Table of Contents

Student Plagiarism: Compliance Statement .... *i*

Acknowledgement .... *ii*

Abstract .... *iii*

Table of Contents .... *iv*

Chapter 1. Introduction *1*

1.1 Contribution **2**

1.1.1 Publications 3

1.2 Document Outline **3**

Chapter 2. Background *4*

2.1 Sequencing **4**

2.2 After Sequencing – The Computational Pipeline **6**

2.3 The need for testing in genomic medicine software **8**

2.4 Software testing concepts and approaches **9**

2.5 Current testing approaches used in the genomics field **10**

2.6 State-of-the-art testing approaches **11**

2.6.1 Adaptive Random Testing 11

2.6.2 Metamorphic Testing 12

2.7 Cloud Computing **13**

Chapter 3. Aim *17*

Chapter 4. Methodology *19*

4.1 Testing Framework **19**

4.2 Tests **20**

4.2.1 Metamorphic Testing 20

4.2.2 Traditional / Simulated Testing 21

Chapter 5. Implementation *23*

5.1 Overview **24**

5.2 Amazon Web Services – Relevant Background **24**

5.2.1 EC2 24

5.2.2 S3 25

5.2.3 Management Console 26

5.2.4 Client and API 26

5.3 Platform and Programming Environment **26**

5.4 Tests **27**

5.4.1 Metamorphic Tests 27
5.4.2 Tests Using Simulated Data 27

5.5 Specific Details of Framework 28

5.5.1 AWS Credentials and Security 28
5.5.2 Instance Payment Model and Type 28
5.5.3 AMI Image 28
5.5.4 SSD Option 29
5.5.5 Framework Configuration 29
5.5.6 GPT Workflow 31
5.5.7 GPT Load Balancing and Scalability 31

