THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Analysing and Comparing Problem Landscapes for Black-Box Optimization via Length Scale

Rachael Ann Morgan

B. Eng. (Hons I)

A thesis submitted for the degree of Doctor of Philosophy at

The University of Queensland in 2015

School of Information Technology and Electrical Engineering

Abstract

Optimization problems are of fundamental practical importance and can be found in almost every aspect of human endeavour. Yet remarkably, we have a very limited understanding of the nature of optimization problems and subsequently of how, why and when different algorithms perform well or poorly. The notion of a problem landscape captures the relationship between the objective function and the problem variables. It is clear that the structure of this landscape is vital in understanding optimization problems, however analysis of this structure presents major challenges. Apart from the often high-dimensionality of problem landscapes, information available in the black-box setting is limited to the solutions in the feasible search space and their respective objective function values. Landscape analysis has received some attention in the optimization literature, mainly in evolutionary computation. However there are some important limitations of this work as well as many open issues around its practical utility.

This thesis proposes a novel framework and practical techniques for the analysis of optimization problems utilizing information available in the black-box setting. The concept of length scale is proposed as a fundamental feature of both combinatorial and continuous optimization problems. Analytical properties of length scale and its distribution over a given problem are established. Techniques from statistics, set theory, visualisation and machine learning are employed to summarise and interpret sampled length scale values. From the length scale distribution, a problem similarity measure is proposed using the entropic Jeffrey divergence. This provides a means of comparing arbitrary black-box optimization problems, between combinatorial or continuous problems of possibly different dimensionality. An alternative realisation of the framework is also developed for quantifying optimization problem similarity via length scale information, based on the notion of Information Distance from Kolmogorov Complexity Theory. Information Distance is a universal distance measure between two arbitrary objects, and in practice can be approximated by Normalised Compression Distance, which relies on binary representations of the problems of interest and a lossless compressor. A novel methodology for calculating the Normalised Compression Distance in the optimization context is developed.

The techniques proposed are implemented and extensively evaluated via experiments on continuous artificial, benchmarking and real-world representative problems, and instances of NP-hard classes of combinatorial problems. The results convincingly show that length scale features are highly effective and robust for comparing and characterizing optimization problems. The calculated Jeffrey divergences and Normalised Compression Distances between length scale distributions are able to identify known similarities among problems, and also provide valuable insights into the relationship between problems. Known phase transitions in the difficulty and structure of the combinatorial problem instances are clearly reflected in the results for both similarity measures. This is remarkable given that only black-box information is used in the analysis. Finally, the theoretical and empirical relationship between the Jeffrey divergence and Normalised Compression Distance is studied. The features are shown to be different but conceptually related, and provide complementary empirical information.

Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

Rachael Ann Morgan

Publications during candidature

Peer Reviewed Articles

R. Morgan and M. Gallagher (2014). Sampling Techniques and Distance Metrics in High Dimensional Continuous Landscape Analysis: Limitations and Improvements. In *IEEE Transactions on Evolutionary Computation*, vol.18, no.3, pp. 456-461. IEEE Press.

R. Morgan and M. Gallagher (2014). Fitness Landscape Analysis of Circles in a Square Packing Problems. In *Simulated Evolution And Learning (SEAL 2014)*. pp. 455-466. Springer.

R. Morgan and M. Gallagher (2012). Length Scale for Characterising Continuous Optimization Problems. In *Parallel Problem Solving from Nature (PPSN XII)*, pp. 407-416. Springer.

Publications included in this thesis

R. Morgan and M. Gallagher (2012). Length Scale for Characterising Continuous Optimization Problems. In *Parallel Problem Solving from Nature (PPSN XII)*, pp. 407-416. Springer.

- Incorporated in Chapters 5, 6 and 7.

Contribution	**R. Morgan (%)**	**M. Gallagher (%)**
Conceptual ideas	70	30
Experimental design	85	15
Experimental implementation	100	0
Data analysis	75	25
Paper write-up	55	45

R. Morgan and M. Gallagher (2014). Sampling Techniques and Distance Metrics in High Dimensional Continuous Landscape Analysis: Limitations and Improvements. In *IEEE Transactions on Evolutionary Computation*, vol.18, no.3, pp. 456-461. IEEE Press.

- Incorporated in Chapter 4.

Contribution	**R. Morgan (%)**	**M. Gallagher (%)**
Conceptual ideas	90	10
Experimental design	95	5
Experimental implementation	100	0
Data analysis	80	20
Paper write-up	85	15

R. Morgan and M. Gallagher (2014). Fitness Landscape Analysis of Circles in a Square Packing Problems. In *Simulated Evolution And Learning (SEAL 2014)*. pp. 455-466. Springer.

- Incorporated in Chapter 7 and Appendix C.

Contribution	**R. Morgan (%)**	**M. Gallagher (%)**
Conceptual ideas	70	30
Experimental design	80	20
Experimental implementation	100	0
Data analysis	80	20
Paper write-up	80	20

Contributions by others to the thesis

No contributions by others.

Statement of parts of the thesis submitted to qualify for the award of another degree

None.

Acknowledgements

This thesis is the culmination of four challenging yet rewarding years of intellectual endeavor. My candidature was truly the candidature most graduate students dream of, and a large part of this is due to my supportive, encouraging and selfless advisory team; Associate Professor Marcus Gallagher and Dr Daniel Angus. First and foremost, I would like to thank my principal supervisor, mentor and friend Marcus. I am particularly grateful for the free-reign Marcus gave me throughout my studies; his trust and patience was unwavering and gave me the confidence to follow strange and exciting research directions. I have learned immensely from Marcus' attention to detail, strong ethics and commitment to high quality research. Together we encountered many interesting challenges, and I am incredibly grateful for his understanding, wisdom and guidance throughout such times. I would also like to thank my associate supervisor and friend, Dan. Dan was continually a source of encouragement and inspiration. His passion for the communication of quality science is contagious, and would often motivate me in times of hardship. I admire and have learned from his innate ability to simplify highly complex thoughts and ideas into beautifully constructed concepts. I am also thankful for his insight and advice regarding a career (and indeed life!) after my candidature.

I would also like to thank all my colleagues and peers at the University of Queensland. Professor Janet Wiles continually inspired me to learn more about the world (not just my own research area!) and ensured my candidature progressed smoothly. Dr Ian Wood and Dr Minh Duc Cao listened to many of my ideas and gave invaluable feedback and advice, for which I am very grateful. I had the pleasure of sharing an office throughout the majority of my candidature with Krishna Mishra, Kirill Makukhin, Scott Heath and Ting Ting Gibson. I treasure our frequent intellectual discussions and debates, as well as our hilarious conversations that seemed to have no imaginative bounds.

I have been incredibly fortunate to meet and converse with many talented researchers at conferences, workshops and institutional visits. I extend a special thanks to Mario Andrés

Muñoz for our numerous technical discussions and debates, his insightful feedback on work within this thesis, as well as his constant encouragement and gentle reassurances. I would also like to extend my gratitude to the Centrum Wiskunde & Informatica (CWI) for graciously hosting me for a month abroad during my candidature. In particular, I would like to thank Professor Paul Vitányi for patiently and enthusiastically teaching me the intricate details of Information Distance and other Kolmogorov Complexity Theory concepts, as well as providing invaluable feedback on the work in this thesis. I was completely overwhelmed by Paul's genuine interest and enthusiasm for my work, as well as his hospitality and kindness. I would also like to thank Dr Peter Bosman for our many lively discussions regarding the intersection of our research interests, my career aspirations, and more generally, life. While much of the work we produced together is not included in this thesis, I am sincerely grateful for his insights into Estimation of Distribution Algorithms and solving large problems of practical interest. I extend my thanks to Dr Dirk Thierens for his advice regarding combinatorial optimization problems, as well as his creative ideas for future applications of my work. Lastly, I would like to acknowledge the contribution of the Dagstuhl Theory of Evolutionary Algorithms Seminar 13271 in the development of the work in this thesis.

