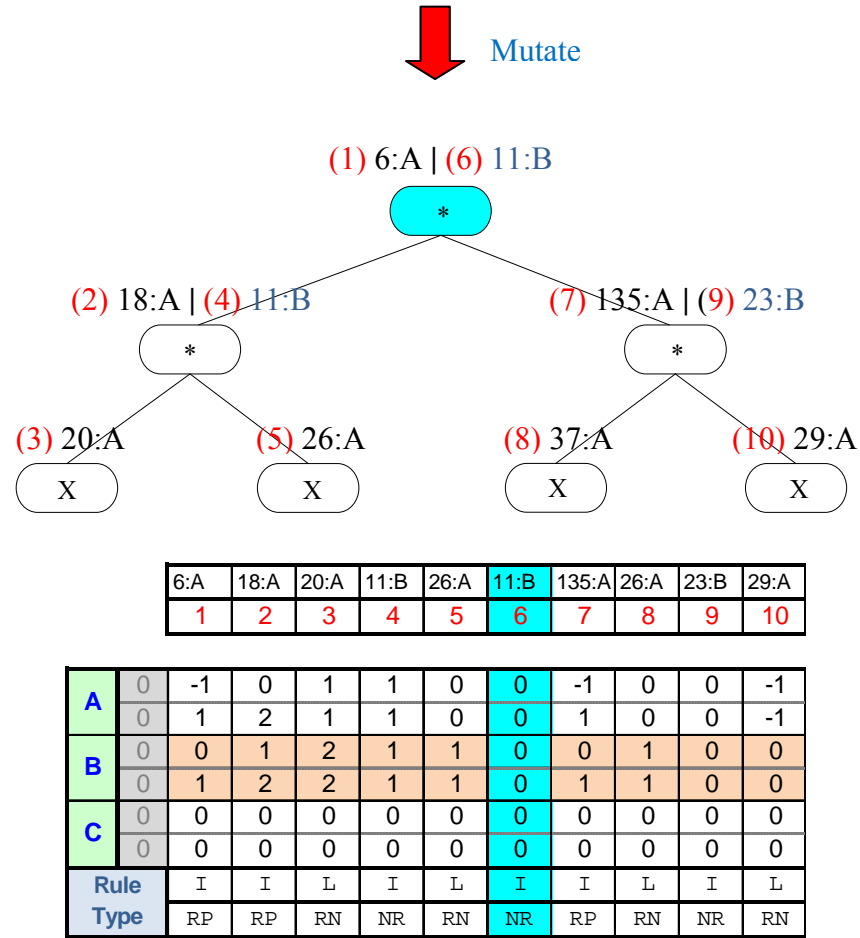
5.7.2.2 *Point Mutation* – if the first mutation point (target) binds with a non-recursive production rule, then we exchange it with a randomly selected second mutation point (source) with the same rule ID to select another choice associated with that non-recursive production rule (see Figure 5-22).



6:A	18:A	20:A	11:B	26:A	128:B	135:A	26:A	23:B	29:A
1	2	3	4	5	6	7	8	9	10

A	0	-1	0	1	1	0	0	-1	0	0	-1
	0	1	2	1	1	0	0	1	0	0	-1
B	0	0	1	2	1	1	0	0	1	0	0
	0	1	2	2	1	1	0	1	1	0	0
C	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
Rule Type	I	I	L	I	L	I	I	L	I	L	
	RP	RP	RN	NR	RN	NR	RP	RN	NR	RN	

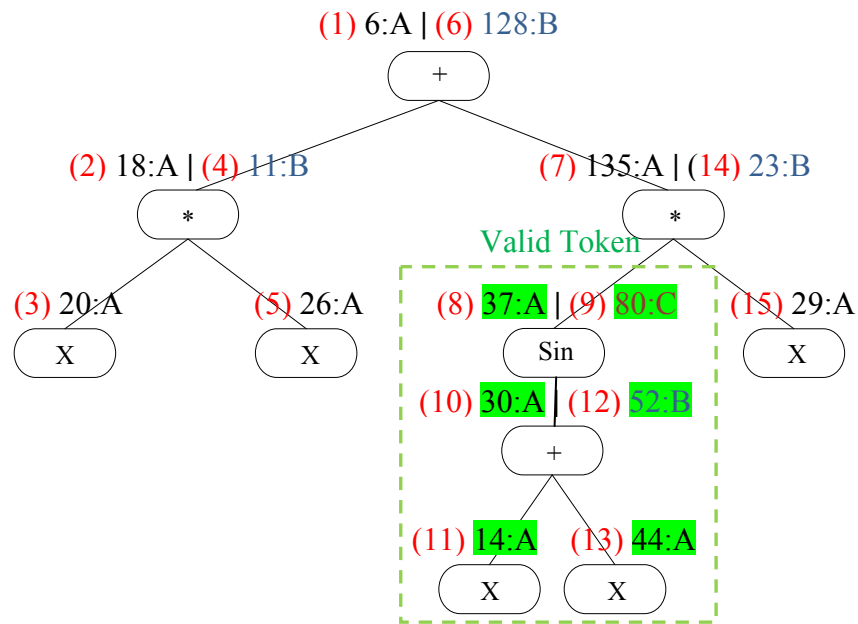
a) Derivation tree of the individual with the first mutation point (target) at position 6 and the second mutation point (source) at position 4 highlighted with bright green colour and bright blue colour respectively.



b) Derivation tree of the individual after replacing the target mutation point with the source mutation point.

Figure 5-22: Point Mutation.

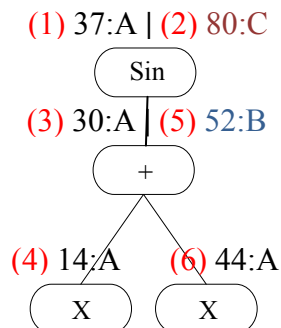
5.7.2.3 *Hoist* – if the mutation point binds with the recursive rule of which the LHS is the start symbol, then we construct a valid token whose start point is that mutation point, and generate a new individual from that valid token (see Figure 5-23).



6:A	18:A	20:A	11:B	26:A	128:B	135:A	37:A	80:C	30:A	14:A	52:B	44:A	23:B	29:A
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

A	0	-1	0	1	1	0	0	-1	0	1	0	1	1	0	0	-1
	0	1	2	1	1	0	0	1	1	1	2	1	1	0	0	-1
B	0	0	1	2	1	1	0	0	1	1	1	2	1	1	0	0
	0	1	2	2	1	1	0	1	1	1	2	2	1	1	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Rule Type	I	I	L	I	L	I	I	I	I	I	I	L	I	L	I	L
	RP	RP	RN	NR	RN	NR	RP	RP	NR	RP	RN	NR	RN	NR	RN	

a) Derivation tree of the individual with the valid token starting at position 8 highlighted with bright green colour.



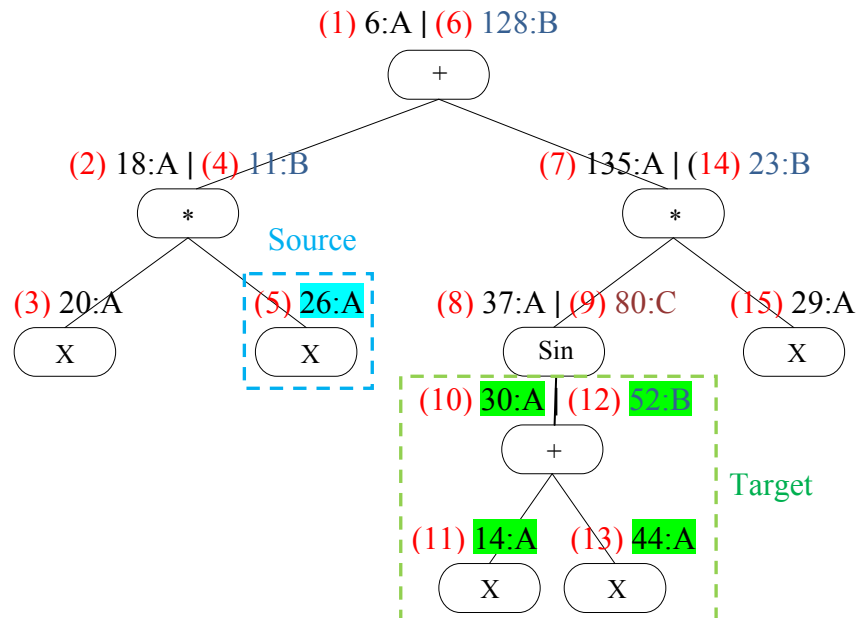
	37:A	80:C	30:A	14:A	52:B	44:A
	1	2	3	4	5	6

A	0	-1	0	-1	0	0	-1
	0	0	0	1	0	0	-1
B	0	0	0	0	1	0	0
	0	0	0	1	1	0	0
C	0	0	0	0	0	0	0
	0	1	0	0	0	0	0
Rule Type	I	I	I	L	I	L	
	RP	NR	RP	RN	NR	RN	

b) Derivation tree of the new individual generated from the valid token.

Figure 5-23: Hoist mutation.

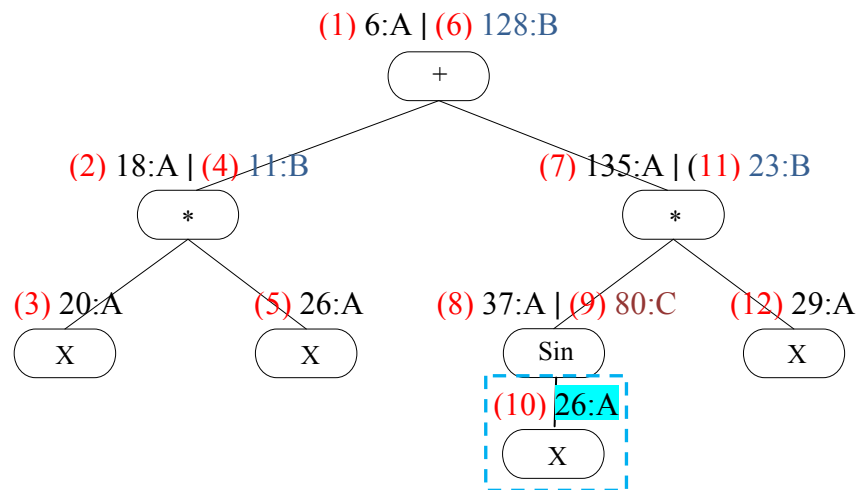
5.7.2.4 *Collapse Subcodon Mutation* – if the first mutation point (target) is associated with a recursive-productive production rule or an internal-node rule, then we construct the valid token by using the first mutation point (target) as its start point. Next, we replace it with another valid token of which the start point is the second mutation point (source) with the same rule ID, and which binds with a recursive-non-productive production rule or a leaf-node rules (see Figure 5-24).



6:A	18:A	20:A	11:B	26:A	128:B	135:A	37:A	80:C	30:A	14:A	52:B	44:A	23:B	29:A
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

A	0	-1	0	1	1	0	0	-1	0	1	0	1	0	0	-1
	0	1	2	1	1	0	0	1	1	1	2	1	1	0	-1
B	0	0	1	2	1	1	0	0	1	1	1	2	1	1	0
	0	1	2	2	1	1	0	1	1	1	2	2	1	1	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Rule Type	I	I	L	I	L	I	I	I	I	I	L	I	L	I	L
	RP	RP	RN	NR	RN	NR	RP	RP	NR	RP	RN	NR	RN	NR	RN

a) Derivation tree of the individual with the first valid token starting at the first mutation point (target) at position 10 and the second valid token starting at the second mutation point (source) at position 5 highlighted with bright green colour and bright blue colour respectively.