Chapter 6. Results 34

6.1 Test Configuration Details 34

6.1.1 AWS 34
6.1.2 Data 35
6.1.3 Details of vGSP under Test 36

6.2 Preliminary Testing 36

6.2.1 Instance Start-up Times 36
6.2.2 Accessing “local” SSD resources on AWS EC2 37

6.3 Main GPT Test 38

6.3.1 Timing Information 39
6.3.2 Cost 42
6.3.3 Test Results 44

Chapter 7. Discussion 49

Chapter 8. Conclusion 53

Appendix 1. File Type Descriptions 55

8.1 Input Data – FASTQ files 55

8.2 Input Data – Reference Genome – FASTA files 56

8.3 Intermediate output – SAM/BAM files 56

8.4 Output file – VCF 57

Appendix 2. GPT Configuration File 58

Appendix 3. Example vGSP Code 59

Appendix 4. Report File from a GPT Run 61

Bibliography 66

CONTENTS

Student Plagiarism: Compliance Statement **ii**

Abstract **iii**

Acknowledgements **iv**

List of Figures **viii**

List of Tables **xiii**

Chapter 1 Introduction **1**

1.1 Background and Motivation 1

1.2 Scientific Contributions and Discoveries 6

1.3 Thesis Structure 7

Part 1. Fungal Classification **8**

Chapter 2 Introduction to Fungal Classification **9**

2.1 Overview 9

2.2 Fungal Taxonomy 9

2.3 Ribosomal RNA Genes for Classification 10

2.4 Classification Tools for Species Identification 13

2.4.1 Similarity-based Classifiers 13

2.4.2 Phylogeny-based Classifiers 17

2.4.3 Composition-based Classifiers 18

2.4.4 Summary 20

2.5 RDP Naïve Bayes Classifier 22

2.5.1 k-mer Feature Space 22

2.5.2 Statistical-based Classification 23

2.5.3 Bootstrapping 24

Chapter 3 Methods for Fungal Classification 25

3.1 Overview 25

3.2 Fungal ITS Sequence Pre-processing and Curation 25

3.3 Training Set Creation and Evaluation 31

3.4 Validation Set Creation and Classifier Evaluation 32

Chapter 4 Results and Discussion 34

4.1 Overview 34

4.2 Characteristics of the Training Set 34

4.2.1 ITS Sequence Lengths 35

4.2.2 Taxonomic Composition 36

4.3 Training Set Performance 39

4.4 Characteristics of the Validation Set 42

4.4.1 ITS Sequence Lengths 43

4.4.2 Taxonomic Composition 43

4.5 Validation Set Performance 44

4.6 Amplicon Sequencing Simulation Performance 46

4.7 Effects of Sequence Composition on Confidence Values 49

4.8 Future Work 51

4.9 Conclusions 53

Part 2. Evolutionary Dynamics of Aflatoxin Genes 54

Chapter 5 Introduction to the Evolutionary Dynamics of Aflatoxin Genes 55

5.1 Overview 55

5.2 Aflatoxin Biosynthesis 55

5.2.1 Aflatoxin Types B and G 59

5.3 Evolutionary Dynamics among Fungal Organisms 60

5.3.1 Classification of *Aspergillus* Species 61

5.3.2 Evolution of *Aspergillus* Species 63

5.4 Summary 66

Chapter 6 Methods for Analysing Evolutionary Dynamics 67

6.1 Overview 67

6.2 Fungal Genome Assembly 67

6.2.1 Next-Generation Sequencing 68

6.2.2 Velvet Assembler 71

6.2.3 Sequence Error Correction and Assembly of Sequence Reads 76

6.3 Evolutionary Dynamics of Aflatoxin Genes 78

6.3.1 Characterisation of Aflatoxin Gene Pathway 78

6.3.2 Phylogenetic Analysis 80

Chapter 7 Results and Discussion 86

7.1 Overview 86

7.2 Fungal Genome Assembly 86

7.2.1 Assembly Statistics 87

7.2.2 Effects of Coverage Cutoff Parameter 94

7.3 Evolutionary Dynamics of Aflatoxin Genes 95

7.3.1 Characterisation of Aflatoxin Biosynthesis Pathway 95

7.3.2 Phylogenetic Analysis 105

7.4 Future Work 129

7.5 Conclusions 131

Chapter 8 Conclusions 133

Bibliography 136

CONTENTS

Student Plagiarism: Compliance Statement ii

Abstract iii

Acknowledgements iv

List of Figures viii

List of Tables x

Chapter 1 Introduction 1

1.1 Embodied AI 1

1.2 Audio-Visual Navigation 2

1.2.1 Importance of Audio-Visual Navigation 2

1.2.2 Challenges of Audio-Visual Navigation 3

1.3 Contributions 4

1.4 Thesis Outline 5

Chapter 2 Literature Review 6

2.1 Embodied AI in 3D Environments 6

2.1.1 Embodied AI Simulators 6

2.1.2 Embodied AI Tasks 7

2.1.3 Reinforcement Learning 9

2.2 Goal-Oriented Navigation 10

2.2.1 Various Navigation Tasks 11

2.2.2 Navigation Map Construction 14

2.2.3 Memory Mechanism 17

2.3 Audio-Visual Multi-Modality Learning 18

2.3.1 Audio-Visual Representation Learning 19

2.3.2 Embodied Audio-Visual Learning 21

2.3.3 Audio-Visual Navigation 23