I am very grateful for the scholarships, awards and grants that I received throughout my candidature, thereby allowing me to focus on my studies and collaborate with internationally renown researchers. This work was directly supported by an Australian Postgraduate Award, and partially supported by a Department Scholarship from the School of Information Technology and Electrical Engineering at the University of Queensland. I would like to acknowledge the Institute of Electrical and Electronics Engineers for granting me a travel scholarship to attend the 2011 Machine Learning Summer School, as well as the University of Queensland Richard Jago Memorial prize for sponsoring my visit to CWI. I also extend my gratitude to Google for awarding me the 2011 Anita Borg Scholarship, providing valuable leadership training and networking opportunities, and sponsoring me to attend the 2014 Grace Hopper Conference.

On a more personal note, I would like to profoundly thank my family and friends for their endless encouragement and support throughout my candidature. In particular, I would like to thank my mother, Ann Morgan, for tirelessly fostering my curiosity and instilling a self-resilience that I have no doubt carried me through this endeavor. I would also like to thank my father, Tony Morgan, for his endless enthusiasm and belief in my abilities. I am especially thankful for the encouragement and advice from my siblings - Ben, Daniel and Kate Morgan - as well as their friendly (but fierce!) competitiveness. Lastly, I would like to thank Aaron Tighe, who has been a solid pillar of love and support and continually brings out the best in me. Words cannot fully describe how thankful I am for his kindness, patience and hilarity throughout the many challenges we have faced together. While this thesis concludes a chapter in our lives, it is simply one chapter of many, in a long and happy story.

Keywords

length scale, black-box optimization, problem analysis, fitness landscapes, information distance

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 080108, Neural, Evolutionary and Fuzzy Computation, 70%
ANZSRC code: 080201, Analysis of Algorithms and Complexity, 15%
ANZSRC code: 080401, Coding and Information Theory, 15%

Fields of Research (FoR) Classification

FoR code: 0801, Artificial Intelligence and Image Processing, 70%
FoR code: 0802, Computation Theory and Mathematics, 15%
FoR code: 0806, Data Format, 15%

Contents

Abstract i

Acknowledgements vii

List of Figures xxiv

List of Tables xxv

List of Algorithms xxvi

List of Acronyms xxvii

List of Symbols xxviii

1 Introduction 1

1.1 Background 1

1.2 Research Motives and Contributions 3

1.3 Scope and Limitations 5

1.4 Thesis Outline 5

2 Optimization and Fitness Landscapes 8

2.1 The Optimization Problem 8

2.1.1 Bounds and Constraints 9

2.1.2 Linearity, Convexity and Smoothness 10

2.1.3 Black-Box Optimization 10

2.1.4 The Neighbourhood and Related Concepts 11

2.2 Fitness Landscapes 13

2.2.1 Common Landscape Terminology 16

2.3 Summary 17

3 Landscape Analysis **19**
3.1 No Free Lunch, NP-Completeness and Problem Difficulty 20
3.2 Landscape Features 21
3.2.1 Topological, Minimum Embedding and Fractal Dimensions 22
3.2.2 Local Optima and Basins of Attraction Based Features 23
3.2.3 Probabilistic Based Features 24
3.2.4 Correlation Based Features 25
3.2.5 Information Based Features 27
3.2.6 Other Features 30
3.3 Combinatorial vs Continuous Feature Development 32
3.4 Measuring Problem Similarity 33
3.5 Landscape Analysis in Other Scientific Disciplines 35
3.6 Summary 36

4 Sampling in Continuous Spaces **38**
4.1 Methods and Techniques 38
4.2 Important Considerations 42
4.3 Lévy Random Walks 47
4.4 Case Study: Dispersion 50
4.4.1 Effects of Other Sampling Methodologies and Distances 54
4.4.2 Summary 60
4.5 Case Study: Fitness Distance Correlation 61
4.5.1 Effects of Other Sampling Methodologies and Distances 65
4.5.2 Summary 74
4.6 Summary 74

5 Length Scales in Optimization **76**
5.1 Length Scale 76
5.1.1 Effect of the Distance Function 77
5.1.2 Analytical Length Scale Expressions 77
5.2 Properties of Length Scale 80
5.3 Length Scale Distribution 82
5.3.1 Probability Density Estimation 84
5.4 Properties of the Length Scale Distribution 89

5.5 Related Work 97
5.6 Summary 99

6 Length Scale Analysis: Techniques and Practical Considerations 100
6.1 Analysing Length Scales 100
6.1.1 Heatmaps of Length Scale Values 101
6.1.2 Length Scale Set 105
6.1.3 Length Scale Distribution 108
6.2 Sampling Length Scales in Practice 113
6.2.1 Assessing Sampling Adequacy with Length Scale Analysis 115
6.3 Practical Considerations 118
6.4 Summary 119

7 Length Scale Analysis: Results 121
7.1 Analysis of Continuous Artificial Problems 121
7.1.1 Elliptical Function 121
7.1.2 Rastrigin 126
7.2 Analysis of BBOB Problems 130
7.2.1 Length Scale Distribution Results 132
7.2.2 Results Comparing Length Scale to Existing Features 140
7.2.3 Results Comparing Length Scale with an Ensemble of Features 147
7.2.4 Relationship to Problem Difficulty 151
7.3 Analysis of Circle in a Square Problems 158
7.3.1 Length Scale Distribution Results 162
7.3.2 Results Comparing Length Scale to Existing Features 164
7.3.3 Results Comparing Length Scale with an Ensemble of Features 170
7.4 Analysis of TSPLib 173
7.4.1 Length Scale Analysis Results 174
7.4.2 Existing Results 180
7.5 Analysis of Asymmetric Travelling Salesman Problems 181
7.5.1 Length Scale Analysis Results 182
7.5.2 Existing Results 188
7.6 Analysis of Number Partitioning Problems 188
7.6.1 Length Scale Analysis Results 189

7.6.2 Existing Results 194
7.7 Summary 197

8 Quantifying Problem Similarity with Information Distance 199
8.1 Kolmogorov Complexity Theory 200
8.1.1 Kolmogorov Complexity 200
8.1.2 Information Distance 201
8.1.3 Normalised Compression Distance 202
8.2 Normalised Compression Distance Between Optimization Problems 204
8.2.1 Representing Optimization Problems with Length Scale 204
8.2.2 Suitable Compression Algorithms 206
8.2.3 Computational Complexity 209
8.3 Results 210
8.3.1 Elliptical Functions 211
8.3.2 Rastrigin 215
8.3.3 Griewank 216
8.3.4 Black-Box Optimization Benchmarking Problems 220
8.3.5 Circle in a Square Problems 229
8.3.6 TSPLib 230
8.3.7 Asymmetric Travelling Salesman Problems 235
8.3.8 Number Partitioning Problems 238
8.4 Quantifying Problem Similarity: Kolmogorov vs Shannon 242
8.4.1 Kolmogorov Complexity vs Shannon Entropy 242
8.4.2 Results: Normalised Compression Distance vs Jeffrey Divergence . . . 243
8.4.3 Discussion 248
8.5 Summary 249

9 Conclusion 251
9.1 Summary and Conclusions 251
9.2 Limitations and Future Work 254

References 274

Appendices 275

A Test Problems 276

A.1 Continuous Problems . . . 276

A.1.1 Artificial Problems . . . 276

A.1.2 Black-Box Optimization Benchmarking Set . . . 278

A.1.3 Circle Packing in a Square . . . 279

A.2 Combinatorial Problems . . . 281

A.2.1 Travelling Salesman Problem . . . 281

A.2.2 Number Partitioning Problem . . . 283

B Definitions and Formulae for Combinatorial Problems 286

C Length Scale Analysis of Circle in a Square Problems 287

C.1 Results Comparing Length Scale to Existing Features . . . 287

C.1.1 Fitness Distance Correlation . . . 287

C.1.2 Dispersion . . . 291

C.1.3 Information Content, Partial Information Content and Information Stability . . . 292

C.2 Interpreting Length Scale Distributions of the Circle in a Square Problems . . 294

List of Figures

2.1 Problem landscape of the Michalewicz function defined over $\mathcal{S} = [0, \pi]^2$. 15

4.1 Four solutions sampled from $\mathcal{S} = [0, 1]^2$ using uniform random, systematic, Latin hypercube and orthogonal sampling. 41