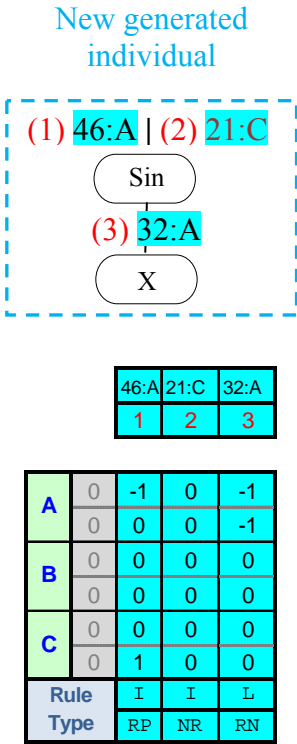
6:A	18:A	20:A	11:B	26:A	128:B	135:A	37:A	80:C	26:A	23:B	29:A
1	2	3	4	5	6	7	8	9	10	11	12

A	0	-1	0	1	1	0	0	-1	0	1	0	0	-1
	0	1	2	1	1	0	0	1	1	1	0	0	-1
B	0	0	1	2	1	1	0	0	1	1	1	0	0
	0	1	2	2	1	1	0	1	1	1	1	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	0	0	0	0
Rule Type	I	I	L	I	L	I	I	I	I	L	I	L	
	RP	RP	RN	NR	RN	NR	RP	RP	NR	RN	NR	RN	

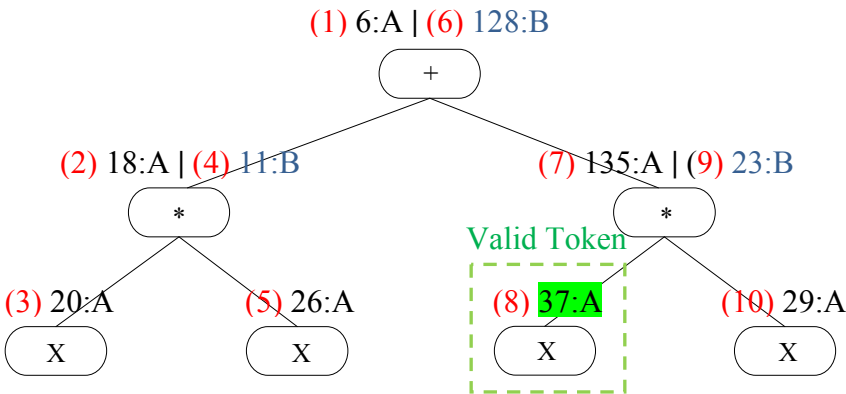
b) Derivation tree of the individual after replacing the first valid token with the second one.

Figure 5-24: Collapse Subcodon Mutation.

5.7.2.5 *Grow Mutation* – we construct the valid token by using the mutation point as its start point, and then replace it with another valid token of a newly generated individual of which the root is the start point with the same rule ID as the mutation point (see Figure 5-25).



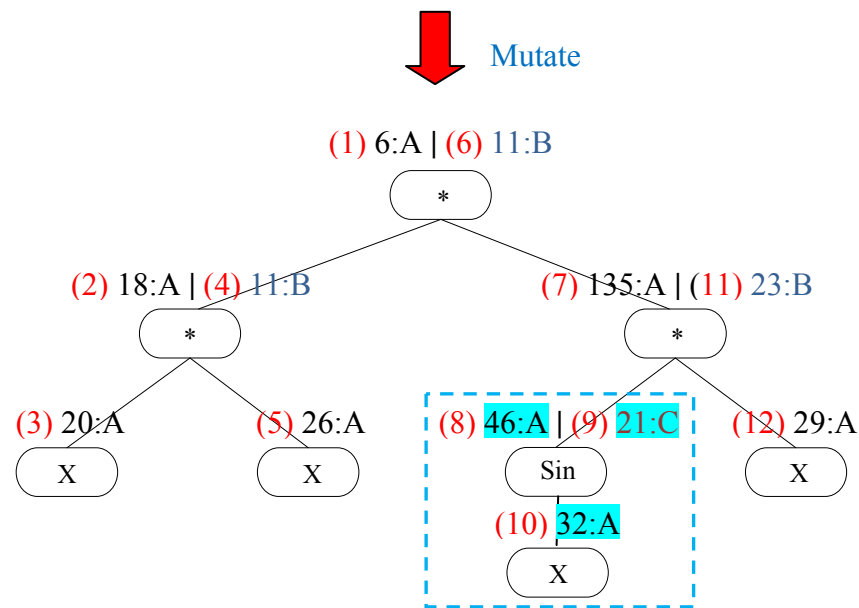
a) Derivation tree of the new generated individual.



6:A	18:A	20:A	11:B	26:A	128:B	135:A	26:A	23:B	29:A
1	2	3	4	5	6	7	8	9	10

A	0	-1	0	1	1	0	0	-1	0	0	-1
	0	1	2	1	1	0	0	1	0	0	-1
B	0	0	1	2	1	1	0	0	1	0	0
	0	1	2	2	1	1	0	1	1	0	0
C	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
Rule Type	I	I	L	I	L	I	I	L	I	L	
	RP	RP	RN	NR	RN	NR	RP	RN	NR	RN	

b) Derivation tree of the individual with the valid token starting at position 8 highlighted with bright green colour.



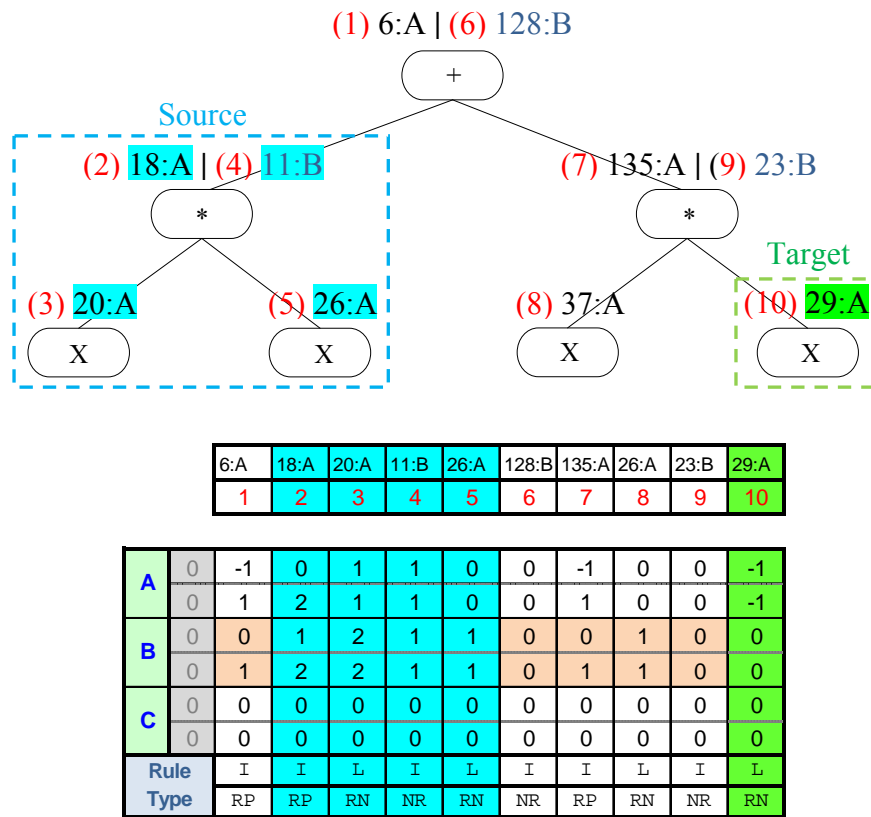
6:A	18:A	20:A	11:B	26:A	128:B	135:A	46:A	21:C	32:A	23:B	29:A
1	2	3	4	5	6	7	8	9	10	11	12

A	0	-1	0	1	1	0	0	-1	0	1	0	0	-1
	0	1	2	1	1	0	0	1	1	1	0	0	-1
B	0	0	1	2	1	1	0	0	1	1	1	0	0
	0	1	2	2	1	1	0	1	1	1	1	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	0	0	0	0
Rule Type	I	I	L	I	L	I	I	I	I	L	I	L	
	RP	RP	RN	NR	RN	NR	RP	RP	NR	RN	NR	RN	

c) Derivation tree of the individual after replacing its valid token with the valid token of new generated individual.

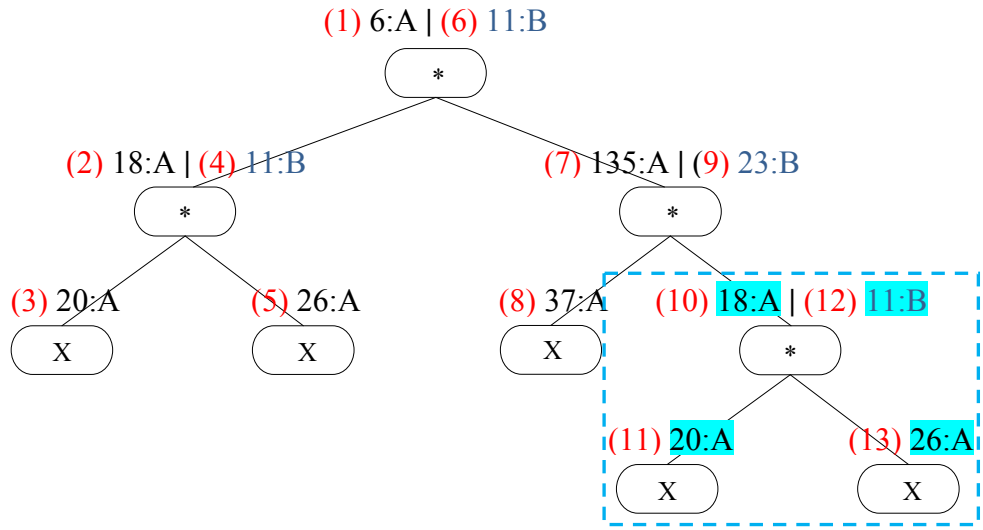
Figure 5-25: Grow Mutation.

5.7.2.6 *Expansion Mutation* – if the first mutation point (target) binds with a recursive-non-productive production rules or a leaf-node rules, then we construct the valid token by using the first mutation point (target) as its start point. Next, we replace it with another valid token of which the start point is a second mutation point (source) with the same rule ID as the start point and which binds with recursive-productive production rules or internal-node rules (see Figure 5-26).



a) Derivation tree of the individual with the first mutation point (target) at position 10 and the second mutation point (source) at position 2 highlighted with bright green colour and bright blue colour respectively.





6:A	18:A	20:A	11:B	26:A	128:B	135:A	26:A	23:B	18:A	20:A	11:B	26:A
1	2	3	4	5	6	7	8	9	10	11	12	13

A	0	-1	0	1	1	0	0	-1	0	0	-1	0	0	-1
	0	1	2	1	1	0	0	1	0	0	1	0	0	-1
B	0	0	1	2	1	1	0	0	1	0	0	1	0	0
	0	1	2	2	1	1	0	1	1	0	1	1	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rule Type	I	I	L	I	L	I	I	L	I	I	L	I	L	
	RP	RP	RN	NR	RN	NR	RP	RN	NR	RP	RN	NR	RN	

b) Derivation tree of the individual after replacing the first valid token with the second one.

Figure 5-26: Expansion Mutation.

The availability and use of such typed operators for GE makes our GE obviously different from that proposed in [77], and a key aspect of this difference is an ability to mix two types of genetic operators: normal (standard) genetic operators and typed genetic operators. We can control the rates of using normal crossover (one-point crossover for binary strings) and typed crossover by using a *Typed Crossover Rate* and the rate of using normal mutation (flip mutation for binary strings) and typed mutation via a *Typed Mutation Rate*. For instance, if the typed crossover rate is 0.25, then this means that 75% of the time GE uses normal crossover and for the rest (25%) it uses typed crossover. The typed mutation rate is interpreted in the same way.

5.8 Results of GE on various test problems

To see the effect of exploiting both types of genetic operators, we have tried several configurations with different values of typed crossover rates and typed mutation rates together as shown in Table 5-10. Results of all configurations in this table on six test problems are reported in Table 5-11 for both approaches (Recursive-non-recursive approach and Internal-Leaf approach).

GE Config	Typed Crossover Rate	Typed Mutation Rate
CF01	0.00	1.00
CF02	0.25	1.00
CF03	0.50	1.00
CF04	0.70	1.00
CF05	0.80	1.00
CF06	0.90	1.00
CF07	1.00	1.00
CF08	1.00	0.90
CF09	1.00	0.80
CF10	0.90	0.90

Table 5-10: Typed genetic operator testing configuration table.

Referring to section 5.5, we saw that the results of using valid map methods, in particular MAP03, led to consistent improvement over our version of a standard GE implementation, but not enough to compare particularly favourably with GP. We are now going to see whether combining MAP03 with the use of typed genetic operators for GE can lead to any further advance in performance over the results we saw earlier. The new results are summarised in Table 5-11. As we saw in Table 5-7, and each row in Table 5-11 is compared with the results of standard GP as the baseline (row 1.1, highlighted in green), and a blue highlight in a cell indicates that the performance of the configuration of that method (identified by the row) on that problem (identified by the column) is better than or equal to standard GP (row 1.1) on the same problem.