Chapter 3 Methods **27**

3.1 Input Processing 29

3.2 Visual Mapping 30

3.3 Acoustic Mapping 31

3.4 AFSO-based Audio Encoding 31

3.4.1 Intuition 32

3.4.2 Define Similar Audio Pairs 33

3.4.3 Optimisation Method 38

3.4.4 Batch Sampling 38

3.5 Source Sound Augmentation 41

3.5.1 Audio Mix-up 41

3.5.2 Audio Reverse 42

3.5.3 Sound Augmentation Module 42

3.6 Waypoint Prediction 43

3.7 Path Planning 44

Chapter 4 Experiments and Results **45**

4.1 Task Definition 45

4.2 Dataset 46

4.3 Metrics 48

4.4 Implementation 48

4.5 Quantitative Comparisons of Navigation Results 49

4.6 Navigation Results Visualisations 50

4.7 Ablation Studies and Results 54

4.8 Extensive Comparisons on Replica 55

Chapter 5 Discussion **57**

5.1 Designs of the AV-GeN Framework 57

5.2 AFSO Method 58

5.2.1 Implications Behind the Contrastive Optimisation 58

5.2.2 Designs of Batch Sampling 59

5.2.3 Designs of Projection Head 60

5.3 Source Sound Augmentation 60
5.4 Limitations in the SoundSpaces AVN Simulator 61

Chapter 6 Conclusion 63

Bibliography 64

Contents

Student Plagiarism: Compliance Statement **ii**

Abstract **iii**

Acknowledgements **iv**

List of Publications **v**

List of Figures **ix**

List of Tables **xi**

Chapter 1 Introduction **1**

1.1 Background and Problem Definition 1

1.2 Current Progress and Challenges 1

1.3 Main Contributions 3

1.4 Thesis Outline 4

Chapter 2 Literature Review **5**

2.1 Deep Learning Networks and Frameworks 5

2.2 Anomaly Detection in Natural Imaging 7

2.3 Anomaly Detection in Medical Imaging 9

2.4 Memory Networks 11

Chapter 3 Methods **13**

3.1 SQUID Overview 13

3.2 Inventing Memory Queue as Dictionary 14

3.2.1 Motivation 14

3.2.2 Space-aware Memory 15

3.2.3 Memory Queue 15

3.3 Gumbel Shrinkage 18

3.3.1 Motivation 18
3.3.2 Combining Hard Shrinkage and Gumbel Softmax 18
3.4 Formulating Anomaly Detection as In-painting 18
3.4.1 Motivation 18
3.4.2 In-painting Block 19
3.4.3 Masked Shortcut 20
3.5 Anomaly Discrimination 20
3.5.1 Motivation 20
3.6 Loss Functions 21
3.7 Network Architectures 22
3.8 Creation of DigitAnatomy 23

Chapter 4 Experiments and Results **26**
4.1 Experimental Designs 26
4.1.1 Public Benchmarks 26
4.1.2 DigitAnatomy 27
4.1.3 Comparing Methods 27
4.1.4 Metrics 28
4.1.5 Implementation Details 29
4.2 Image Reconstruction Results on DigitAnatomy 30
4.3 Results on Public Datasets 30
4.4 Ablation Studies 37
4.5 Extensive Studies 38
4.6 Robustness to Disease Samples in the Training Set 42

Chapter 5 Discussion **43**
5.1 Limitations 43
5.2 Future Work 44

Chapter 6 Conclusion **47**

Bibliography **49**

Contents

Student Plagiarism: Compliance Statement ii

Abstract iii

Acknowledgements iv

List of Figures viii

Chapter 1 Introduction 1

1.1 Our Contribution 2

1.2 Outline 3

Chapter 2 Preliminary Notions 5

2.1 Spanners 5

2.1.1 The Θ-Graph 6

2.1.2 Bounded Degree and Light Planar Spanners Exist 7

2.2 Online Routing 7

2.2.1 Routing in the Θ-Graph 8

2.2.2 Greedy Routing 8

2.2.3 Compass Routing 9

2.2.4 Right-Hand Routing 9

2.3 Delaunay Triangulations 9

Chapter 3 Everything But Lightness 12

3.1 Description of $\mathcal{BDG}(V)$ 12

3.2 Degree Bound of $\mathcal{BDG}(V)$ 14

3.3 Spanning Properties of $\mathcal{BDG}(V)$ 14

3.3.1 A Separation of Neighbours of Neighbours 14

3.3.2 Notion of Protection 15

3.3.3 Lesser Protection Buys Neighbours in the Same Cone 15

3.3.4 Penultimate and Middle Edges are Bought Full Protection . . . 16
3.3.5 Hull Paths Within Cones are Almost Complete . . . 16
3.3.6 Proof of the Spanning Ratio . . . 17
3.4 Algorithmic Construction of $\mathcal{BDG}(V)$ . . . 20
3.5 $\mathcal{BDG}(V)$ Defeats Both Greedy and Compass Routing . . . 21

Chapter 4 Marking the Terrain 22

4.1 Everything is Protected . . . 22
4.2 Signposts . . . 23
4.3 Algorithmic Construction of $\mathcal{MBDG}(V)$ . . . 26

Chapter 5 Routing 28

5.1 Chew's Routing Algorithm . . . 28
5.2 Compromised Chew's Routing Algorithm . . . 29
5.2.1 Termination . . . 31
5.2.2 Routing Ratio . . . 31
5.3 Simulating Compromised Chew's Routing Algorithm on $\mathcal{MBDG}(V)$ . . . 35
5.3.1 Preliminaries . . . 35
5.3.2 The Simulation . . . 38