4.2 Behaviour of $\frac{\max(\|\mathbf{x}^i - \mathbf{x}^j\|)}{\min(\|\mathbf{x}^i - \mathbf{x}^j\|)}$ for $100D$, $1000D$, $10000D$ and $1000D^2$ points randomly sampled from $\mathcal{U}[0, 1]^D$, where $D = 1, 2, \dots, 30, 40, 50$. 46

4.3 Example Lévy distributions. 47

4.4 Examples of Lévy random walks in $[0, 1]^2$. 48

4.5 Behaviour of $\frac{\max(\|\mathbf{x}^i - \mathbf{x}^j\|)}{\min(\|\mathbf{x}^i - \mathbf{x}^j\|)}$ for $1000D$ points sampled from a Lévy random walk in $[0, 1]^D$, where $D = 1, 2, \dots, 30, 40, 50$. 50

4.6 Behaviour of dispersion for the Sphere (solid lines) and Reverse Sphere (dashed lines) functions. 53

4.7 Dispersion for the BBOB problem set. 53

4.8 Bound-normalised dispersion for the Sphere (blue line) and Reverse Sphere (red line) functions. 55

4.9 Bound-normalised dispersion for the BBOB problem set. 56

4.10 Dispersion using L_p norms where $p = 0.1$, 0.5, 1 and 2 for the Sphere (solid lines) and Reverse Sphere (dashed lines) functions. 57

4.11 Dispersion using L_p norms where $p = 0.1$, 0.5, 1 and 2 for the BBOB problem set. 57

4.12 Dispersion using fixed-step (of size 10^{-2}) random walks for the Sphere (blue line) and Reverse Sphere (red line) functions. 58

4.13 Dispersion using fixed-step (of size 1) random walks for the BBOB problem set. 59

4.14 Dispersion using a Lévy random walk for the Sphere (blue line) and Reverse Sphere (red line) functions. 59

4.15 Dispersion using a Lévy random walk for the BBOB problem set. 60

4.16 $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ for the Sphere and Reverse Sphere functions for $D = 2, \dots,$ $50, 100, 150, 200$. . . . 63

4.17 $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ (using L_2) for the BBOB problem set for $D = 2, \dots, 50, 100,$ $150, 200$. . . . 64

4.18 $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ using $L_{0.1}$, $L_{0.5}$, L_1 and L_2 for the Sphere and Reverse Sphere functions. . . . 66

4.19 $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ using $L_{0.1}$ for the BBOB problem set for $D = 2, \dots, 50, 100,$ $150, 200$. . . . 67

4.20 $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ using $L_{0.5}$ for the BBOB problem set for $D = 2, \dots, 50, 100,$ $150, 200$. . . . 68

4.21 $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ using L_1 for the BBOB problem set for $D = 2, \dots, 50, 100,$ $150, 200$. . . . 69

4.22 FDC using fixed-step (of size 10^{-2}) random walks for the Sphere and Reverse Sphere functions. . . . 70

4.23 FDC using fixed-step (of size 1) random walks for the BBOB problem set. . . . 71

4.24 FDC using a Lévy random walk for the Sphere and Reverse Sphere functions. 72

4.25 FDC using a Lévy random walk for the BBOB problem set. . . . 73

5.1 $p(r)$ for $f(x) = 2x$. . . . 83

5.2 $p(r)$ for $f(x) = 2x^2, x \in [-10, 10]$. . . . 84

5.3 Kernel density estimator with a Gaussian kernel for $X = [1, 4, 4.5, 6, 10]$ and $h = 1.5$. . . . 87

5.4 Examples of multimodal length scale distributions. . . . 90

5.5 $n = 10$ with sequentially assigned pieces. . . . 91

5.6 $n = 30$ with sequentially assigned pieces. . . . 92

5.7 $n = 100$ with sequentially assigned pieces. . . . 93

5.8 $n = 10$ with uniform randomly assigned pieces. . . . 94

5.9 $n = 30$ with uniform randomly assigned pieces. . . . 95

5.10 $n = 100$ with uniform randomly assigned pieces. . . . 96

6.1 Heatmap of r for $f(x) = 2x^2, x \in [-10, 10]$. The blue line indicates where $x^i = x^j$, and hence r is undefined. . . . 102

6.2 1-D mixed-structure function enumerated at increments of 10^{-3}. . . . 104

6.3 Sorted length scale multisets for the 1-D quadratic (where $a = 2$) and 1-D mixed-structure functions. . . . 106

6.4 Example length scale distributions. . . . 109

6.5 Estimating sampling adequacy via convergence of $\hat{p}(r)$. . . . 117

7.1 Example length scale distributions for ellipse functions with $a = 1, 5.5$ and 10. 122

7.2 Heatmap of D_J values calculated between pairs of 2-D ellipse functions, where $a = 1, 1.25, \ldots, 10$. . . . 123

7.3 t-SNE of D_Js (cost of 0.2040) calculated between pairs of 2-D ellipse functions, where $a = 1, 1.25, \ldots, 10$. . . . 124

7.4 Dendrogram of D_Js calculated between pairs of 2-D ellipse functions, where $a = 1, 1.25, \ldots, 10$. . . . 125

7.5 Example length scale distributions for 1-D Rastrigin. . . . 126

7.6 Heatmap of D_J values calculated between pairs of 1-D Rastrigin functions with $A = 0, 0.25, \ldots, 10$. . . . 127

7.7 t-SNE of D_Js (cost of 0.1303) calculated between pairs of 1-D Rastrigin functions with $A = 0, 0.25, \ldots, 10$. . . . 127

7.8 Dendrogram of D_Js calculated between pairs of 1-D Rastrigin functions with $A = 0, 0.25, \ldots, 10$. . . . 129

7.9 D_{KL} between $p(r)$s from subsequent sample sizes $1000D^2$, $5000D^2$ and $10000D^2$ on the BBOB problems in 20-D. . . . 132

7.10 Ellipsoidal Function in 2-D. . . . 133

7.11 Ellipsoidal Function in 5-D. . . . 133

7.12 Ellipsoidal Function in 10-D. . . . 134

7.13 Ellipsoidal Function in 20-D. . . . 134

7.14 Rastrigin Functions in 2-D. . . . 135

7.15 Rastrigin Functions in 5-D. . . . 136

7.16 Rastrigin Functions in 10-D. . . . 136

7.17 Rastrigin Functions in 20-D. . . . 136

7.18 Rosenbrock Functions in 2-D. . . . 137

7.19 Rosenbrock Functions in 5-D. . . . 137

7.20 Rosenbrock Functions in 10-D. . . . 138

7.21 Rosenbrock Functions in 20-D. . . . 138

7.22 Ellipsoidal, Rastrigin and Rosenbrock Functions in 2-D. . . . 139

7.23 Rastrigin and Rosenbrock Functions in 2-D. . . . 140

7.24 Correlation Length of the BBOB problem set in $D = 2, 5, 10$ and 20. . . . 140

7.25 Dispersion of the BBOB problem set in $D = 2, 5, 10$ and 20. 141

7.26 $FDC_{\mathbf{x}^*}$ of the BBOB problem set in $D = 2, 5, 10$ and 20. 142

7.27 $FDC_{\hat{\mathbf{x}}^*}$ of the BBOB problem set in $D = 2, 5, 10$ and 20. 142

7.28 Information content of the BBOB problem set in $D = 2, 5, 10$ and 20. 143

7.29 Partial information content of the BBOB problem set in $D = 2, 5, 10$ and 20. . . 143

7.30 Information stability of the BBOB problem set in $D = 2, 5, 10$ and 20. 144

7.31 Length scale entropy of the BBOB problem set in $D = 2, 5, 10$ and 20. 145

7.32 Maximum r of the BBOB problem set in $D = 2, 5, 10$ and 20. 145

7.33 Median of r of the BBOB problem set in $D = 2, 5, 10$ and 20. 146

7.34 Mean of r of the BBOB problem set in $D = 2, 5, 10$ and 20. 146

7.35 Variance of r of the BBOB problem set in $D = 2, 5, 10$ and 20. 146

7.36 Feature spaces of 20-D BBOB problems reduced via t-SNE. 149

7.37 Dendrograms of the 20-D BBOB problems. 150

7.38 Relationship between correlation length and the best ERT. 154

7.39 Relationship between dispersion and the best ERT. 154

7.40 Relationship between $FDC_{\mathbf{x}^*}$ and the best ERT. 155

7.41 Relationship between $FDC_{\hat{\mathbf{x}}^*}$ and the best ERT. 155

7.42 Relationship between information content and the best ERT. 156

7.43 Relationship between partial information content and the best ERT. 156

7.44 Relationship between information stability and the best ERT. 157

7.45 Relationship between length scale and the best ERT. 157

7.46 Circle centres of the optimal Circle in a Square packings for similar n_C that show very different arrangements of circle centres. 159