#	Method	ANT	SR1	SR2	SR3	SR4	SR5
1	GP - Standard						
1.1	GP (Standard)	0.38	0.46	1.00	0.87	0.22	0.13
1.2	GP (O'Neill and Ryan's Work)	0.15	0.99	0.37	N/A	N/A	N/A
2	GE - Standard						
2.1	GE (Standard)	0.89	0.15	0.22	0.27	0.06	0.03
2.2	GE (Ramped-half-and-half Initialization)	0.73	0.14	0.86	0.49	0.29	0.08
2.3	GE (O'Neill and Ryan's Work)	0.90	0.98	0.80	N/A	N/A	N/A
4	GE - With Valid Map Method and Typed Genetic Operators with Recursive-Non-recursive Approach						
4.1	Random Initialization (INIT1)						
4.1.1	GE-MAP03-INIT1-CF01	0.83	0.34	0.96	0.35	0.09	0.03
4.1.2	GE-MAP03-INIT1-CF02	0.85	0.40	0.97	0.44	0.11	0.02
4.1.3	GE-MAP03-INIT1-CF03	0.86	0.43	0.96	0.47	0.14	0.03
4.1.4	GE-MAP03-INIT1-CF04	0.85	0.54	0.97	0.53	0.11	0.05
4.1.5	GE-MAP03-INIT1-CF05	0.81	0.55	1.00	0.58	0.15	0.08
4.1.6	GE-MAP03-INIT1-CF06	0.79	0.59	0.96	0.57	0.22	0.08
4.1.7	GE-MAP03-INIT1-CF07	0.72	0.65	0.97	0.58	0.20	0.08
4.1.8	GE-MAP03-INIT1-CF08	0.71	0.64	0.98	0.59	0.21	0.11
4.1.9	GE-MAP03-INIT1-CF09	0.64	0.57	0.99	0.56	0.27	0.19
4.1.10	GE-MAP03-INIT1-CF10	0.84	0.60	0.99	0.52	0.20	0.10
4.2	Ramped-half-and-half Initialization (INIT2)						
4.2.1	GE-MAP03-INIT2-CF01	0.75	0.39	0.99	0.48	0.09	0.04
4.2.2	GE-MAP03-INIT2-CF02	0.81	0.41	1.00	0.61	0.08	0.05
4.2.3	GE-MAP03-INIT2-CF03	0.82	0.51	1.00	0.69	0.21	0.07
4.2.4	GE-MAP03-INIT2-CF04	0.73	0.54	1.00	0.72	0.18	0.02
4.2.5	GE-MAP03-INIT2-CF05	0.71	0.54	1.00	0.74	0.21	0.06
4.2.6	GE-MAP03-INIT2-CF06	0.69	0.57	1.00	0.68	0.17	0.10
4.2.7	GE-MAP03-INIT2-CF07	0.65	0.58	0.99	0.76	0.29	0.07
4.2.8	GE-MAP03-INIT2-CF08	0.61	0.59	0.99	0.78	0.36	0.10
4.2.9	GE-MAP03-INIT2-CF09	0.68	0.61	1.00	0.76	0.37	0.10
4.2.10	GE-MAP03-INIT2-CF10	0.71	0.56	1.00	0.77	0.25	0.10
5	GE - With Valid Map Method and Typed Genetic Operators with Internal-Leaf Approach						
5.1	Random Initialization (INIT1)						
5.1.1	GE-MAP03-INIT1-CF01	0.87	0.34	0.96	0.35	0.09	0.03
5.1.2	GE-MAP03-INIT1-CF02	0.81	0.38	0.98	0.46	0.09	0.07
5.1.3	GE-MAP03-INIT1-CF03	0.92	0.53	0.97	0.59	0.14	0.07
5.1.4	GE-MAP03-INIT1-CF04	0.81	0.48	1.00	0.67	0.20	0.11
5.1.5	GE-MAP03-INIT1-CF05	0.83	0.54	1.00	0.64	0.19	0.15
5.1.6	GE-MAP03-INIT1-CF06	0.80	0.61	0.97	0.61	0.16	0.13
5.1.7	GE-MAP03-INIT1-CF07	0.76	0.55	1.00	0.62	0.18	0.08
5.1.8	GE-MAP03-INIT1-CF08	0.75	0.68	1.00	0.68	0.23	0.12
5.1.9	GE-MAP03-INIT1-CF09	0.74	0.59	0.98	0.73	0.31	0.14
5.1.10	GE-MAP03-INIT1-CF10	0.81	0.52	0.98	0.73	0.18	0.15
5.2	Ramped-half-and-half Initialization (INIT2)						
5.2.1	GE-MAP03-INIT2-CF01	0.82	0.39	0.99	0.48	0.09	0.04
5.2.2	GE-MAP03-INIT2-CF02	0.75	0.33	1.00	0.63	0.16	0.05
5.2.3	GE-MAP03-INIT2-CF03	0.84	0.55	1.00	0.67	0.14	0.06
5.2.4	GE-MAP03-INIT2-CF04	0.77	0.46	1.00	0.80	0.17	0.08
5.2.5	GE-MAP03-INIT2-CF05	0.73	0.49	1.00	0.80	0.19	0.07
5.2.6	GE-MAP03-INIT2-CF06	0.81	0.53	1.00	0.78	0.29	0.10
5.2.7	GE-MAP03-INIT2-CF07	0.66	0.60	1.00	0.86	0.31	0.09
5.2.8	GE-MAP03-INIT2-CF08	0.67	0.51	1.00	0.85	0.35	0.19
5.2.9	GE-MAP03-INIT2-CF09	0.63	0.43	0.99	0.90	0.47	0.20
5.2.10	GE-MAP03-INIT2-CF10	0.76	0.44	0.99	0.84	0.29	0.13

Table 5-11: Results of standard GP, standard GE and GE with valid map methods and typed genetic operators on six problems, displaying success rate of finding the solution over 100 runs for two approaches: Recursive-non-recursive approach and Internal-Leaf approach, and the highest values of ANT, SR1, SR2, SR3, SR4, SR5 are surrounded by red dash-line boxes.

Let consider the first problem in Table 5-11, the Santa Fe ant trial – ANT; the best configuration was CF03, half the time using normal crossover and half the time using

typed crossover, and 100% using typed mutation, with the Internal-Leaf approach and with random initialization. For the symbolic regression problem SR1, the target function being of polynomial degree 4, the highest success rate was again obtained by using the Internal-Leaf approach with random initialization, but this time with configuration CF08, using pure (100%) typed crossover and 90% typed mutation. On the symbolic integration problem, SR2, all configurations from both approaches succeeded to find solutions with a success rate varying between 96% to 100%, considerably improved over the standard approaches in rows 1.1—1.2 and 2.1—2.3. For the three additional problems (SR3, SR4 and SR5), the highest success rates we see are again associated with the Internal-Leaf approach, however, this time with ramped-half-and-half initialization and configuration CF09 (100% typed crossover and 80% typed mutation).

Problem	Method	Success Rate	90% Confidence Interval	Method	Success Rate	90% Confidence Interval
ANT	Standard GP	0.38	[0.3043 0.4620]	GE-INTERNAL-LEAF-MAP03-INIT1-CF03	0.92	[0.8635 0.9543]
SR1	Standard GP	0.46	[0.3802 0.5420]	GE-INTERNAL-LEAF-MAP03-INIT1-CF08	0.68	[0.5994 0.7511]
SR2	Standard GP	1.00	[0.9737 1.0000]	GE-INTERNAL-LEAF-MAP03-INIT2-CF02-CF08	1.00	[0.9737 1.0000]
SR3	Standard GP	0.87	[0.8048 0.9157]	GE-INTERNAL-LEAF-MAP03-INIT2-CF09	0.90	[0.8396 0.9393]
SR4	Standard GP	0.22	[0.1597 0.2950]	GE-INTERNAL-LEAF-MAP03-INIT2-CF09	0.47	[0.3898 0.5518]
SR5	Standard GP	0.13	[0.0843 0.1952]	GE-INTERNAL-LEAF-MAP03-INIT2-CF09	0.20	[0.1425 0.2733]

Table 5-12: Confidence interval at 90% of the best methods for each problem compared with standard GP.

To provide a statistically valid comparison of performance with standard GP, Table 5-12 provides 90% confidence intervals relating to the success rates of standard GP compared in turn with the methods that achieved the best success rate on each problem. These comparisons are based on a standard statistical method for obtaining a confidence interval around a proportion (here we are comparing success rates, each of which is a proportion emerging from trials over 100 experiments). There are two conditions that are required for us to consider method *A* better than method *B* with 90% confidence. First, the 90% confidence interval for method *A* has to higher than the 90% confidence interval for method *B*, and second, these intervals must not overlap. Looking at the first

row in Table 5-12, referring to the ANT problem, Table 5-12 tells us that there is a 90% chance that the true success rate of GE with configuration CF03 on ANT is between 86.35% and 95.43%. This statistically outperforms standard GP, whose 90% confidence interval is between 30.43% and 46.20%. Statistically superior performance of GE over GP was also found in SR1 and SR4 with success rates in SR1 between 38.02% to 54.20% for GP and 59.94% to 75.11% for GE, and in SR4 between 15.97% to 29.50% for GP and 38.98% to 55.18% for GE. In the remaining problems, SR2, SR3 and SR5, the results were inconclusive as the 90% confidence intervals between standard GP and GE methods were overlapping.

When we consider the best result achieved across all configurations, all initialisation strategies, and all mapping methods, and consider this for each problem in turn, we find the following. There are 8 cases in which the best (or equal best) result for a problem is achieved by the valid map method with the Recursive-Non-recursive approach, and there are 16 cases in which the best (or equal best) result for a problem is achieved by the Internal-Leaf approach. On this basis, the Internal-Leaf approach seems more promising than the Recursive-Non-recursive approach. In consequence, we used the Internal-Leaf approach on the financial trading problems in the next section. Deciding which overall configuration should be used, however, turns out to be a hard decision because there was no configuration with particularly prominent results for all six test problems. Having said that, integrating both valid map method MAP03 and typed genetic operators into GE has clearly shown that it can improve GE performance, and this has been reflected in the results in Table 5-11. To carry out on empirical tests with our GE on a more ambitious problem – financial trading, we have to pick up a GE configuration from Table 5-10 as a standard configuration for this test, so we have made an arbitrary choice to choose CF07 using pure typed crossover and pure typed mutation as our standard GE configuration.

5.9 Results of GE on Real Word Problem – Financial Trading

In all GE experiments in this section, we set the GE grammars and GE parameters for trading in a way that is as similar as possible to match the GP function and terminal sets and GP parameters used in sections 3.2.2 and 3.3.1 respectively. We have run both GE and GP for three types of financial trading: monthly, weekly and daily, using the same data and data splits as before, allowing direct comparison with the GP results. Let us begin with monthly trading. The average returns of trading on the eight monthly data splits for both GE and GP are given in Table 5-13. Having identified the method with the highest return on each data split, we then automatically consider the hypothesis that this is the best method among those compared on that data split, and then test that hypothesis with paired one-tailed T-Tests, assuming unequal variance. The T-Test results in Table 5-14 support the claim that GP performed better in monthly trading than GE on the three monthly splits M02, M03 and N_M01 (using 90% confidence), and, in the opposite direction, the hypothesis that GE performed better than GP in N_M03 was backed up by the results of the associated T-Test. However, results were inconclusive for the remaining four data splits M01, M04, N_M02 and N_M04.

GE versus GP									
#	Objectives	Return (\$)							
		M01	M02	M03	M04	N_M01	N_M02	N_M03	N_M04
1	Single Objective	GE							
1.1	GE-MR	1755.373	2079.598	1842.959	3904.714	1700.288	2614.749	2785.087	3334.674
1.2	GE-PC_LK12	1684.526	1550.536	1253.849	2629.755	1658.197	2329.447	2033.352	2375.616
1.3	GE-PC_LK24	1720.779	1852.604	1437.367	3553.837	1196.123	2481.912	1944.386	3007.685
2	Single Objective	GP							
2.1	GP-MR	1763.835	2214.391	2030.462	3936.528	1862.515	2492.902	2348.802	3171.527
2.2	GP-PC_LK12	1665.645	1525.763	1171.552	2247.561	1775.168	2335.074	1943.946	2100.316
2.3	GP-PC_LK24	1740.363	1708.306	1315.774	3634.667	1107.977	2490.896	1971.084	2781.237

Table 5-13: Monthly trading results of GE versus GP.