Chapter 6 Lightness, and Routing Redux 45

6.1 The Levcopoulos and Lingas Protocol . . . 45
6.1.1 Preliminaries . . . 46
6.1.2 How the Polygon P Grows . . . 46
6.1.3 Condition For Including an Edge . . . 48
6.2 $weight$ Refers to Face Paths . . . 55
6.3 Signposts II . . . 56
6.4 Routing on $\mathcal{LMBDG}(V)$ . . . 56

Chapter 7 Conclusion 59

7.1 Further Work . . . 59
7.1.1 Tighter Bounds on the Routing Ratio of $\mathcal{LMBDG}(V)$ . . . 60
7.1.2 Navigating Unmarked Terrain . . . 60
7.1.3 Pushing the Degree Down . . . 60
7.1.4 Memoryless Navigators . . . 60

7.1.5 Tighter Weight Bound 61

7.1.6 Simulation of Alternative Algorithms 61

References **62**

CONTENTS

Student Plagiarism: Compliance Statement **ii**

Abstract **iii**

Acknowledgements **iv**

List of Figures **vii**

List of Tables **ix**

Chapter 1 Introduction **1**

Chapter 2 Background **5**

2.1 Markov Decision Process (MDP) 5

2.2 Dynamic Programming and Reinforcement Learning 9

2.3 Linear Temporal Logic 13

2.4 Automata 16

Chapter 3 Problem Statement **19**

Chapter 4 Related Work **20**

4.1 Satisfaction Goals 20

4.2 Challenges of Objective Satisfaction in RL 21

4.3 Objective-based RL for LTL 22

4.3.1 Satisfaction Guarantees without Automata 23

4.3.2 Automata-Based Approaches 23

4.3.3 Miscellaneous RL Settings for LTL 25

4.4 Objective-based RL for Finite LTL 26

4.4.1 Direct Translation to Rewards 26

4.5 Reward Machines 27

4.6 Planning with Finite LTL 29

4.7 RL with LTL Landscape 29

4.8 Finite Horizon Problem in Reward-Based RL 31

4.8.1 Time-Awareness in Deep RL 31

Chapter 5 Methods **32**

5.1 Objective-Based RL Theorems 32

5.2 Objective-Based RL Algorithms 36

5.3 Speeding Up the Agent with Heuristics 39

Chapter 6 Experimental Evaluation **41**

6.1 Frozen Lake Environment 41

6.2 Reward Machines for Tasks 44

6.3 Optimal Policies 44

6.3.1 Infinite Horizon 45

6.3.2 Finite Horizon 46

6.4 Heuristic Methods 49

6.5 Model Efficiency Experiments 51

6.6 Code 55

Chapter 7 Discussion **56**

7.1 Terminal States 56

7.2 Other RL Paradigms 57

7.3 Improving Update Efficiency 57

7.4 Reward Shaping 58

7.5 Time-Dependent and Reward-Machine-Dependent Environment Transitions 59

7.6 Minimising Time 59

Chapter 8 Conclusion **61**

Bibliography **62**

Appendix Appendix **66**

1 Additional Background 66

2 Additional Figures 66

Contents

Student Plagiarism: Compliance Statement ii

Abstract iii

Acknowledgements iv

List of Figures vii

Chapter 1 Introduction 1

1.1 Problems 2

1.2 Motivation 2

1.3 Contributions 3

1.4 Related work 6

1.5 Outline 6

Chapter 2 Background 8

2.1 Problem definition 8

2.1.1 Deadline variant 9

2.1.2 Delay variant 9

2.2 Online setting 9

2.2.1 Measuring performance 9

2.2.2 Running time 10

2.2.3 Deterministic vs Randomized 10

2.2.4 Clairvoyance 11

2.3 TCP acknowledgement 12

2.4 Greedy approach 13

2.4.1 Joint replenishment problem 13

Chapter 3 Multi-level aggregation problem 16

3.1 Joint replenishment problem 16

3.1.1 Problem definition 16

3.1.2 Lower bound 16

3.1.3 Deadline variant 18

3.1.4 Delay variant 25

3.2 Multi-level aggregation problem 33

3.2.1 Problem definition 33