7.47 Circle centres of the optimal Circle in a Square packings for similar n_C that show very similar arrangements of circle centres. 160

7.48 Four globally optimal packings for $n_c = 2$. The centres of each circle are shown, where circles 1 and 2 and represented by □ and ◇ respectively. 161

7.49 Length scale distributions for $n_c = 2, 3, 4$ and 5. 162

7.50 Length scale distributions for $n_c = 97, 98, 99$ and 100. 163

7.51 Ratio of the mode and 99th percentile of r, suggesting a non-uniform decrease in r values as n_c increases. 163

7.52 Correlation Length of the CiaS problems for $n_c = 2, \ldots, 100$. 164

7.53 Dispersion of the CiaS problems for $n_c = 2, \ldots, 100$. 165

7.54 $FDC_{\mathbf{x}^*}$ of the CiaS problems for $n_c = 2, \ldots, 100$. 165

7.55 $FDC_{\hat{\mathbf{x}}^*}$ of the CiaS problems for $n_c = 2, \ldots, 100$. 166

7.56 Information content of the CiaS problems for $n_c = 2, \ldots, 100$. 167

7.57 Partial information content of the CiaS problems for $n_c = 2, \ldots, 100$. 167

7.58 Information stability of the CiaS problems for $n_c = 2, \ldots, 100$. 168

7.59 Length scale entropy of the CiaS problems for $n_c = 2, \ldots, 100$. 169

7.60 Feature spaces of CiaS problems ($n_c = 2, \ldots, 100$) reduced via t-SNE. 171

7.61 Dendrograms of the CiaS problems ($n_c = 2, \ldots, 100$) problems. 172

7.62 Length scale distributions of "kro"-type instances from TSPLib. 175

7.63 Length scale distributions of "bay"-type instances from TSPLib. 175

7.64 Length scale distributions of "gr"-type instances from TSPLib. 175

7.65 Length scale distributions of "ft"-type instances from TSPLib. 176

7.66 Length scale distributions of the TSPLib instances. 176

7.67 Zoomed-in length scale distributions taken from Figure 7.66. 176

7.68 Length scales for instance p43. 177

7.69 Heatmap of D_J values calculated between pairs of TSPLib instances. 178

7.70 t-SNE of D_Js (cost of 0.2041) calculated between pairs of TSPLib instances. . . 178

7.71 Dendrogram of D_Js calculated between pairs of TSPLib instances. 179

7.72 Proportion of unique distances averaged across 10 random instances of each b. 182

7.73 The change in shape of $p(r)$ as b increases. 183

7.74 ATSP distance distributions for the ATSP instances. 184

7.75 Heatmap of D_J values calculated between pairs of ATSP instances with $b = 0.5, 0.6, \ldots, 6.5$. 185

7.76 t-SNE of D_Js (cost of 0.2473) calculated between pairs of ATSP instances with $b = 0.5, 0.6, \ldots, 6.5$. 186

7.77 Dendrogram of D_Js calculated between pairs of ATSP instances with $b = 0.5, 0.6, \ldots, 6.5$. 187

7.78 Length scale sets for the NPP instances. Lines are coloured according to k, with blue representing $k = 0.4$ and red representing $k = 1.3$. 190

7.79 Hamming distance distributions for the NPP instances. 190

7.80 Length scale sets for the NPP instances normalised by 2^{nk}. Lines are coloured according to k, with blue representing $0.4 \leq k \leq 0.8$, green representing $0.8 < k \leq 1$ and red representing $1 < k \leq 1.3$. 191

7.81 Example $p(r)$s for the 10 NPP instances normalised by 2^{nk}. Lines are coloured according to k, with blue representing $k = 0.4$, green representing $k = 0.925$ and red representing $k = 1.3$. 192

7.82 Heatmap of D_J values calculated between pairs of NPP instances with $k = 0.4, 0.425, \ldots, 1.3$. 193

7.83 t-SNE of D_Js (cost of 0.2272) calculated between pairs of NPP instances with $k = 0.4, 0.425, \ldots, 1.3$. 194

7.84 Dendrogram of D_Js (cost of 0.2041) calculated between pairs of NPP instances with $k = 0.4, 0.425, \ldots, 1.3$. 195

8.1 Heatmap of NCD values calculated (using LZMA) between pairs of 2-D elliptical functions, where $a = 1, 1.25, \ldots, 10$. 212

8.2 t-SNE of NCD values (cost of 0.1451) calculated (using LZMA) between pairs of 2-D elliptical functions, where $a = 1, 1.25, \ldots, 10$. 213

8.3 Dendrogram of NCD values calculated (using LZMA) between pairs of 2-D elliptical functions, where $a = 1, 1.25, \ldots, 10$. 214

8.4 Heatmap of NCD values calculated (using LZMA) between pairs of 1-D Rastrigin functions, where $A = 0, 0.25, \ldots, 10$. 215

8.5 t-SNE of NCD values (cost of 0.3205) calculated (using LZMA) between pairs of 1-D Rastrigin functions, where $A = 0, 0.25, \ldots, 10$. 216

8.6 Dendrogram of NCD values calculated (using LZMA) between pairs of 1-D Rastrigin functions, where $A = 0, 0.25, \ldots, 10$. 217

8.7 t-SNE of NCD values (cost of 0.0497) calculated (using LZMA) between pairs of Griewank instances, where $D = 1, 2, \ldots, 15$. "G"denotes Griewank functions, while 'C"denotes the convex component. 218

8.8 Heatmap of NCD values calculated (using LZMA) between pairs of Griewank instances, where $D = 1, 2, \ldots, 15$. "G"denotes Griewank functions, while 'C"denotes the convex component. 218

8.9 Dendrogram of NCD values calculated (using LZMA) between pairs of Griewank instances, where $D = 1, 2, \ldots, 15$. "G"denotes Griewank functions, while 'C"denotes the convex component. 219

8.10 Heatmap of NCD values calculated (using LZMA) between pairs of 2-D BBOB instances. 221

8.11 Heatmap of NCD values calculated (using LZMA) between pairs of 5-D BBOB instances. 221

8.12 Heatmap of NCD values calculated (using LZMA) between pairs of 10-D BBOB instances. 222

8.13 Heatmap of NCD values calculated (using LZMA) between pairs of 20-D BBOB instances. 222

8.14 t-SNE of NCD values (cost of 0.0611) calculated (using LZMA) between pairs of 2-D BBOB instances. 223

8.15 t-SNE of NCD values (cost of 0.0534) calculated (using LZMA) between pairs of 5-D BBOB instances. 224

8.16 t-SNE of NCD values (cost of 0.0443) calculated (using LZMA) between pairs of 10-D BBOB instances. 224

8.17 t-SNE of NCD values (cost of 0.0370) calculated (using LZMA) between pairs of 20-D BBOB instances. 225

8.18 Dendrogram of NCD values calculated (using LZMA) between pairs of 2-D BBOB instances. 226

8.19 Dendrogram of NCD values calculated (using LZMA) between pairs of 5-D BBOB instances. 226

8.20 Dendrogram of NCD values calculated (using LZMA) between pairs of 10-D BBOB instances. 227

8.21 Dendrogram of NCD values calculated (using LZMA) between pairs of 20-D BBOB instances. 227

8.22 Heatmap of NCD values calculated (using LZMA) between pairs of 2, 5, 10 and 20-D BBOB instances. 228

8.23 t-SNE of NCD values (cost of 0.4136) calculated (using LZMA) between pairs of 2, 5, 10 and 20-D BBOB instances. 228