Period	Best Method	Configurations to compare with the Best Method using one-tailed paired T-Test					
M01	GP-MR 1763.834845	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-PC_LK12	GP-PC_LK24
		mean	1755.372602	1684.526498	1720.779094	1665.645028	1740.362935
		p-value	0.152816653	2.38E-05	0.029467702	5.58822E-08	0.056056674
M02	GP-MR 2214.390803	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-PC_LK12	GP-PC_LK24
		mean	2079.598237	1550.535708	1852.60437	1525.763477	1708.30581
		p-value	0.049514112	1.11E-08	8.8821E-05	4.47557E-10	2.49843E-06
M03	GP-MR 2030.461758	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-PC_LK12	GP-PC_LK24
		mean	1842.95934	1253.849272	1437.367478	1171.551724	1315.7742
		p-value	0.050011994	7.80E-09	1.20815E-05	3.3647E-09	7.1553E-07
M04	GP-MR 3936.527613	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-PC_LK12	GP-PC_LK24
		mean	3904.714129	2629.755143	3553.837446	2247.561093	3634.666662
		p-value	0.43910497	7.53E-08	0.067213393	4.27093E-08	0.07137425
N_M01	GP-MR 1862.514801	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-PC_LK12	GP-PC_LK24
		mean	1700.288292	1658.19719	1196.123253	1775.167739	1107.976716
		p-value	0.054688004	1.50E-02	2.01302E-06	0.09056039	3.03556E-09
N_M02	GE-MR 2614.749356	config	GE-PC_LK12	GE-PC_LK24	GP-MR	GP-PC_LK12	GP-PC_LK24
		mean	2329.446573	2481.911537	2492.901991	2335.07424	2490.895893
		p-value	2.46E-02	0.193990935	0.208392232	0.027586846	0.132678823
N_M03	GE-MR 2785.086949	config	GE-PC_LK12	GE-PC_LK24	GP-MR	GP-PC_LK12	GP-PC_LK24
		mean	2033.352232	1944.386145	2348.802023	1943.945833	1971.083524
		p-value	5.13E-05	0.000285297	0.009858024	5.34857E-06	0.000354414
N_M04	GE-MR 3334.673761	config	GE-PC_LK12	GE-PC_LK24	GP-MR	GP-PC_LK12	GP-PC_LK24
		mean	2375.616494	3007.685203	3171.526994	2100.315969	2781.237415
		p-value	6.36E-04	0.116216069	0.271304724	5.6663E-05	0.026302795

Table 5-14: Paired one-tailed T-Test on the hypothesis that the best method is the method with highest average return for each monthly data splits. Each row displays mean and p-value for each configuration, and if p-value > 0.10, then it is highlighted with grey.

In the case of weekly trading, the results are summarised in Table 5-15 and the statistical tests are summarised in Table 5-16. Here, we find statistical significance in the better performance of GP on splits W02, W04 and N_W02, but there were no statistically significant differences between performance of GP and GE for the weekly trading splits W01, W03, N_W01, N_W03 and N_W04.

GE versus GP									
#	Objectives	Return (\$)							
		W01	W02	W03	W04	N_W01	N_W02	N_W03	N_W04
1	Single Objective	GE							
1.1	GE-MR	1458.639	1461.982	1633.585	1204.277	874.9361	1598.309	2022.828	1496.991
1.2	GE-PC_LK12	1396.324	1398.741	1473.506	1087.451	876.0133	1470.324	2030.001	1160.286
1.3	GE-PC_LK24	1275.205	1369.821	1452.096	1121.527	720.0218	1515.025	1633.354	1009.192
2	Single Objective	GP							
2.1	GP-MR	1440.849	1565.496	1642.733	1254.492	935.3488	1641.875	2005.712	1478.938
2.2	GP-PC_LK12	1384.502	1422.879	1552.805	1002.48	781.0203	1499.37	1941.13	1154.695
2.3	GP-PC_LK24	1259.466	1425.196	1498.664	1074.237	715.7163	1470.2	1943.249	977.7959

Table 5-15: Weekly trading results of GE versus GP.

Period	Best Method	Configurations to compare with the Best Method using one-tailed paired T-Test					
W01	GE-MR	config	GE-PC_LK12	GE-PC_LK24	GP-MR	GP-PC_LK12	GP-PC_LK24
		mean	1396.324231	1275.204677	1440.849036	1384.502144	1259.466289
	1458.638783	p-value	2.52E-03	3.10467E-06	0.263475087	0.008232221	7.37623E-07
W02	GP-MR	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-PC_LK12	GP-PC_LK24
		mean	1461.981811	1398.741251	1369.821067	1422.879238	1425.195607
	1565.496149	p-value	0.005625084	1.62E-04	0.00020799	0.000536915	0.00074962
W03	GP-MR	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-PC_LK12	GP-PC_LK24
		mean	1633.584772	1473.505759	1452.09558	1552.804579	1498.663726
	1642.733456	p-value	0.352630692	2.03E-03	0.000208914	0.02454975	0.0015559
W04	GP-MR	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-PC_LK12	GP-PC_LK24
		mean	1204.276892	1087.451189	1121.52683	1002.479523	1074.237428
	1254.49184	p-value	0.086338126	5.84E-03	0.003764856	0.000107801	0.002217061
N_W01	GP-MR	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-PC_LK12	GP-PC_LK24
		mean	874.9361396	876.0132813	720.0218498	781.0202642	715.7162659
	935.3488047	p-value	0.15501006	1.82E-01	0.000600343	0.017775053	0.000156888
N_W02	GP-MR	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-PC_LK12	GP-PC_LK24
		mean	1598.309463	1470.324455	1515.025393	1499.369692	1470.199522
	1641.874987	p-value	0.091351921	8.13E-04	0.001284454	0.002064254	0.000481412
N_W03	GE-PC_LK12	config	GE-MR	GE-PC_LK24	GP-MR	GP-PC_LK12	GP-PC_LK24
		mean	2022.828179	1633.354389	2005.712205	1941.129929	1943.248828
	2030.001471	p-value	0.447520672	3.31259E-06	0.377987619	0.156067017	0.097733896
N_W04	GE-MR	config	GE-PC_LK12	GE-PC_LK24	GP-MR	GP-PC_LK12	GP-PC_LK24
		mean	1160.285976	1009.192431	1478.938414	1154.694684	977.7958652
	1496.99081	p-value	1.42E-05	4.96827E-08	0.333016932	0.00066132	3.75064E-07

Table 5-16: Paired one-tailed T-Test on the hypothesis that the best method is the method with highest average return for each weekly data splits. Each row displays mean and p-value for each configuration, and if p-value > 0.10, then it is highlighted with gray colour.

Turning now to the case of daily trading, for which the results are summarised in Table 5-17 and the T-Tests are given in Table 5-18, we find there is a tie between the performance of GE and GP. GP beat GE in two splits (D02 and N_D03) while GE could outperform GP in two other data splits, D03 and N_D02, and there were no statistically significant differences in GE and GP daily trading performance in the remaining four daily periods (D01, D04, N_D01 and N_D04).

GE versus GP									
#	Objectives	Return (\$)							
		D01	D02	D03	D04	N_D01	N_D02	N_D03	N_D04
1	Single Objective	GE							
1.1	GE-MR	751.9025	926.0272	990.3231	871.3754	1092.385	1008.635	950.7187	1008.829
1.2	GE-PC_LK12	752.1372	938.5457	953.8975	886.3768	1030.75	1014.408	935.4551	964.2883
1.3	GE-PC_LK24	752.4865	917.3178	1003.682	864.72	1057.521	1019.301	950.9823	960.586
2	Single Objective	GP							
2.1	GP-MR	750.8226	924.8952	984.5875	874.7393	1092.446	1009.926	956.7973	1005.624
2.2	GP-PC_LK12	742.5342	939.5055	933.9582	876.7977	1032.984	1005.159	924.5394	948.6596
2.3	GP-PC_LK24	752.8806	926.2618	992.0299	877.02	1066.217	1002.273	968.2754	939.742

Table 5-17: Daily trading results of GE versus GP.

Period	Best Method	Configurations to compare with the Best Method using one-tailed paired T-Test					
D01	GP-PC_LK24	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-MR	GP-PC_LK12
		mean	751.9024999	752.1372484	752.4865116	750.8226341	742.5341921
		752.8806005	p-value	0.164938398	1.65E-01	0.164938395	0.1649384
D02	GP-PC_LK12	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-MR	GP-PC_LK24
		mean	926.0271746	938.5457255	917.3178495	924.8952332	926.2618071
		939.5054612	p-value	0.029857243	4.45E-01	0.01606409	0.01601872
D03	GE-PC_LK24	config	GE-MR	GE-PC_LK12	GP-MR	GP-PC_LK12	GP-PC_LK24
		mean	990.323098	953.8975309	984.5874623	933.9581716	992.0299301
		1003.682076	p-value	4.66671E-05	4.64E-06	0.009127846	1.40236E-08
D04	GE-PC_LK12	config	GE-MR	GE-PC_LK24	GP-MR	GP-PC_LK12	GP-PC_LK24
		mean	871.3754269	864.720016	874.7393067	876.7977476	877.0200257
		886.3768202	p-value	0.056680235	0.018265683	0.152287949	0.184296062
N_D01	GP-MR	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-PC_LK12	GP-PC_LK24
		mean	1092.385464	1030.749514	1057.520827	1032.983983	1066.217103
		1092.446401	p-value	0.491331516	2.39E-09	0.000223007	1.19758E-08
N_D02	GE-PC_LK24	config	GE-MR	GE-PC_LK12	GP-MR	GP-PC_LK12	GP-PC_LK24
		mean	1008.634956	1014.408028	1009.926274	1005.158727	1002.273119
		1019.30102	p-value	0.028738181	2.14E-01	0.049249172	0.086191982
N_D03	GP-PC_LK24	config	GE-MR	GE-PC_LK12	GE-PC_LK24	GP-MR	GP-PC_LK12
		mean	950.7187002	935.4550617	950.9823258	956.7973255	924.5394471
		968.2754102	p-value	4.31497E-05	1.78E-04	0.00533851	0.014782791
N_D04	GE-MR	config	GE-PC_LK12	GE-PC_LK24	GP-MR	GP-PC_LK12	GP-PC_LK24
		mean	964.2883374	960.5859647	1005.624097	948.6595942	939.7420284
		1008.828765	p-value	5.53E-03	0.001885892	0.295500613	0.00011882

Table 5-18: Paired one-tailed T-Test on the hypothesis that the best method is the method with highest average return for each daily data splits. Each row displays mean and p-value for each configuration, and if p-value > 0.10, then it is highlighted with gray colour.

In addition to single-objective trading experiments on monthly, weekly and daily data, we have tried our new GE approach in conjunction with some of the multi-objective configurations described in section 4.2.2.1, concentrating on profit driven approaches, and the results indicate a similar situation to the single-objective trading results summarised above. Overall, standard GP outperforms our new GE approach in

multi-objective financial trading. In consequence, to avoid increasing the length of this thesis, we have decided to omit the results.

5.10 Concluding Discussion

In this chapter we have proposed a new approach to GE in which the major changes are in the genome mapping process and in the genetic operators. The main concept driving our new approach is based on how to reduce the occurrences of invalid genomes resulting from the incomplete mapping processes, whether this arises from randomly generated individuals or from the result of genetic operations on valid individuals. Starting with attempts to repair invalid genomes with three different methods called *Valid Map* methods, the effect of using these repair methods in the GE mapping process delivered substantial improvement on tackling six test benchmark problems of three kinds (Symbolic Regression, Symbolic Integration and Santa Fe Ant trial), as we saw from the results in section 5.5. Having said that, on some problems the performance of GE with valid map repair methods were still not comparable with standard GP since there were too many invalid genomes generated via applying the standard genetic operators. Though using a valid map method can reduce the number of invalid genomes by repairing them, normal genetic operators (both crossover and mutation) for binary (integer) strings can have an adverse impact on GE performance. Exchanging genetic materials in one-point or two-point crossover between two parents of GE binary genomes to produce two new individuals is often similar to jumping to a very different solution in a new area of the search space, because the new exchanged material in the offspring can be interpreted very differently in the mapping process from how it was interpreted in its original parent. This means that standard operators used in the context of GE perhaps cause more *exploration* than *exploitation*, and in particular crossover in GE is more exploratory than crossover in GP. Moreover, invalid genomes can occur in

the crossover phase due to the new material in a child genome leading to incomplete mapping, and this is also true in case of mutation. To conclude, normal crossover that works in a highly exploratory way can lead to inefficient search, and both normal crossover and normal mutation can bring about too many invalid genomes. The new approach to GE genetic operators, typed genetic operators, was been introduced in section 5.7 in an attempt to overcome these problems. Using the concept of a valid token, typed crossover for GE works more in an exploitation mode, being similar to sub-tree crossover in GP, maintaining a proper balance between exploration and exploitation in the GE system. If a configuration is used at which typed operators are used 100% of the time, there is a guarantee that no invalid genomes are generated. The results from section 5.8 told us that combining the repair strategy and the typed genetic operators had a beneficial impact on the performance of GE, performing better than standard GP in ANT, SR1 and SR4 problems. However when we applied the new GE approach in the context of financial trading, the overall performance of GE in monthly trading and weekly trading was outperformed by GP, while in case of daily trading the conclusion seems indeterminate. Although GE seems less applicable than GP in the cases of monthly trading and weekly trading, it should be noted that the GE configuration used for financial trading in section 5.9 has some limitations and more investigation may find GE configurations more well-suited. First, the combination between other valid map methods (MAP01 and MAP02) and typed genetic operators may be worth further testing. Second, referring to Table 5-10, additional configurations with more mixing of different values of typed crossover rates and typed mutation rates could be examined more as well. Lastly, the GE configuration we used in the trading experiments was in some ways set to be similar to the GP configuration, in attempt to achieve at least similarly good performance. Alternative configurations which include more exploratory settings might be appropriate, especially in the case of daily trading.