3.2.2 Algorithm description 33

3.2.3 Analysis 36

Chapter 4 Set cover with delay problem **46**

4.1 Problem definition 46

4.2 Non-clairvoyant setting 46

4.2.1 Intuition 47

4.2.2 Preliminary notations 48

4.2.3 Algorithm description 48

4.2.4 Analysis 51

4.3 Clairvoyant setting 56

4.3.1 Intuition 56

4.3.2 Prize-collecting set cover 57

4.3.3 $O(\log m)$-competitive algorithm 58

4.3.4 $O(\log n)$-competitive algorithm 58

Chapter 5 Conclusion **62**

Bibliography **63**

CONTENTS

Student Plagiarism: Compliance Statement ii

Abstract iii

Acknowledgements iv

List of Figures vii

Chapter 1 Introduction 1

1.1 Results and Techniques 4

1.2 Further Related Work 6

1.3 Thesis Outline 10

Chapter 2 Overview of Cardinality JRP Algorithms 11

2.1 Problem Definition: Piecewise Cardinality JRP 11

2.2 A reduction to Piecewise Cardinality JRP 12

2.3 Algorithm Challenges 14

2.4 Algorithm Intuition Overview 15

Chapter 3 Piecewise Cardinality JRP with Deadlines Algorithms 18

3.1 Intuition: Rent-or-Buy JRP with Deadlines 18

3.2 Rent-Or-Buy Cardinality JRP with Deadlines Algorithm 22

3.2.1 Algorithm Description 22

3.2.2 Analysis Sketch 24

3.3 Intuition: Piecewise Cardinality JRP with Deadlines 25

3.4 Piecewise Cardinality JRP with Deadlines Algorithm 28

3.4.1 Algorithm Description 28

3.4.2 Analysis Overview 32

3.4.3 Analysis: Solution Structure 33

3.4.4 Analysis: Charging service costs 33

3.4.5 Analysis: Charging to OPT 37

Chapter 4 Piecewise Cardinality JRP with Delay Algorithm 42

4.1 Intuition 42

4.2 Algorithm Description 47

4.3 Analysis Overview 52

4.4 Analysis: Solution Structure 53

4.5 Analysis: Charging service costs 57

4.6 Analysis: Charging to OPT 60

4.7 Time Complexity Considerations 70

Chapter 5 Weighted Cardinality JRP 74

5.1 Algorithm Description 74

5.2 Analysis 75

Chapter 6 Conclusions and Open Problems 83

Bibliography 84

Appendix A Rent-Or-Buy Cardinality JRP with Deadlines Pseudocode 86

Appendix B Rent-Or-Buy Cardinality JRP with Deadlines Analysis 88

B.1 Analysis: Solution Structure 88

B.2 Analysis: Charging tail service costs 89

B.3 Analysis: Charging to OPT 91

CONTENTS

Abstract **ii**

Contents **iii**

List of Figures **v**

List of Programs **vi**

1 Introduction **1**

1.1 Contributions 2

2 Background **3**

2.1 Data placement 3

2.2 Data migration 5

2.3 Random data layout 6

2.4 Expander graphs 6

2.5 Randomised load balancing 7

3 Analysis **8**

3.1 Problem statement 8

3.2 Evaluating a layout 10

3.3 Random data layouts 12

3.4 Prelude to the analysis 13

3.5 Expander graph analysis 14

3.6 Alternate analysis 16

3.7 Baseline model 19

4 Evaluation **21**

4.1 The Zipf distribution 21

4.2 Threshold phenomenon 22

4.3 Results 23

4.4 Other sizes 25

5 Partitioning the demand **29**

5.1 Analysis: two parts 29

5.2 Partitioning heuristics: two parts 30

5.3 Evaluation: two parts 32

5.4 Analysis: t parts . . . 33
5.5 Partitioning heuristics: t parts . . . 34
5.6 Evaluation: t parts . . . 38

6 Bound on performance **40**
6.1 Approximate lower bound . . . 40
6.2 Mixed integer program formulation . . . 42
6.3 Evaluation . . . 44