8.24 Heatmap of NCD values calculated (using LZMA) between pairs of CiaS instances, where $n_c = 2, 3, \ldots, 30$. 230

8.25 t-SNE of NCD values (cost of 0.0885) calculated (using LZMA) between pairs of CiaS instances, where $n_c = 2, 3, \ldots, 30$. 231

8.26 Dendrogram of NCD values calculated (using LZMA) between pairs of CiaS instances, where $n_c = 2, 3, \ldots, 30$. 232

8.27 Heatmap of NCD values calculated (using LZMA) between pairs of TSPLib instances. 233

8.28 t-SNE of NCD values (cost of 0.1026) calculated (using LZMA) between pairs of TSPLib instances. 233

8.29 Dendrogram of NCD values calculated (using LZMA) between pairs of of TSPLib instances. 234

8.30 Heatmap of the mean NCD values calculated (using LZMA) between pairs of random ATSP instances, generated with $b = 0, 0.1, \ldots 6.5$. 236

8.31 t-SNE of the mean NCD values (cost of 0.0419) calculated (using LZMA) between pairs of random ATSP instances, generated with $b = 0, 0.1, \ldots 6.5$. . . . 236

8.32 Dendrogram of the mean NCD values calculated (using LZMA) between pairs of random ATSP instances, generated with $b = 0, 0.1, \ldots 6.5$. 237

8.33 Heatmap of the NCD values calculated (using LZMA) between pairs of NPP instances, where $k = 0.4, 0.425, \ldots, 1.4$. 239

8.34 t-SNE of the NCD values (cost of 0.1158) calculated (using LZMA) between pairs of NPP instances, where $k = 0.4, 0.425, \ldots, 1.4$. 239

8.35 Dendrogram of the NCD values calculated (using LZMA) between pairs of NPP instances, where $k = 0.4, 0.425, \ldots, 1.4$. 240

8.36 NCD vs D_J for 2-D elliptical functions, where $a = 1, 1.25, \ldots, 10$. 244

8.37 NCD vs D_J for 1-D Rastrigin functions, where $A = 0, 0.25, \ldots, 10$. 244

8.38 NCD vs D_J for 2, 5, 10 and 20-D BBOB problems. 245

8.39 NCD vs D_J for Circle in a Square problems, where $n_c = 2, 3, \ldots, 100$. 245

8.40 NCD vs D_J for TSPLib instances. 245

8.41 NCD vs D_J for Asymmetric Travelling Salesman Problem instances, randomly generated with $b = 0, 0.1, \ldots 6.5$. 246

8.42 NCD vs D_J for Number Partitioning Problem instances, where $k = 0.4, 0.425, \ldots, 1.4$. 246

C.1 FDC for CiaS problems. Lines show the mean of the 30 trials, while error bars indicate one standard deviation. 288

C.2 Typical fitness-distance scatter plots for $n_c = 2$ circles. 290

C.3 Typical fitness-distance scatter plots using $\mathbf{x}^*$ as the global optimum. 291

C.4 Bound-Normalised dispersion for CiaS problems as n_c increases. 292

C.5 Information-Theoretic Analysis for CiaS problems as n_c increases. 293

C.6 $h(r)$ for CiaS problems as n_c increases. 294

C.7 The change in shape of $p(r)$ as n_c increases. 295

C.8 Ratio of the mode and 99th percentile of r, confirming the significant change in the shape of $p(r)$ as n_c increases. 296

List of Tables

6.1 Sample Estimates of Common Statistical Measures 110

7.1 Pearson's correlation coefficient, ρ_p, between BBOB ERT values and problem metrics. 153

8.1 Average Compression Ratios 208
8.2 Normality Properties 209
8.3 Correlation coefficients between NCD and D_J with $\alpha = 0.05$ 247

A.1 Objective Functions 277
A.2 Black-Box Optimization Benchmarking Problem Set 280
A.3 TSPLib Instances 284

List of Algorithms

2.1 Local Search (LS) 13

4.1 Levy Random Walk 49

6.1 Generation of the length scale multiset 114

8.1 Normalised Compression Distance (NCD) 206

List of Acronyms

ATSP	Asymmetric Travelling Salesman Problem
CEC	Congress on Evolutionary Computation
BBOB	Black-Box Optimization Benchmarking
CiaS	Circle in a Square
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
ERT	Expected Running Time
FDC	Fitness Distance Correlation
FLAC	Free Lossless Audio Compressor
GA	Genetic Algorithm
LH	Latin Hypercube
LS	Local Search
LZMA	Lempel-Ziv-Markov chain Algorithm
MIDI	Musical Instrument Digital Interface
NCD	Normalised Compression Distance
NFL	No Free Lunch
NIAH	Needle In A Haystack
NID	Normalised Information Distance
NP	Non-deterministic Polynomial-time
NPP	Number Partitioning Problem
PSO	Particle Swarm Optimization
TSP	Travelling Salesman Problem
TSPLib	Travelling Salesman Problem benchmarking Library
t-SNE	t-Distributed Stochastic Neighbour Embedding
VI	Variation of Information

List of Symbols

f	Objective function
$\mathcal{S}$	Search space
$\mathbf{x}$	Candidate solution
$\mathbf{x}^*$	Globally optimum solution
d	Distance function
$\mathcal{L}$	Problem landscape
N	Neighbourhood function
D	Dimension
$\mathcal{U}$	Uniform distribution
$\mathbb{Z}$	Set of integers
$\mathbb{R}$	Set of reals
$\mathbb{N}$	Set of natural numbers
$\|A\|$	Cardinality of set A
$A \cup B$	Union of sets A and B
$A \cap B$	Intersection between sets A and B
$A \setminus B$	Relative compliment of set B in A
$\binom{n}{k}$	$\frac{n!}{k!(n-k)!}$
μ	Mean
σ	Standard deviation
γ	Scale parameter for Lévy distribution
δ	Location parameter for Lévy distribution
Θ	Model parameters
ρ_p	Pearson's correlation coefficient
ρ_s	Spearman's correlation coefficient
τ	Kendall's correlation coefficient

r	Length scale value
$p(r)$	Length scale distribution
$h(r)$	Entropy of the length scale distribution
D_J	Jeffrey divergence between two length scale distributions
U	Universal Prefix Turing Machine
$K(x)$	Kolmogorov complexity of string x
Z	Compressor

CHAPTER 1

Introduction

The most exciting phrase to hear in science,
the one that heralds new discoveries, is not
'Eureka!', but 'That's funny...'
Isaac Asimov

This chapter introduces the research area considered in this thesis. A general background to the research area is provided in Section 1.1. Section 1.2 considers important research issues, defines the motivations of the thesis and summarises its major contributions. Section 1.3 discusses the scope of the thesis and limitations of the work. An outline of the thesis structure is given in Section 1.4.

1.1 Background

The optimization of resources according to some criterion or objective, such as maximising efficiency or minimising waste, is a universal problem found in almost every aspect of human endeavour. Humans and animals exhibit remarkable abilities in solving conceptually small optimization problems, such as finding the shortest path between rooms in a building. However, many real-world problems relevant to industrial and commercial applications are extremely large and complex, and are thus beyond the capabilities of humans to solve exactly. These types of problems include optimizing the aerodynamic drag and structural weight of an aircraft wing [191], the placement of wind turbines to maximise energy [195], and the selection of stocks in a portfolio in order to maximize its return on investment [24].

Optimization can be stated generally: given a number of variables, $\mathbf{x} = (x_1, \ldots, x_D)$ and some (objective) function of those variables, $f(\mathbf{x})$, determine values for the variables that minimize (or maximize) the corresponding value of f. Optimization problems can be categorised based on the types of their variables; *continuous* problem solutions are entirely comprised of continuous/real-valued variables, while combinatorial problem solutions are comprised of discrete-valued variables. An important class of optimization problems are known as “black-box”, where the only information available about a problem to a solver is the ability to evaluate the objective function value of candidate solutions. Black-box problems typically occur in the optimization of the output of an unknown system or mechanism, such as a simulation.