In summary, we have shown evidence that our new GE approach has the ability to solve some problems with comparable or better performance than standard GP, and our results also strongly suggest that the use of valid map methods and typed GE operators leads to results that can be significantly better than standard GE.

6. Conclusion

6.1 Summary

Genetic Programming (GP) has been actively utilized as a tool to make profits in the financial markets over the past decade, and the application of GP in evolving technical trading rules has been actively explored with various levels of success. Technical trading rules are used in the Buy-and-Sell (B&S) strategy to generate the right trading signals to earn potential profits. However, normally successful trading rules found during the training phase quite often produce disappointing performance on unseen data due to the overfitting problem, and they are therefore not competitive with the straightforward strategy of Buy-and-Hold (B&H) in upwardly moving markets or investing in risk-free assets in down-trending markets. Our work in Chapter 3 aimed to overcome this difficulty. With variations to the approach used in Becker and Seshadri's work involving more extensive investigation on more single-objective methods in various different trading environments, the results of the experiments in the context of the single-objective methodology in Chapter 3 indicated that we were able to identify the regime that outperforms buy-and-hold with considerably robust performance in the case of monthly trading and with fairly robust performance in the case of weekly trading (however poor performance in case of daily trading). And, this work has served as an introductory study for future exploration of the guidelines for generating robust technical trading rules which may be able to outperform B&H. Moreover, we also noticed that considering risk measures in the process of fitness evaluation seems to be a vital factor to bring about substantial improvement in the robustness and effectiveness of the method for trading frequently in volatile markets, as is the case in weekly and daily trading.

Based on work in Chapter 3, in Chapter 4 we extended our investigation to find out whether we can deliver improved performance in the context of weekly and daily trading from the multi-objective configurations by applying the NSGA-II algorithm to GP to be used as a main tool in this chapter. The results in this chapter have successfully shown that multi-objective approaches, particularly when a majority voting approach is used, are able to generally generate comprehensible trading rules that outperform single-objective approaches and robustly outperform buy and hold trading and risk-free investment in monthly trading and also weekly trading. Also, it should be observed that trading with the majority voting technique performed consistently better than the alternative. In the last chapter, we introduced a new approach to GE with repair strategies, referred to as Valid Map methods, which deal with invalid genomes in the mapping process, and a new set of typed genetic operators, and we also demonstrated the effectiveness of this new approach through empirical tests on various problems, including financial trading. The test results revealed convincing evidence supporting satisfactory performance of the new GE approach over standard GE on most of the test problems.

6.2 Contributions

A list of contribution of this thesis was given in section 1.3 - we provide a review of that list here, with some extra discussion.

1.6.1 Contributions to Computer Science

1. We contribute proper practice using a validation set (a three-data-sets methodology) for model selection to choose the rules, and use varied data spits to test the robustness and sensitivity to the data of the rules. This proper practice was accounted mainly in Chapter 3 and Chapter 4 to evolve technical trading rules with both single-objective and multi-objective approaches. Previously, a two data set

methodology, with a first data set (training set) used to train the learning algorithm and a second data set (test set) with unseen data used to estimate the performance of the algorithm, which has been widely accepted and applied. However, this methodology is vulnerable to over-fitting, and should be avoided in future research as the learning algorithm using this methodology can evolve a model that fits the data set too closely in the training period, basically fitting noise in the training set that leads to poor performance on unseen data. To help the learning algorithm stop evolving the model before it has started to over-fit, the third data set (validation set) is added to interrupt the learning algorithm and bring about the best-of-run solutions with good generalization performance. Having said that, it should be clear that the use of a validation set can have the drawback of discarding a significant amount of data from the training set, which can be harmful to the learning process [41]. We have shown that proper practice based on a three-data-sets methodology has generalization capability, supported by robust performance in case of monthly and weekly trading for both single-objective and multi-objective approaches, and this proper practice can be used as a basic setup to solve other kinds of real-life problems.

2. We contribute general lesson related to the significant open issue of generalization, which indicated that the unpromising results of previous attempts to evolve profitable trading rules was most likely due to poor generalization caused by lack of proper experimental design. This fact has been backed up by the results in Chapter 3, which showed that with proper practice, mentioned in the first contribution above, we can find trading rules that perform consistently and robustly in the cases of monthly and weekly data from both types of single and multi objective configurations, and this was contradictory to the basic setup used in [6, 8]

which we found was sensitive to the data periods involved, which in turn is further evidence of over-fitting in that setup.

3. We contribute evidence that the multi-objective methodology, which avoids over-fitting by spreading functional complexity of the solutions throughout different expressions of each objective, can be used to increase the level of generalization performance. This was shown mainly in Chapter 4, and was supported by the results from the experiments, indicating that multi-objective approaches, especially when using a majority voting approach, are able to generally outperform single-objective approaches in the case of monthly trading and more clearly in the case of weekly trading. In addition, the results also revealed that in both types of profit driven and risk-adjusted approaches, the worst performance for monthly and weekly cases came from the configurations within the ‘basic group’, which comprised the most straightforward multi-objective versions of our single-objective approaches. However, when augmenting the multi-objective methodology by combining the corresponding single objective methods as part of a multi-objective configuration, this seemed to bring about a robust and effective configuration.
4. We contribute additional evidence that making appropriate parameter choices can lead to a successful GP application even in a dynamic problem environment, and this setup has been done in terms of a dynamic form of GP by using the dynamic forms of mutation (4 different mutation forms) during a single GP run. This dynamic form was used in all experiments of Chapter 3 and Chapter 4 to evolve technical trading rules with both single-objective and multi-objective approaches. Using GP to solve problems, GP users have many open choices to make decisions such as the form and rate of mutation, the form and rate of crossover, the size of population, and so on. All these settings can have significant impact on GP’s performance, especially when running in dynamic environments. In fact,

appropriate settings tend to change dynamically during the evolutionary run [109]. In consequence, the use of adaptive settings, in which settings can adapt dynamically during the GP run and form the dynamic form of GP, may give rise to better performance than using static settings, and the promising trading results in the dynamic environment of financial markets from Chapter 3 and Chapter 4 have supported this assumption.

5. We contribute a new approach to grammatical evolution that allows us to find GE configurations that are effective in solving various kinds of problems including the trading context (unlike standard GE). This new approach was introduced in Chapter 5 with an empirical study of genome mapping methods, repairing invalid genome strategies, and a new suite of typed genetic operators (typed mutation and typed crossover) with capability that effectively allow us to vary a GE search between GE and GP style, by changing the rates of applying these two operators. We have shown evidence to back up the claim that our new GE approach is effective in comparison with standard GP and standard GE, in testing on a range of standard GE and GP test functions.

1.6.2 Contributions to Finance

1. We contribute a new and thorough evaluation of the capability of genetic programming to evolve profitable technical trading rules that can outperform a buy-and-hold strategy. This was described mainly in Chapter 3. We made extensive investigations based on the promising work of Becker and Seshadri, and investigated more single-objective configurations in several different trading environments. The results of our experiments indicated that the approach used in Becker and Seshadri's work was sensitive to the data set, and by making radical alterations to Becker and Seshadri's approach including the use of a

validation set to choose the rule, we discovered the robust conditions in which GP-evolved technical trading rules may be able to outperform the buy-and-hold strategy for monthly and weekly trading. We also carried out more extensive investigation with different single objectives, classified into two approaches: profit driven and risk-adjusted approaches, and we have shown that incorporating risk factors in the fitness model can lead to reliable and effective trading rules when applying them in high varying conditions of weekly and daily data. In addition, the work in Chapter 3 has served as a basic guideline of evolving robust trading rules for future research as well.

2. We contribute the development and evaluation of several multi-objective approaches to evolving technical trading rules with genetic programming. This was accounted mainly in Chapter 4. By extending work from Chapter 3 from the single to the multi-objective context, comprehensive experiments were provided in Chapter 4 with the comparison of performance to single-objective approaches, and the results showed that we successfully found subsets of multi-objective configurations from the group of profit driven approaches that produced robust and successful trading rules for monthly and weekly trading. Especially interesting was the case of weekly trading, in which performance was distinctly superior to the single-objective approaches. Moreover, we also noticed that methods focusing on profit only could perform efficiently in monthly and weekly trading, while integrating risk to the methods could produce better performance in the case of daily trading.
3. We contribute supporting evidence to back up the prediction ability of technical analysis and therefore add to the criticism surrounding the Random Walk and Efficient Market Hypotheses. This was supported from the results from Chapter 3 and Chapter 4.

4. We contribute some insights into trading strategies that are appropriate for different trading environments by analysing the technical trading rules that were generated from multiple experiments with both the single and multi-objective approaches. This was done mainly in sections 3.5 and 4.5. Furthermore, an interesting point emerged that we might have found interesting new and simple rules that perform effectively in certain market conditions, and we discussed this issue in section 3.6.

6.3 Future Work

Additional directions for future research are suggested in the following list.

1. Achieving beneficial technical trading rules from evolutionary algorithms is very much dependent on the choice of fitness model, and the use of a more sophisticated risk-adjusted fitness model with successful robust trading regardless of market conditions (upward or downward markets) has been reported in [29]. In consequence, it is very interesting and challenging to investigate whether the use of sophisticated risk-adjusted fitness models such as the Conditional Sharpe Ratio with the experimental regime provided in this thesis on both single-objective and multi-objective approaches can increase the level of generalization and therefore result in impressive performance on unseen data.
2. The use of a new set of technical indicators, by reducing the use of less active indicators indicated by the results of the rule analyses in this thesis, would reduce the search space, while adding new high-level, generally promising technical indicators (as suggested from our level 2 rule analyses at the ends of Chapters 3 and 4) could again usefully reduce the search space and improve the comprehensibility of the evolved technical trading rules, and possibly improve trading results.

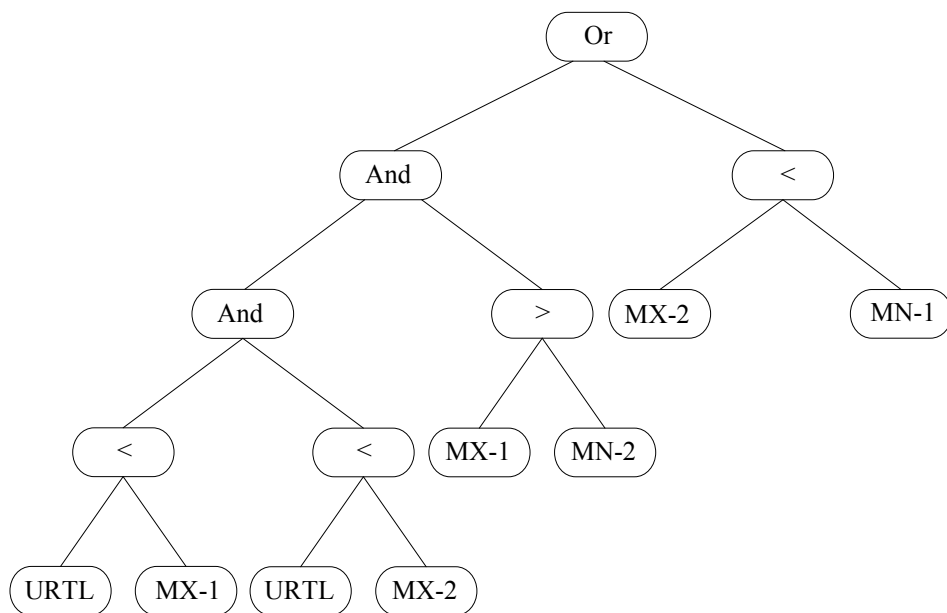
3. The use of adaptive rates of crossover and mutation [109] in GP or GE during a single run should be investigated more for the trading problem, to see whether these adaptive rates, combined with dynamic forms of mutation, can deliver better performance when working in the dynamic environment of the financial markets.
4. The majority vote technique of multi-objective trading in Chapter 4 should be explored more to achieve a better understanding about it in future work.
5. With the successful results reported in [7, 8], the use of co-evolution in technical trading rules between a specialized buy rule and a specialized sell rule should be investigated with both single and multi-objective configurations.
6. Testing of the new GE approach with multi-objective configurations on trading problems should be done with more different GE configurations and setups such as the use of other repair methods (MAP01 and MAP02) and more parameter configuration, to vary a GE search between GE and GP styles.
7. The promising approach of several recent research efforts in considering stocks of individual companies, rather than composite market indices such as S&P500 and Dow Jones, should be investigated, and the use of the moving window technique for data splits should be included in the future experiments as well.