7 Maximum coverage **46**
7.1 Coverage derived from the maximum load . . . 46
7.2 Upper bound . . . 52
7.3 Simulations . . . 55

8 Discussion **58**
8.1 Results . . . 58
8.2 Open problems . . . 60

9 Conclusion **62**

Bibliography **64**

Contents

Student Plagiarism: Compliance Statement **ii**

Abstract **iii**

Acknowledgements **iv**

List of Figures **ix**

List of Tables **xi**

Chapter 1 Introduction **1**

1.1 Contributions 3

1.1.1 Study of Temporal Compatibility 3

1.1.2 Practical ILP Formalisation 4

1.2 Outline 5

Chapter 2 The Cophylogeny Reconstruction Problem **6**

2.1 General Definitions 6

2.1.1 Phylogeny Trees 6

2.1.2 Host-Parasite Associations and Dependence 8

2.1.3 Tanglegrams in Cophylogeny Reconstruction 8

2.2 Problem Definition 9

2.2.1 Events 11

2.2.2 Constraints 18

2.3 Problem Variations 22

2.3.1 Widespread Parasites 22

2.3.2 Non-Independent Parasites 23

2.3.3 Reticulate Networks 23

Chapter 3 Background Work **24**

3.1 Early Developments 24

3.1.1 Brooks Parsimony Analysis . . . 24
3.1.2 Reconciled Tree Methods . . . 25
3.2 Toward Contemporary Cophylogeny – Tree Mapping . . . 26
3.2.1 Event Costs . . . 27
3.2.2 Pareto Sets . . . 28
3.3 Current Heuristics and Approaches . . . 28
3.3.1 Cost Matrix . . . 29
3.3.2 Jungles . . . 29
3.3.3 TARZAN . . . 30
3.3.4 JANE . . . 30
3.3.5 CORE-PA . . . 31
3.3.6 TREEMAP 3 . . . 32
3.4 Intractibility and Approximations Results . . . 32
3.4.1 Moving Back Landing Site Problem . . . 32
3.4.2 The Cophylogeny Reconstruction Problem . . . 33
3.5 Summary . . . 33

Chapter 4 Temporal Incompatibilities and Host Switch Resolution 35
4.1 Background . . . 36
4.1.1 Anatomy of a Host Switch . . . 36
4.1.2 Incompatibilities . . . 37
4.1.3 Newer Approaches . . . 38
4.2 Feasible Reconstructions and Event Orders . . . 39
4.2.1 Phylogeny Implicit Orders . . . 39
4.2.2 Host Switch Orders . . . 40
4.3 j-vertex – The Jungle Method . . . 44
4.3.1 Sufficiency . . . 45
4.3.2 Complexity . . . 46
4.4 Proposed Method . . . 47
4.4.1 Necessity . . . 48
4.4.2 Sufficiency . . . 50
4.4.3 Complexity . . . 53
4.5 Summary . . . 54