Assuming that an optimization problem can be expressed or abstracted in computational form, computer algorithms can be employed to solve the problem *heuristically*. That is, given a candidate solution, decisions are made (based on predefined heuristics) to determine areas of potentially high quality solutions, which are then explored in greater detail. The heuristics are a critical component to the success of heuristic algorithms, and are typically based on the decision process used by human solvers. For example, the commonly-used *greedy* algorithm generates a list of solutions from a “base” solution, replaces the base solution with the *best* solution in the list, and repeats this process until no solutions are found that are better than the base solution [181]. The advantage of a computational approach is that heuristic decisions can be evaluated and acted on typically much faster and with less error than a human.

Metaheuristic algorithms are a class of algorithms that utilise multiple heuristics in order to efficiently solve optimization problems. The heuristics within metaheuristic algorithms are often inspired by real world phenomenon where systems are “optimized” (i.e. improved) naturally. For example, Genetic Algorithms utilise known evolutionary processes like mutation and reproduction to “evolve” solutions towards a higher quality [138]. Ant Colony Optimization algorithms are another nature-inspired group of metaheuristics that model the behaviour and collective intelligence of ants within a colony to “search” for high quality solutions [181]. Other metaheuristics utilise general heuristics that are intuitively conducive towards solving optimization problems. For example, Tabu Search maintains a list of solutions visited during the search to ensure that they are not revisited [181]. Metaheuristic algorithms aim to be general-purpose solvers, and they typically only utilise information afforded by solutions and their respective f-values. Consequently, metaheuristics are highly applicable to black-box optimization problems.

The metaheuristic literature is largely dominated by the development of new and improved metaheuristic algorithms, and as a result, there is an abundance of algorithms. Intuitively, for a particular “class” of problems (nominally, problems that are in some sense more similar to each other than a randomly chosen set of problems), certain specified algorithms are more well-suited (e.g. obtain the best objective function value given a fixed budget of evaluations) than other algorithms. An important research direction is to better understand the relationship between algorithms and the problems to which they are applied. In other words, given a particular problem, what algorithms are likely to efficiently solve it, and why? Alternatively, given a particular algorithm, what problems does it perform well on, and why? Metaheuristic algorithms have been applied successfully to many practical black-box problems [181], however due to the heuristic nature of these algorithms and the lack of problem knowledge, it is often difficult to understand their behaviour. Consequently, there is very little scientific understanding or explanation of the relationship between the performance of metaheuristic algorithms and the problems to which they are applied [59]. Theoretical work in metaheuristic optimization continues to develop but there is currently a significant gap between this and real-world problems.

1.2 Research Motives and Contributions

The notion of a *landscape* is used to model the structural topology induced by the objective function defined over the solution space. Optimization algorithms, including metaheuristics, conceptually solve a problem by "navigating" through the problem's landscape in pursuit of low or high-valued solutions. The problem landscape also provides an abstract framework from which particular problem features, characteristics and properties can be defined and analysed in order to describe and provide insight into the nature of the problem. In practice, expert knowledge and intuition is often used to analyse problems. However, the structural topology of the landscape is unknown in the black-box optimization context, and so the development of features, characteristics and properties that are able to capture (i.e. detect or show sensitivity to) the structural topology is highly challenging. Practitioners are faced with a conundrum; how can one describe something, without knowing what it is?

Fundamentally, total enumeration of the candidate solutions and their respective objective function values completely describes a problem. Hence, problem features developed in the black-box optimization literature commonly utilise candidate solutions and/or their f-values. However, optimization problems of practical interest typically have an enormous number of candidate solutions, and as a result, features and characteristics are defined over finite samples of solutions and their objective function values. While numerous features have been proposed in the literature, there are some important limitations of this work (e.g. many features do not utilise all information afforded by the solutions and their f-values) as well as many open issues around its practical utility.

Collectively, the contributions in this thesis provide a new framework and practical techniques for the analysis of optimization problems by utilizing all information available in the black-box setting. The framework is based on a novel summary of the landscape, called *length scale*, that uniquely describes and hence characterises problems. The length scale framework developed in this thesis is comprised of:

- A technique for assessing the adequacy of a sample from either continuous and combinatorial problems.
- A new sampling methodology for obtaining the length scale information for continuous problems.
- A suite of techniques from set theory, statistics, machine learning, information theory and visualisation to analyse and interpret length scale information.
- Two explicit problem similarity measures.

Specifically, the major contributions of this thesis are as follows:

1. A comprehensive review of the analysis of both combinatorial and continuous black-box problem landscapes (Chapter 3).
2. The identification of theoretical issues relevant to sampling continuous problems, and hence, landscape analysis (Chapter 4).
3. The application of Lévy random walks to reduce and eliminate the identified sampling issues in continuous optimization (Chapter 4).
4. The notion of length scale and derived summaries as problem landscape features for **both** continuous and combinatorial problems (Chapter 5).
5. A methodology to assess the adequacy of a sample from **both** continuous and combinatorial problems (Chapter 6).
6. The application of methods from set theory, statistics, machine learning, information theory and visualisation to analyse and interpret length scale information (Chapter 6).
7. The use of the entropic Jeffrey divergence to quantify the similarity between continuous problems, as well as combinatorial problems (Chapters 6 and 7).

8. The use of the Normalised Compression Distance to quantify the similarity between continuous problems, as well as combinatorial problems (Chapter 8).

9. Investigation into the theoretical and empirical relationship between the Jeffrey divergence and Normalised Compression Distance (Chapter 8).

Source code for all experimental investigations in this thesis is available at `https://github.com/RachM/thesis`.

1.3 Scope and Limitations

This thesis focuses on combinatorial and continuous optimization problems, and as a result, problems with a mixture of discrete and real-valued variables are outside of the thesis' scope. The methodologies developed throughout this dissertation may be directly applicable (or alternatively, adapted) to such problems, however the feasibility and efficacy of this is not explored.

Historically, many optimization problem features have often been developed as indicators for problem difficulty. This research pursuit is problematic for a number of reasons. Firstly, a notion of "difficulty" is highly subjective and varies throughout the literature. Consequently, there is no clear, well-established difficulty measure with which to correlate problem features. Secondly, empirical benchmarking results show that an "easy" problem for one algorithm can be "hard" for another. Fundamentally, a given problem is not globally "easy" or "hard" for all algorithms; the ability of algorithms to navigate particular problem structures varies, and hence it is the structures (and their interactions) that influence algorithm performance. One major contribution of the thesis is the development of features that are shown to capture (i.e. detect or show sensitivity to) the structures within problems. While the features may be predictive of particular algorithms' performances, the focus of the thesis is the development and ability of the features to analyse and compare problems. Hence, notions of problem difficulty are largely absent from the development of the length scale framework, as well as analyses and discussions.

1.4 Thesis Outline

Chapter 2 formally defines optimization and important concepts relevant to the work in this thesis. In addition, a consolidated notion of the problem landscape for a given optimization is defined, and terms from the landscape vernacular are explained.

Chapter 3 reviews existing landscape analysis and comparison techniques formulated predominately in the evolutionary computation community, and significant issues and research gaps are identified. Related landscape analysis techniques from geography, ecology, biology, chemistry and physics are also reviewed.

Landscape analysis techniques rely heavily on finite samples of solutions, and so a focused review of sampling methodologies is provided in Chapter 4. The review raises serious concerns regarding the efficacy of the sampling methodologies commonly employed in high dimensional continuous problem analysis. In an effort to address the concerns, a Lévy random walk is proposed for the analysis of high dimensional continuous problems. Experimental case studies are conducted using two well-known landscape features in order to investigate the practical implications of the concerns, as well as the efficacy of the proposed sampling technique.

Chapter 5 establishes the notion of *length scale* as a fundamental feature of problem landscapes. The length scale values of small, intuitive problems are analysed in order to investigate the ability of the length scale information to capture important problem structure. Concepts developed from length scale, such as the *length scale distribution*, are introduced. Properties of length scale and related concepts are defined, and related work is discussed.

In Chapter 6, Lévy random walks are used to sample length scale information, and a methodology for evaluating the adequacy of the sample is developed. Based on the sample of length scale values, analysis techniques from set theory, statistics, machine learning and visualisation are proposed to analyse and compare problems. One major contribution of this chapter is to propose and demonstrate the use of the Jeffrey divergence to explicitly quantify the similarity between optimization problems.