Appendix A

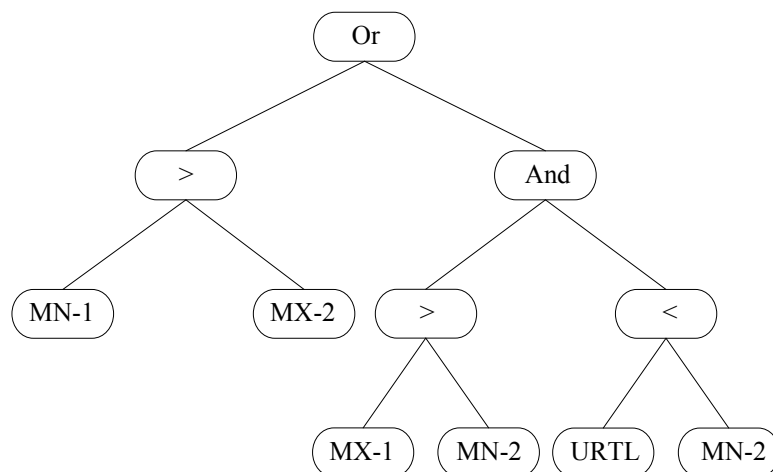
A.1 Samples of Technical Trading Rules for Single-objective Approach

A.1.1 Monthly Trading

A.1.1.1 Profit Driven Approach: MR

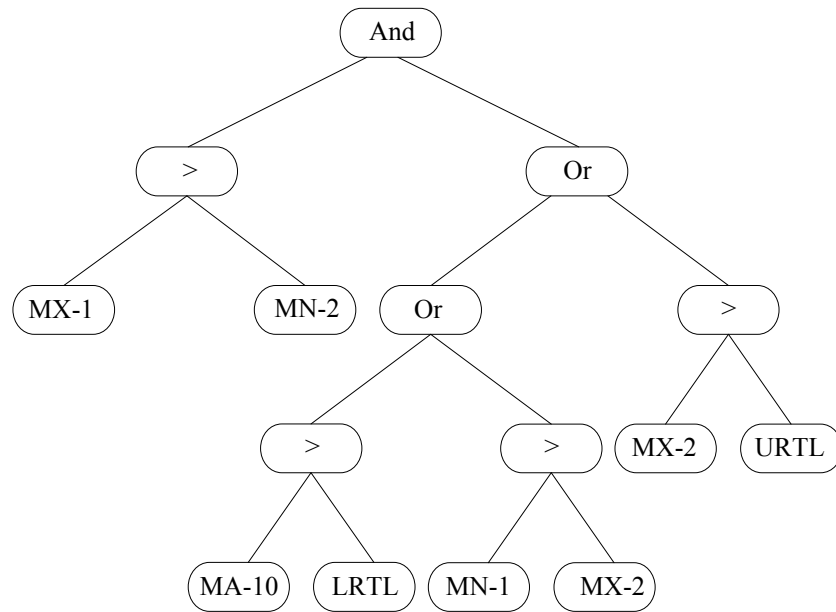


A.1.1.2 Risk-Adjusted Approach: SHARO

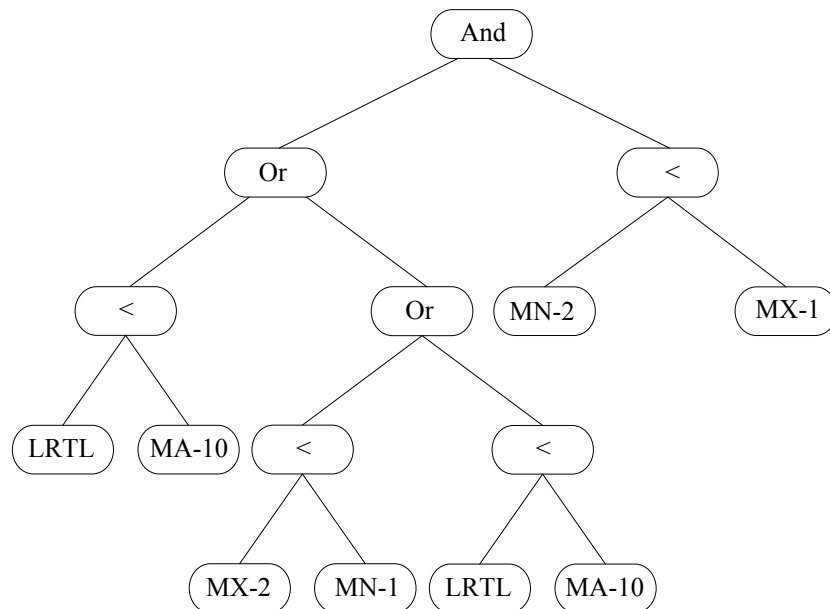


A.1.2 Weekly Trading

A.1.2.1 Profit Driven Approach: MR

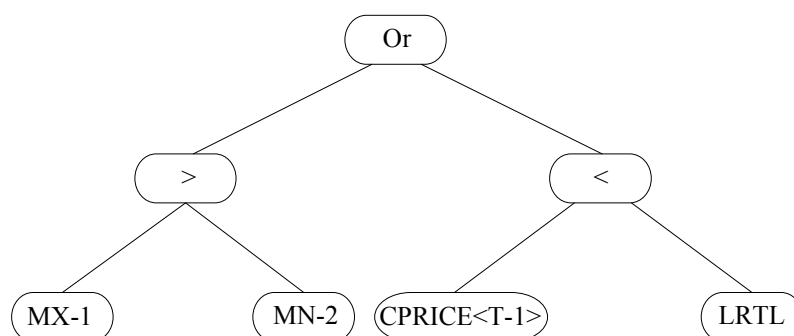


A.1.2.2 Risk-Adjusted Approach: SHARO

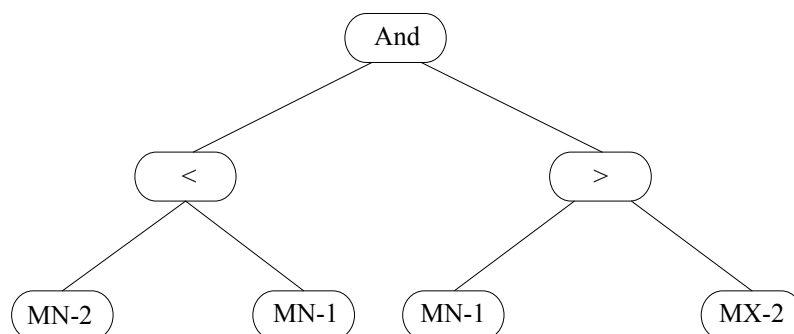


A.1.3 Daily Trading

A.1.3.1 Profit Driven Approach: MR



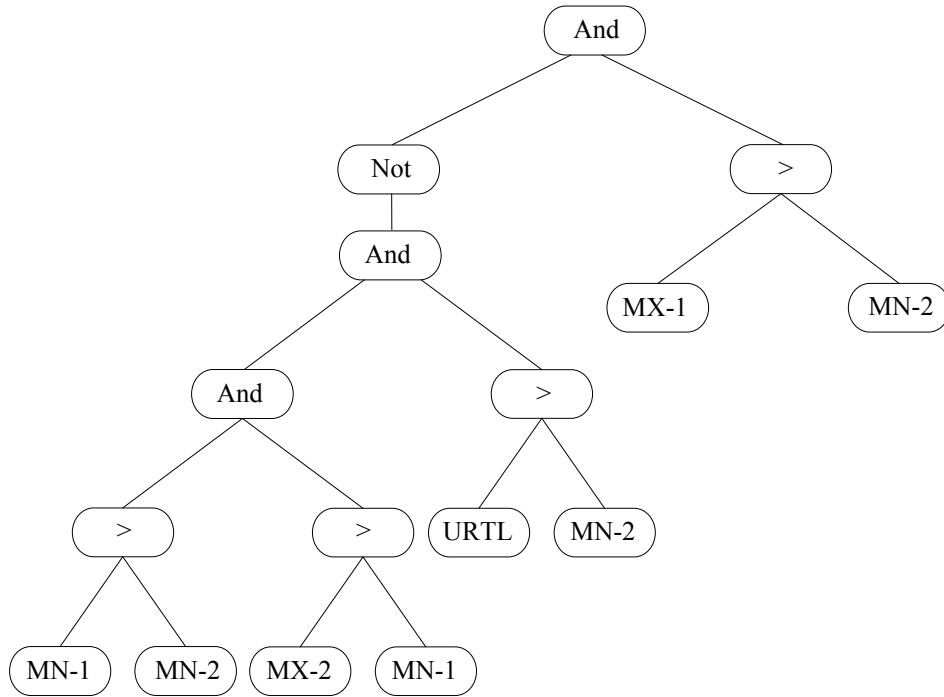
A.1.3.2 Risk-Adjusted Approach: MSTLRO



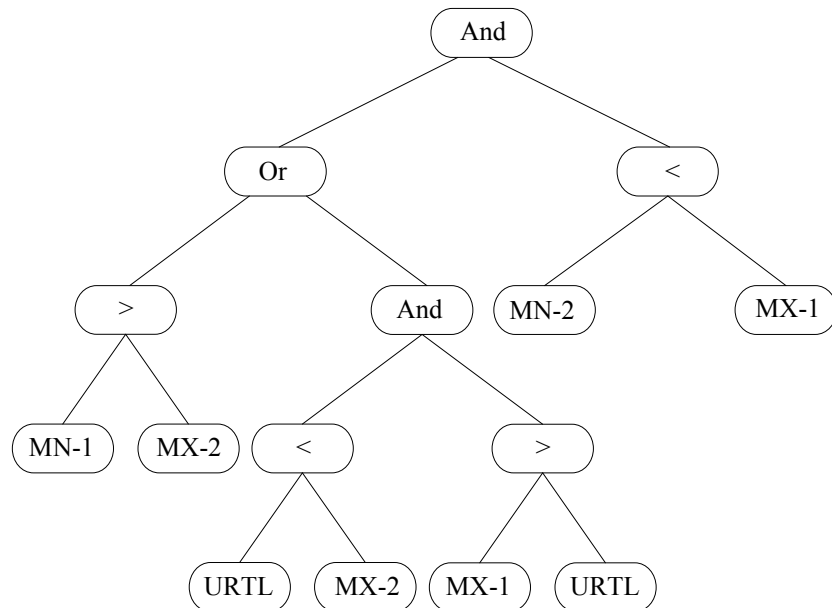
A.2 Samples of Technical Trading Rules for Multi-objective Approach