Chapter 5 Integer Linear Program Formulation 55

5.1 Previous ILP Formulation 55

5.1.1 Fixed Event Times 56

5.2 Proposed ILP Formulation 57

5.2.1 Model Assumptions 58

5.2.2 ILP Formulation 59

5.3 Formulation Correctness 66

5.3.1 Necessity 66

5.3.2 Sufficiency 69

5.3.3 Event Counts 71

5.4 Variable and Constraint Counts 73

5.5 Summary 73

Chapter 6 Reference Implementation 74

6.1 Implementations 74

6.1.1 JANE 74

6.1.2 Proposed Integer Linear Program (ILP) 74

6.1.3 Previous ILP 75

6.2 Experiments 75

6.2.1 Platform 75

6.2.2 Datasets 76

6.3 Results 78

6.3.1 Synthetic Datasets 78

6.3.2 Real World Datasets 81

Chapter 7 Conclusion 84

7.1 Future Work 84

7.1.1 Pushing the bounds of intractability further 84

7.1.2 Expanding the model of reconstruction 85

7.2 Conclusions 86

Appendix A Corrections to Previous ILP Formulation 87

A.1 Missing Constraints 87

A.2 Incorrect Constraints 88

Appendix B Full ILP Listing **91**

Appendix C Full Result Tables **93**

C.1 Column Descriptions 93

C.2 Synthetic Datasets 94

C.3 Real World Datasets 103

Bibliography **105**

CONTENTS

Student Plagiarism: Compliance Statement **ii**

Abstract **iii**

Acknowledgements **iv**

List of Figures **vii**

List of Tables **ix**

Notation **x**

Chapter 1 Introduction **1**

1.1 Contribution . . . 2

1.2 Outline . . . 3

Chapter 2 Background **4**

2.1 Approximation & Randomised Algorithms . . . 4

2.2 Set Function Optimisation Problems . . . 5

2.2.1 Matroid Constraints . . . 6

2.2.2 Knapsack Constraints . . . 7

2.2.3 Independent Set in Interval Graphs . . . 7

2.3 Algorithm Techniques . . . 7

2.3.1 Greedy Algorithms . . . 8

2.3.2 Continuous Greedy . . . 9

2.3.3 Contention Resolution Schemes . . . 12

2.3.4 Pipage Rounding . . . 14

Chapter 3 Preliminaries **17**

3.1 Maximum Facility Location . . . 17

3.2 Integer Program . . . 18

3.3 Linear Relaxation 19

Chapter 4 Algorithms **20**

4.1 Independent Rounding 20

4.1.1 Algorithm Engineering 24

4.2 Dependent Rounding 25

4.3 Hardness 32

4.3.1 Reducing from Maximum Coverage 32

Chapter 5 Conflicting Predictions from Multi-mapping Reads **34**

5.1 Modelling the Problem 35

5.2 Implementation 36

5.3 Related Work 38

5.4 Experiments 39

5.4.1 Simulated Reads: Craig Venter 39

5.4.2 Illumina Reads: NA12878 43

Chapter 6 Discussion **47**

6.1 Algorithms 47

6.2 Experiments 48

6.3 Open Problems and Future Work 48

Chapter 7 Conclusion **50**

Bibliography **52**

CONTENTS

Student Plagiarism: Compliance Statement **iii**

Abstract **iv**

Acknowledgements **v**

List of Figures **viii**

List of Tables **x**

Chapter 1 Introduction **1**

Chapter 2 Background **5**

2.1 The Changing Nature of Database Systems 5

2.1.1 The Issues with Mutual Exclusion as a Synchronisation Tool 6

2.2 Non-blocking Synchronisation Techniques 7

2.2.1 Definition of Non-Blocking 7

2.2.2 Atomic Primitives 7

2.2.3 The ABA Problem 9

2.2.4 Classifying Primitives 9

2.2.5 Multi-word Primitives 10

2.2.6 Non-Blocking Data Structures 11

Chapter 3 Problem Statement and Methodology **14**

3.1 Implementing a Non-Blocking Key-Value Store 14

3.1.1 The Key-Value Store Interface 15

3.1.2 Non-Blocking Synchronisation 15

3.1.3 Linearisability: A Correctness Condition 15

3.1.4 Memory Reclamation for Unmanaged Languages 16

3.2 Testing Performance in the Face of Contention 16

3.2.1 Synchrobench 16

3.2.2 The Comparison Skip List: Fraser's Non-Blocking Skip List . . . 17
3.2.3 Definitions of Performance Metrics . . . 17
3.2.4 Experimental Settings . . . 17
3.2.5 The Test Machine . . . 18

Chapter 4 The Rotating Skip List 19

4.1 Overview of the No Hot Spot Non-Blocking Skip List . . . 19
4.2 The Naive Design . . . 20
4.3 Consolidation of Index Nodes into Rotating Arrays . . . 21
4.4 Reducing the Frequency of Background Maintenance . . . 24
4.5 Delegating Physical Deletes to the Background Thread . . . 26
4.6 Memory Reclamation for The Rotating Skip List . . . 29
4.6.1 Reducing the Frequency of Memory Reclamation . . . 31

Chapter 5 Discussion and Suggestions for Future Work 33

5.1 Insights Reached During Development of The Rotating Skip List . . . 33
5.2 Suggestions for Future Work . . . 34
5.2.1 Improving Performance in Other Test Scenarios . . . 34
5.2.2 Expanding the Interface of The Rotating Skip List . . . 36

Chapter 6 Conclusion 38

Bibliography 39