The length scale analysis techniques developed are experimentally evaluated and compared to several popular landscape analysis techniques in Chapter 7. The techniques are used to analyse and compare continuous artificial problems, well-known benchmark problems from both continuous and combinatorial optimization, geometric packing problems, and two widely studied combinatorial problems.

Chapter 8 proposes an alternative problem similarity measure, based on a universal distance function in Kolmogorov Complexity theory known as Information Distance. A novel methodology to estimate the Information Distance between optimization problems in practice is developed. The methodology utilises a related measure, known as the Normalised Compression Distance, between samples of length scale values. Experimental analysis of the Normalised Compression Distance is conducted on the problems analysed in Chapter 7. The Normalised Compression Distance and Jeffrey divergence are two novel measures of problem similarity developed in this work, and their relationship is theoretically and empirically investigated in Chapter 8.

Chapter 9 concludes the thesis by reflecting on the work's novel contributions, including arguments, concepts, methodologies and results. Limitations of the work and avenues for future work are also discussed.

The optimization problems used throughout the experiments in this thesis are formally defined and described in Appendix A. Certain concepts, definitions and formulae differ between continuous and combinatorial optimization. To assist with readability, continuous definitions are provided in the text, while the related combinatorial versions are given in Appendix B. Appendix C contains an additional experimental investigation comparing the length scale analysis and several popular landscape analysis techniques. The experiment is identical to the analysis of the geometric packing problems in Chapter 7, with the exception that a uniform random sampling technique is used instead of the Lévy random walk.

CHAPTER 2

Optimization and Fitness Landscapes

The one common experience of all humanity
is the challenge of problems.
R. Buckminster Fuller

This chapter formally defines optimization and concepts important to the work in this thesis. In addition, a consolidated notion of optimization problem landscapes is defined, and important terms from the landscape vernacular are reviewed. The aim of this chapter is to familiarise readers with notational conventions and commonly used terminology, and readers who are interested in reviews of optimization in a more general sense can consider the reviews in [76, 181].

2.1 The Optimization Problem

Optimization problems are ubiquitous, and as a result, optimization has been well-studied in a variety of domains including science, engineering, design, management and finance [181]. From a high-level, an optimization problem is defined by two major components: 1) a set of candidate solutions, $\mathcal{S}$, known as the *search space*, and 2) an *objective function*, $f : \mathcal{S} \rightarrow \mathbb{R}$, that measures the quality of a given solution. Without loss of generality, the assumption throughout this dissertation is *minimization*[1] of f.

Definition 2.1 (Global optimum). *A solution* $\mathbf{x}^* \in \mathcal{S}$ *is the* ***global optimum*** *if:*

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \ \ \forall \, \mathbf{x} \in \mathcal{S} \tag{2.1.1}$$

In *global optimization*, the aim is to find the solution, $\mathbf{x}^*$, with the lowest possible objective

[1] Maximization of f is equivalent to the minimization of $-f$.

function value. Multiple solutions may minimise f, and so it is possible to have multiple global optima.

Definition 2.1 involves a single objective function, and such problems are known as *single-objective* optimization problems. Problems with multiple objectives can be formulated using multiple objective functions [181], however multi-objective optimization is outside of the scope of this thesis.

Optimization problems are often categorised based on the specification of $\mathcal{S}$. One very general type of optimization problem is the *combinatorial* problem, where $\mathcal{S}$ consists of a discrete set of inputs, such as people, facilities and machines. The aim is to find a configuration/combination of the input parameters, known as *variables*, such that the objective function is minimised. The generality of this definition allows many real-world problems, such as scheduling, packing and routing problems, to be formulated as combinatorial optimization problems. As an aside, combinatorial optimization problems are closely related to *decision problems* from computational complexity theory [48, 76]. Decision problems are essentially posed as a simple yes/no question, and can often be answered/solved by solving a related optimization problem.

In situations where all of the input parameters of the problem are continuous (i.e. real-valued), the candidate solutions are thus continuous vectors, $\mathbf{x} \in \mathcal{S} \subseteq \mathbb{R}^D$ for a specified D, and the optimization problem is known as *continuous* (or real-valued), with a *dimensionality* of D. Optimization problems over integer-valued and mixtures of integer and real-valued variables are also widely studied [14], but are outside of the scope of this thesis.

2.1.1 Bounds and Constraints

The search space of candidate solutions is often restricted by the specification of *constraints* that explicitly define the feasibility of solutions. Solutions satisfying all specified constraints are known as *feasible*, while solutions that do not satisfy the constraints are *infeasible*. In continuous optimization, *bounds* can also be imposed on the feasible search space. The bounds typically define the minimum and maximum values for each variable in the candidate solution vector, and they are often utilised by solvers to ensure that only solutions within the bounds are evaluated. Optimization problems without bounds are known as *unbounded*, and similarly, problems without constraints are known as *unconstrained*. Given that many real-world optimization problems frequently involve the optimization of physical objects which are inherently subject to physical limitations and constraints, real-world problems are often bounded and can have numerous constraints [181].

2.1.2 Linearity, Convexity and Smoothness

Optimization problems are also classified with respect to certain properties of the objective function. When the analytic form of f is known, problems can be classified based on their linearity, convexity and smoothness [133]. Problems where both f and the constraints are linear are known as *linear programming* problems, while problems with a non-linear f and/or constraints are known as *non-linear*. Similarly, the objective function and constraints in *convex* optimization problems are convex, while *non-convex* problems have a non-convex f and/or non-convex constraints[2]. In continuous optimization, a problem is *smooth* if f and all of the constraints are at least twice differentiable, and problems where the constraints and f are not twice differentiable are known as *non-smooth*. Prior knowledge of linearity, convexity and smoothness can be very advantageous in both analysing and solving optimization problems, and hence there are specific sub-fields of the optimization community devoted to each (e.g. see [102] for introductions to linear and non-linear programming, and [62] for further details on convex, smooth and non-smooth optimization).

2.1.3 Black-Box Optimization

In certain situations, analytical expressions for the objective function and/or constraints are not available, and so the optimization problem cannot be analytically formulated. Without an analytic formulation, the objective function is effectively a *black-box*; solutions can be evaluated by the objective function, but no other information regarding f is provided.

Definition 2.2 (Black-box optimization problem.). *An optimization problem is classed as a* ***black-box*** *problem if [80, 181]:*

1. *$\mathcal{S}$ is defined*
2. *f can be evaluated for each $\mathbf{x} \in \mathcal{S}$*
3. *No other information is known*

Because almost no information is known a priori, black-box problems are analysed and solved by inferring problem structure from the analysis of finite samples of solutions. Indeed, in metamodel or surrogate-based optimization (known as fitness approximation in

[2] Convexity is more general than linearity, and so a linear programming problem is automatically by definition convex.

evolutionary computation), an explicit model (e.g. regression) is built from samples of $\mathcal{S}$ and f, and the model is subsequently used to determine search areas likely to contain optimal solutions [7, 53, 81]. Examples of real-world black-box problems typically occur when f is the output of an unknown system or mechanism, such as simulation and shape optimization [153].

2.1.4 The Neighbourhood and Related Concepts

This section defines and discusses several important optimization problem concepts dependent on the notion of a *neighbourhood*, described in Definition 2.3. The concepts presented in this section are based on the definitions of optimization and related concepts described by Horst and Tuy [75] and Stadler [171].