A.2.1 Monthly Trading

A.2.1.1 Profit Driven Approach: 2MR-CXP

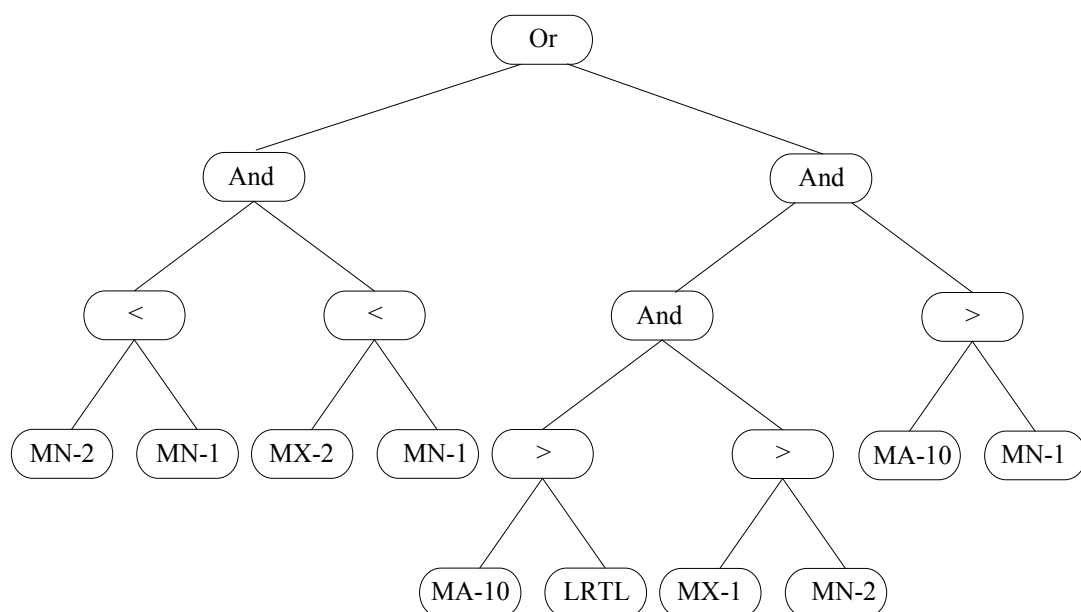


A.2.1.2 Risk-Adjusted Approach: CMR_CPC-LK24_SHARO

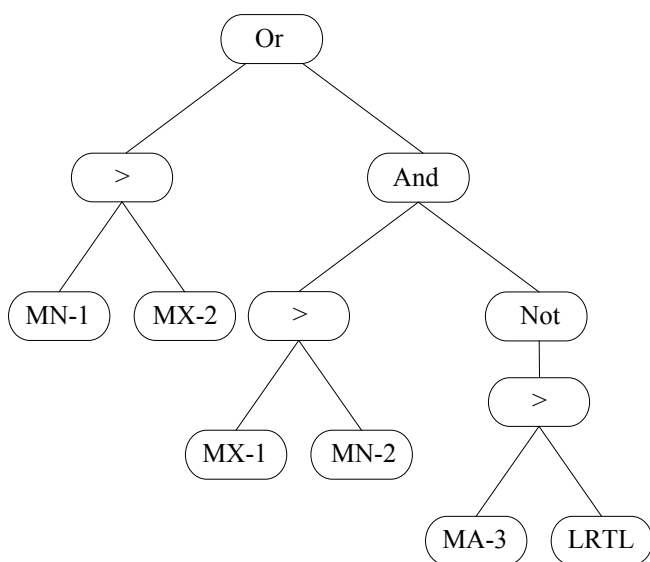


A.2.2 Weekly Trading

A.2.2.1 Profit Driven Approach: MR_PC-LK12_CXP

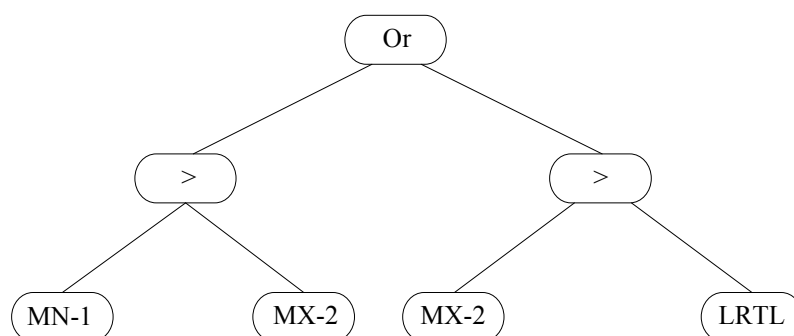


A.2.2.2 Risk-Adjusted Approach: CMR_CPC-LK12_SHARO

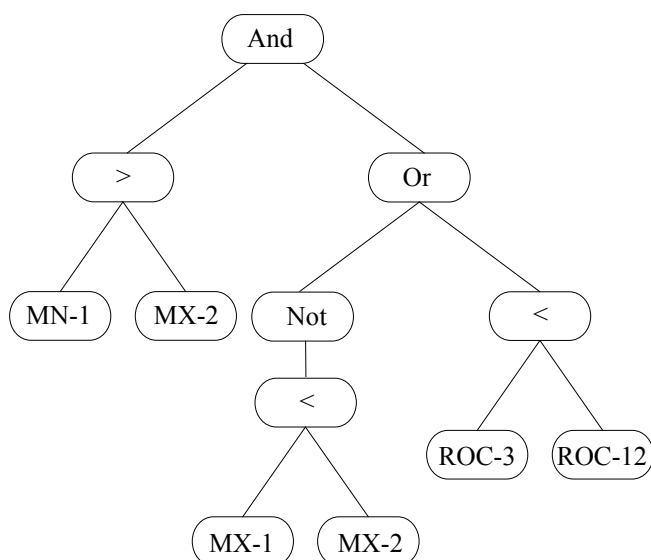


A.2.3 Daily Trading

A.2.3.1 Profit Driven Approach: *MR_PC-LK12_CXP*



A.2.3.2 Risk-Adjusted Approach: *MSTLRO_CXP*



Bibliography

- [1] Alexander, S. S. (1964). "Price movements in speculative markets: trends or random walks." *Proc., The random character of stock market prices*, MIT Press, 338-372.
- [2] Allen, F., and Karjalainen, R. (1999). "Using genetic algorithms to find technical trading rules." *J. Financ. Econ.*, 51(2), 245-271.
- [3] Angeline, P. J. (1996). "Genetic Programming's Continued Evolution." *Advances in genetic programming*, P. J. Angeline, and K. E. Kinnear, eds., MIT Press, Cambridge, MA, 89-110.
- [4] Banzhaf, W., Nordin, P., Keller, R., and Francone, F. (1998). *Genetic programming : an introduction: On the Automatic Evolution of Computer Programs and Its Applications*, Morgan Kaufmann, San Francisco.
- [5] BDS (2010). "Biocomputing and Developmental Systems (BDS)." <<http://bds.ul.ie/>>. (February, 2010).
- [6] Becker, L. A., and Seshadri, M. (2003). "Comprehensibility and Overfitting Avoidance in Genetic Programming for Technical Trading Rules." Worcester Polytechnic Institute, Computer Science Technical Report WPI-CS-TR-03-09.
- [7] Becker, L. A., and Seshadri, M. (2003). "Cooperative Coevolution of Technical Trading Rules." Worcester Polytechnic Institute, Computer Science Technical Report WPI-CS-TR-03-15.
- [8] Becker, L. A., and Seshadri, M. (2003). "GP-evolved technical trading rules can outperform buy and hold." *Proceedings of the 7th Joint Conference on Information Sciences*, S. H. Chen, H. D. Cheng, D. K. Y. Chiu, S. Das, R. Duro, E. E. Kerre, H. V. Leong, Q. Li, M. Lu, M. G. Romay, D. Ventura, and J. Wu, eds., Assoc Intelligent Machinery, Durham, 1136-1139.
- [9] Bessembinder, H., and Chan, K. (1995). "The Profitability of Technical Trading Rules in the Asian Stock Markets." *Pacific Basin Finance Journal*, 7, 257-284.

- [10] Brabazon, A., and O'Neill, M. (2004). "Evolving technical trading rules for spot foreign-exchange markets using grammatical evolution." *Computational Management Science*, 1(3-4), 311-327.
- [11] Brabazon, A., and O'Neill, M. (2005). *Biologically Inspired Algorithms for Financial Modelling (Natural Computing Series)*, Springer, New York.
- [12] Briza, A. C., C., P., and Jr., N. (2008). "Design of stock trading system for historical market data using multiobjective particle swarm optimization of technical indicators." *GECCO*, 1871–1878.
- [13] Brock, W., Lakonishok, J., and Lebaron, B. (1992). "Simple Technical Trading Rules and the Stochastic Properties of Stock Returns." *J. Financ.*, 47(5), 1731-1764.
- [14] Chang, T. J., Meade, N., and Beasley, J. E. (2000). "Heuristics for Cardinality Constrained Portfolio Optimization." *Computers and Operations Research*, 27(13), 1271–1302.
- [15] Chen, S. H. (2002). *Genetic Algorithms and Genetic Programming in Computational Finance*, Kluwer, Boston, MA.
- [16] Chen, S. H., and Yeh, C. (1996). "Toward a Computable Approach to the Efficient Market Hypothesis: An Application of Genetic Programming." *J. of Econ Dynamics & Control*, 21, 1043-1063.
- [17] Cheng, S. L., and Khai, Y. L. (2002). "GP-Based Optimisation of Technical Trading Indicators and Profitability in FX Market." *Proc. 9th ICONIP*, 1159-1163.
- [18] Chiam, S. C., Tan, K. C., and Mamun, A. A. (2009). "Investigating technical trading strategy via a multi-objective evolutionary platform." *Expert Systems with Applications*, 36(7), 10408–10423.
- [19] Coello Coello, C. A., Lamont, G. B., and Veldhuizen, D. A. v. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer.

- [20] Coello., C. A. (2006). *Evolutionary Multi-Objective Optimization and its Use in Finance*, Idea Group Publishing.
- [21] De Jong, K. A. (1975). "An analysis of the behavior of a class of genetic adaptive systems." PhD Thesis, University of Michigan, Ann Arbor, MI, USA.
- [22] Deb, K. (2008). *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester.
- [23] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II." *Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI)*, 849-858.
- [24] Dempster, M. A. H., and Jones, C. M. (2000). "A real-time adaptive trading system using genetic programming." *Quantitative Finance*, 1, 397-413.
- [25] Doerner, K., Gutjahr, W. J., Hartl, R. F., Strauss, C., and Stummer, C. (2004). "Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection." *Annals of Operations Research*, 131((1-4)), 79-99.
- [26] Ehrgott, M., Klamroth, K., and Schwehm, C. (2004). "An MCDM approach to portfolio optimization." *European Journal of Operational Research*, 155(3), 752-770.
- [27] Eiben, A. E., and Smith, J. E. (2007). *Introduction to evolutionary computing*, Springer.
- [28] Esfahanipour, A., Karimi, B., and Mousavi, S. (2009). "The Profitability of Technical Trading Rules in Tehran Stock Exchange: An Application of Genetic Programming." *International Symposium on INnovations in Intelligent SysTems and Applications (INISTA 2009)*Karadeniz Technical University, Trabzon, Turkey.
- [29] Esfahanipour, A., and Mousavi, S. (2011). "A genetic programming model to generate risk-adjusted technical trading rules in stock markets." *Expert Systems with Applications*, 38(7), 8438-8445.

- [30] Fama, E. F. (1965). "Random walks in stock market prices." *Financial Analysts Journal*, 21, 55-59.
- [31] Fama, E. F. (1970). "Efficient capital markets: A review of theory and empirical work." *J. Financ.*, 25(2), 383-423.
- [32] Fama, E. F. (1991). "Efficient capital markets: II." *J. Financ.*, 46, 1575-1617.
- [33] Fama, E. F. (1998). "Market efficiency, long-term returns, and behavioral finance." *J. Financ. Econ.*, 49, 283-306.
- [34] Fama, E. F., and Blume, M. (1970). "Filter Rules and Stock-Market Trading." *Journal of Business*, 39, 226-241.
- [35] Farnsworth, G. V., Kelly, J. A., Othling, A., and Pryor, R. (2004). "Successful Technical Trading Agents Using Genetic Programming." Sandia National Laboratories.
- [36] Fogel, L. J., Owens, A. J., and Walsh, M. J. (1965). "Artificial Intelligence through a simulated of evolution." *Biophysics and Cybernetic Systems* Spartan, Washington, DC, 131-156.
- [37] Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). *Artificial Intelligence through Simulated Evolution*, John Wiley, New York, USA.
- [38] Fonseca, C. M., and Fleming, P. J. (1995). "An overview of evolutionary algorithms in multi-objective optimization." *Evolutionary Computation Journal*, 3(1), 1-6.
- [39] Francone, F. D., Conrads, M., Banzhaf, W., and Nordin, P. (1999). "Homologous Crossover in Genetic Programming." *Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, Orlando, Florida, USA, 1021-1026.
- [40] Fyfe, C., Marney, J. P., and Tarbert, H. (1999). "Technical Trading versus Market Efficiency: A Genetic Programming Approach." *Applied Financial Economics*, 9, 183-191.

- [41] Gagné, C., Schoenauer, M., Parizeau, M., and Tomassini, M. (2006). "Genetic Programming, Validation Sets, and Parsimony Pressure Genetic Programming." P. Collet, M. Tomassini, M. Ebner, S. Gustafson, and A. Ekárt, eds., Springer Berlin, Heidelberg, 109-120.
- [42] GEVA (2010). "GEVA v1.2." <<http://ncra.ucd.ie/Site/GEVA.html>>. (February, 2010).
- [43] Goldberg, D. E. (1989). *Genetic Algorithms for Search, Optimization, and Machine Learning*, Addison-Wesley, Reading.
- [44] Gorse, D. (2011). "Application of stochastic recurrent reinforcement learning to index trading." *Proc. European Symposium on Artificial Neural Networks –ESANN 2011, to appear*.
- [45] Hassan, G. N. A. (2010). "Multiobjective genetic programming for financial portfolio management in dynamic environments." Doctor of Philosophy Doctoral Thesis, University College London, London.
- [46] Holland, J. H. (1973). *Genetic Algorithms and the Optimal Allocation of Trials*, SIAM.
- [47] Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, Oxford, England: U Michigan Press.
- [48] Horn, J., Nafploitis, N., and Goldberg, D. (1994). "A niched Pareto genetic algorithm for multi-objective optimization." *Proceedings of the First IEEE Conference on Evolutionary Computation*, 82-87.
- [49] Java-GE (2010). "Java GE (jGE)." <<http://www.bangor.ac.uk/~eep201/jge/>>. (February, 2010).
- [50] Jensen, M., and Bennington, G. (1970). "Walks and Technical Theories: Some Additional Evidences." *J. Financ.*, 25, 469-482.

- [51] Keijzer, M., Ryan, C., O'Neill, M., Cattolico, M., and Babovic, V. (2001). "Ripple crossover in genetic programming." *Proc., Proceedings of the European Conference on Genetic Programming, EuroGP'2001*.
- [52] Knowles, J. D., and Corne, D. W. (2000). "Approximating the nondominated front using the Pareto Archived Evolution Strategy." *Evolutionary Computation Journal*, 8(2), 149-172.
- [53] Konak, A., Coit, D. W., and Smith, A. E. (2006). "Multi-objective optimization using genetic algorithms: A tutorial." *Reliability Engineering & System Safety*, 91(9), 992-1007.
- [54] Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA.
- [55] Koza, J. R. (1994). *Genetic Programming II*, MIT Press, Cambridge, MA.
- [56] Langdon, W. B. (2000). "Size Fair and Homologous Tree Crossovers for Tree Genetic Programming." *Genetic Programming and Evolvable Machines*, 1(1), 95-119.
- [57] Levich, R., and Thomas, L. (1993). "The Significance of Technical Trading-Rule Profits in the Foreign Exchange Market: A Bootstrap Approach." *Journal of International Money and Finance*, 12, 451-474.
- [58] Lin, D., Wang, S., and Yan, H. (2001). "A multiobjective genetic algorithm for portfolio selection." *Working Paper* Institute of Systems Science, Academy of Mathematics and Systems Science Chinese Academy of Sciences, Beijing, China.
- [59] Lo, A. W., Mamaysky, H., and Wang, J. (2000). "Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation." *J. Financ.*, 55(4), 1705-1765.
- [60] Lohpetch, D., and Corne, D. (2009). "Discovering Effective Technical Trading Rules with Genetic Programming: Towards Robustly Outperforming Buy-and-Hold." *World Congress on Nature and Biologically Inspired Computing (NABIC) 2009*, IEEE Press, 431—437.

- [61] Lohpetch, D., and Corne, D. (2010). "Outperforming Buy-and-Hold with Evolved Technical Trading Rules: Daily, Weekly and Monthly Trading." *Applications of Evolutionary Computation, Pt II, Proceedings*, C. DiChio, A. Brabazon, G. A. DiCaro, M. Ebner, M. Farooq, A. Fink, J. Grahl, G. Greenfield, P. Machado, M. Oneill, E. Tarantino, and N. Urquhart, eds., Springer-Verlag Berlin, Berlin, 171-181.
- [62] Malkiel, B. G. (2003). "The efficient market hypothesis and its critics." *Journal of Economic Perspectives*, 17(1), 59–82.
- [63] Mallick, D., Lee, V. C., and Yew Soon, O. (2008). "An empirical study of Genetic Programming generated trading rules in computerized stock trading service system." *Proc., Service Systems and Service Management, 2008 International Conference on*, 1-6.
- [64] Marney, J. P., Fyfe, C., Tarbert, H., and Miller, D. (2001). "Risk Adjusted Returns to Technical Trading Rules: A Genetic Programming Approach." *Computing in Economics and Finance* Soc. for Computational Economics, Yale University, USA.
- [65] Marney, J. P. T., H., and Fyfe, C. (2005). "Risk Adjusted Returns from Technical Trading: A Genetic Programming Approach." *Applied Financial Economics*, 15, 1073-1077.
- [66] Montana, D. J. (1995). "Strongly Typed Genetic Programming." *Evolutionary Computation*, 3(2), 199-230.
- [67] Murphy, J. J. (1999). *Technical Analysis of the Financial Markets*, New York Institute of Finance, New York.
- [68] Neely, C. (2001). "Risk-adjusted, ex ante, optimal technical trading rules in equity markets." *Working Papers 99-015D, Revised August 2001*, Federal Reserve Bank of St. Louis.
- [69] O'Neill, M., Brabazon, A., Ryan, C., and Collins, J. J. (2001). "Evolving Market Index Trading Rules Using Grammatical Evolution." *Proceedings of the EvoWorkshops on Applications of Evolutionary Computing*, Springer-Verlag, 343-352.

- [70] O'Neill, M., Hemberg, E., Gilligan, C., Bartley, E., McDermott, J., and Brabazon, A. (2008). "GEVA: grammatical evolution in Java." *SIGEVolution*, 3(2), 17-22.
- [71] O'Neill, M., Ryan, C., Keijzer, M., and Cattolico, M. (2003). "Crossover in Grammatical Evolution." *Genetic Programming and Evolvable Machines*, 4(1), 67-93.
- [72] O'Neil, M. (2001). "Automatic Programming in an Arbitrary Language: Evolving Programs in Grammatical Evolution." PhD thesis, University of Limerick.
- [73] O'Neil, M., and C., R. (2001). "Grammatical Evolution." *IEEE Trans. Evolutionary Computation*. 2001.
- [74] O'Neil, M., and Ryan, C. (2000). "Crossover in Grammatical Evolution: A Smooth Operator?" *Proceedings of the European Conference on Genetic Programming, EuroGP'2000*, Springer-Verlag., 149-162.
- [75] O'Neil, M., Ryan, C., Keijzer, M., and Cattolico, M. (2001). "Crossover in grammatical evolution: The search continues." *Proceedings of the European Conference on Genetic Programming, EuroGP'2001*, Springer-Verlag, Berlin, 337–347.
- [76] O'Neill, M., and Ryan, C. (2000). "Crossover in grammatical evolution: A smooth operator?" *Proceedings of the European Conference on Genetic Programming, EuroGP'2000*, Springer-Verlag, Berlin, 149–162.
- [77] O'Neill, M., and Ryan, C. (2003). *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*, Kluwer Academic Publishers, Dordrecht.
- [78] O'Neill, M., Vanneschi, L., Gustafson, S., and Banzhaf, W. (2010). "Open issues in genetic programming." *Genetic Programming and Evolvable Machines*, 11(3), 339-363.
- [79] Potvin, J. Y., Soriano, P., and Vallee, M. (2004). "Generating trading rules on the stock markets with genetic programming." *Comput. Oper. Res.*, 31(7), 1033-1047.

- [80] Prentis, E. L. (2011). "Evidence on a New Stock Trading Rule that Produces Higher Returns with Lower Risk." *International Journal of Economics and Finance*, 3(1), 92-104.
- [81] Pring, M. J. (1980). *Technical Analysis Explained*, McGraw-Hill, New York.
- [82] Pruitt, S. W., and While, R. E. (1988). "CRISMA Trading System: Who says Technical Analysis Can't Beat the Market?" *Journal of Portfolio Management*, 14, 55-58.
- [83] Rechenberg, I. (1973). *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*, Frommann-Holzboog.
- [84] Rothlauf, F., and Oetzel, M. (2006). "On the Locality of Grammatical Evolution." *Proc., 9th European Conference on Genetic Programming, EuroGP2006*, Springer, 320-330.
- [85] Ruggiero, M. A. (1997). *Cybernetic Trading Strategies*, Wiley, NY.
- [86] Ryan, C., Collins, J. J., and O'Neill, M. (1998). "Grammatical Evolution: Evolving Programs for an Arbitrary Language. Lecture Notes in Computer Science 1391." *Proceedings of the First European Workshop on Genetic Programming*, Springer, Verlag, 83-95.
- [87] Schaffe, J. D. (1984). "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms." PhD Thesis, Vanderbilt University, Tennessee
- [88] Schwefel, H.-P. P. (1993). *Evolution and Optimum Seeking: The Sixth Generation*, John Wiley & Sons, Inc.
- [89] Seshadri, M. (2003). "Comprehensibility, Overfitting and Co-evolution in Genetic Programming for Technical Trading Rules." Master Thesis, Worcester Polytechnic Institute.
- [90] Sharpe, W. F. (1966). "Mutual Fund Performance." *Journal of Business*, 39(S1), 119-138.

- [91] Shoaf, J. S., and Foster, J. A. (1996). "A Genetic Algorithm Solution to the Efficient Set Problem: A Technique for Portfolio Selection Based on the Markowitz Model." *Proceedings of the decision sciences institute annual meeting* Orlando, Florida, 571–573.
- [92] Srinivas, N., and Deb, K. (1994). "Multi-objective function optimization using non-dominated sorting genetic algorithms." *Evolutionary Computation Journal*, 2(3), 221–248.
- [93] StockCharts.com (2011). "Moving Averages." *ChartSchool-Technical Indicators and Overlays*,
<http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:moving_averages>. (August, 2011).
- [94] StockCharts.com (2011). "Rate of Change (ROC)." *ChartSchool-Technical Indicators and Overlays*,
<http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:rate_of_change_roc_a>. (August, 2011).
- [95] StockCharts.com (2011). "Support and Resistance." *ChartSchool-Chart Analysis*,
<http://stockcharts.com/school/doku.php?id=chart_school:chart_analysis:support_and_resistance>. (August, 2011).
- [96] StockCharts.com (2011). "Trend Lines." *ChartSchool-Chart Analysis*,
<http://stockcharts.com/school/doku.php?id=chart_school:chart_analysis:trend_lines>. (August, 2011).
- [97] StockCharts.com (2011). "Why Analyze Securities?" *ChartSchool-Overview*
<http://stockcharts.com/school/doku.php?id=chart_school:overview:why_analyze_securities>. (August, 2011).
- [98] Streichert, F., Ulmer, H., and Zell, A. (2004). "Comparing Discrete and Continuous Genotypes on the Constrained Portfolio Selection Problem." *Genetic and evolutionary computation–gecco 2004. proceedings of the genetic and evolutionary computation conference. part ii*, K. D., ed., Springer-Verlag, Seattle, Washington, USA, 1239–1250.

- [99] Sweeney, R. (1986). "Beating the Foreign Exchange Market." *J. Financ.*, 41(1), 163-182.
- [100] Taylor, M., and Allen, H. (1992). "The Use of Technical Analysis in the Foreign Exchange Market." *Journal of International Money and Finance*, 11, 304-314.
- [101] Uy, N. Q., Hoai, N. X., O'Neill, M., McKay, R. I., and Galvan-Lopez, E. (2011). "Semantically-based Crossover in Genetic Programming: Application to Real-Valued Symbolic Regression." *Genetic Programming and Evolvable Machines*, 12(2), 91-119.
- [102] Vanneschi, L., Rochat, D., and Tomassini, M. (2007). "Multi-optimization improves genetic programming generalization ability." *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, ACM, London, England, 1759-1759.
- [103] Vedarajan, G., Chan, L. C., and Goldberg, D. E. (1997). "Investment Portfolio Optimization using Genetic Algorithms." *Late breaking papers at the genetic programming 1997 conference*, J. R. Koza, ed., Stanford Bookstore, Stanford University, California, 255–263.
- [104] Wikipedia (2011). "Momentum (technical analysis)."
<http://en.wikipedia.org/wiki/Rate_of_change_%28technical_analysis%29>. (August, 2011).
- [105] Wikipedia (2011). "Pivot point."
<http://en.wikipedia.org/wiki/Pivot_point_calculations>. (August, 2011).
- [106] Wikipedia (2011). "Support and Resistance."
<http://en.wikipedia.org/wiki/Support_and_resistance>. (August, 2011).
- [107] Wikipedia (2011). "Trend line (Technical Analysis)."
<http://en.wikipedia.org/wiki/Trend_line_%28technical_analysis%29>. (August, 2011).
- [108] Wilson, G., and Banzhaf, W. (2010). "Interday foreign exchange trading using linear genetic programming." *GECCO '10: Proceedings of the 12th annual conference on Genetic and evolutionary computation*, ACM, Portland, Oregon, USA, 1139-1146.

- [109] Yin, Z., Brabazon, A., O'Sullivan, C., and O'Neill, M. (2007). "Genetic Programming for Dynamic Environments." *Proc., Proceedings of the International Multiconference on Computer Science and Information Technology*, 437-446.
- [110] Yu, T., and Bentley, P. (1998). "Methods to Evolve Legal Phenotypes." *In Proceedings of the Fifth Int. Conf. on Parallel Problem Solving From Nature*, Springer, 280--291.
- [111] Zitzler, E., Deb, K., and Thiele, L. (2000). "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results." *Evolutionary Computation*, 8(2), 173-195.
- [112] Zitzler, E., Laumanns, M., and Thiele, L. (2001). "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization." *Proc., International Center for Numerical Methods in Engineering (CIMNE)*, 95-100.
- [113] Zitzler, E., and Thiele, L. (1999). "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach." *Evolutionary Computation, IEEE Transactions on*, 3(4), 257-271.