Definition 2.3. *Let $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ be a distance function between candidate solutions. The* ***neighbourhood*** *for a given solution, $\mathbf{x}'$, yields a set of feasible candidate solutions that are within distance $\epsilon > 0$ of $\mathbf{x}'$:*

$$N(\mathbf{x}') = \{\mathbf{x} \in \mathcal{S} \mid d(\mathbf{x}', \mathbf{x}) \leq \epsilon\} \tag{2.1.2}$$

The choice of distance function, d, is generally dependent on the *representation* of candidate solutions, although distances have been proposed based on the move operators of particular search algorithms [82]. The aim of this thesis is to develop *algorithm independent* landscape analysis techniques, and so "canonical" distances appropriate for the solution representations are used throughout. For example, the candidate solutions in continuous optimization problems are represented by vectors of reals, i.e. $\mathbf{x} \in \mathcal{S} \subseteq \mathbb{R}^D$. Hence, distance metrics defined for continuous spaces, such as the L_p norm, are appropriate. If Euclidean distance (i.e. L_2 norm) is used, $N(\mathbf{x}')$ defines a hypersphere with radius ϵ centred at $\mathbf{x}'$, and assuming infinite precision, there is an infinite number of neighbouring solutions to $\mathbf{x}'$. In combinatorial optimization, vectors of discrete values (including bit-vectors and bit-matrices) and permutation vectors are typically used in formulating problems. For example, candidate solutions of the NP-hard Number Partitioning Problem can be represented using bit-vectors, $\mathbf{x} \in \{0, 1\}^n$ (see Appendix A.2.2). Well-known distances for vectors of discrete values and permutation vectors include Hamming distance, swap distance, edit-distance and the Cayley distance [44]. The neighbourhood defined for a combinatorial optimization problem is a finite set of solutions within ϵ of $\mathbf{x}'$.

An important concept that relates to neighbourhood definitions is the concept of a *local optimum*.

Definition 2.4 (Local optimum.). *A* ***local optimum*** *is a solution,* $\mathbf{x}'$*, where the objective function value are better (i.e. smaller in the context of minimization) than all of its neighbouring solutions. Specifically,* $\mathbf{x}'$*, satisfies:*

$$f(\mathbf{x}') < f(\mathbf{x}), \ \ \forall \, \mathbf{x} \in N(\mathbf{x}') \tag{2.1.3}$$

The inequality in Equation 2.1.3 implies that none of the neighbouring solutions to $\mathbf{x}'$ can also be local optima, and so $\mathbf{x}'$ is known as a *unique* local optimum. A *non-unique* local optimum is defined by relaxing the inequality of Equation 2.1.3:

$$f(\mathbf{x}') \leq f(\mathbf{x}), \ \ \forall \, \mathbf{x} \in N(\mathbf{x}') \tag{2.1.4}$$

Non-unique local optima imply a region of the search space where f is of a constant, locally optimal, value. In general, local optima are typically defined with a small neighbourhood (i.e. minimal ϵ). A global optimum is a special case of a local optimum where *all* solutions in the search space have equal or worse objective function values than its own objective function value. Problems with a single optimum are known as *unimodal*, while problems with multiple local optima are known as *multimodal*.

Let f be a continuous function, and let $f'(\mathbf{x})$ and $f''(\mathbf{x})$ denote the first and second derivatives of f respectively. A candidate solution, $\mathbf{x} \in \mathcal{S}$ is a *stationary point* when f is neither increasing nor decreasing in all directions from $\mathbf{x}$. Formally, $\mathbf{x}$ is a stationary point if $f'(\mathbf{x}) = 0$. Similarly, $\mathbf{x}$ is a *critical point* if it is either stationary, or non-differentiable at $f(\mathbf{x})$. By Fermat's theorem, all optima occur at either the boundaries of f or critical points [176]. *Saddle points* are critical points that are not local optima. Specifically, for a given saddle point $\mathbf{x}_S$, there does not exist a constant, $\varepsilon > 0$, such that $f(\mathbf{x}_S) < f(\mathbf{x})$ where $\|\mathbf{x}_S - \mathbf{x}\| < \varepsilon$.

Closely related to the notion of a local optimum is a *basin of attraction*. The basin of attraction for a given local optimum is loosely defined as the set of solutions that, when an *idealised local search* algorithm is initialised at each solution, leads to the given local optimum [171]. More formally, given the Local Search procedure in Algorithm 2.1, the basin of attraction for a given local optimum, $\mathbf{x}' \in \mathcal{S}$, is $B(\mathbf{x}') = \{\mathbf{x} \in \mathcal{S} \mid \mathrm{LS}(\mathbf{x}) = \mathbf{x}'\}$.

The Local Search procedure in Algorithm 2.1 is not guaranteed to terminate at a particular local optimum in the presence of saddle points. Hence in the above definition of a basin of attraction, it is assumed that the local search is idealised, that is, the search navigates

Algorithm 2.1 Local Search (LS)

```
Input:
    Initial solution, $\mathbf{x} \in \mathcal{S}$
repeat
    $\mathbf{x}' \leftarrow \mathbf{x}' \in N(\mathbf{x})$ such that $\forall\ \mathbf{x}'' \in N(\mathbf{x}), f(\mathbf{x}') \leq f(\mathbf{x}'')$
    if $f(\mathbf{x}') < f(\mathbf{x})$ then
        $\mathbf{x} \leftarrow \mathbf{x}'$
    end if
until $f(\mathbf{x}') = f(\mathbf{x})$
return $\mathbf{x}$
```

through saddle points and terminates at a local optimum. The saddle point separating two basins of attraction is known as a *barrier*.

A *plateau*, or *neutral region*, is an area of the search space where the objective function is of constant value (i.e. flat/neutral). Specifically, a plateau is a set of solutions having equal f-values and where each solution is "connected" by a neutral neighbour (i.e. there is a path from any neutral solution to any other neutral solution in the plateau).

2.2 Fitness Landscapes

The notion of a *fitness landscape* (also known as an *adaptive landscape*) originates from theoretical biology as an abstract representation of the interaction between the genetic "encoding" of an organism, known as its genotype, and the evolutionary quality of the organism, known as *fitness* [205]. Traditionally, genotypes with a better genetic quality correspond to large fitness values, and the process of evolution can be abstractly conceptualised as "moving" throughout the fitness landscape, in "search" of the genotype with the largest fitness. The transition between genotypes corresponds to the evolution of the organism via reproduction and/or mutation. Fitness landscapes have also been used to represent the observable traits of an organism (known as phenotypes) with respect to their fitness [129].

The landscape metaphor is also a popular model in physics and chemistry for the analysis of materials' *potential energy surfaces*, also known as *energy landscapes* [171]. Specifically, energy landscapes describe the relationship between a molecule's geometry and the corresponding energy for each geometrical configuration. In contrast to fitness landscapes, configurations with *low* energy are typically desirable.

Landscapes provide a useful metaphor to model the complex relationship between object configurations and a corresponding measure of configuration quality. The fitness landscape metaphor was introduced into evolutionary computation as a model of the objective function values over the search space for a given optimization problem [169, 199]. Specifically, fitness landscapes provide a framework to analyse the interaction between solutions, their variables, and their corresponding fitness (objective) function values.

The fitness landscape metaphor in optimization has evolved significantly since its first applications in evolutionary computation. One major contribution in this area was the work of Jones [82], who provided a rigorous - yet highly accessible - definition of optimization problem landscapes. Stadler [171] then proposed a more generalised landscape definition, thereby facilitating the inclusion of concepts and results from statistical physics. Indeed, Jones' and Stadler's formulations of fitness landscapes persist favourably in the landscape analysis literature [19, 74, 120, 190].

Given the evolution of the landscape notion throughout recent years, definitions in the literature vary and are often considerably vague. For example, common definitions of the landscape, $\mathcal{L}$, include:

- $\mathcal{L} = (\mathcal{S}, f, d)$ [114, 120, 122, 130, 138]
- $\mathcal{L} = (\mathcal{S}, N, f)$ [28, 74]
- $\mathcal{L} = (\mathcal{S}, \chi, f)$, where χ is qualitatively defined as a notion of "connectedness", "neighbourhood", "nearness", "distance" or "accessibility" [19, 131, 171]
- $\mathcal{L} = (V, E)$, where V is a vertex set of the solutions' objective function values, and E represents the "connections" between solutions (defined using a notion of neighbourhood) [190]

In addition to the definitions of $\mathcal{L}$ above, $\mathcal{L}$ is frequently described as as vague interaction between $\mathcal{S}$ and f. For example, Picek and Jakobovic [128] define $\mathcal{L}$ as "a set of two functions f and d that define the fitness value and the distance between encoded solutions in the landscape". Equally confusingly, Talbi [181] describe $\mathcal{L}$ as "the tuple (G, f), where the graph G represents the search space and f represents the objective function that guides the search". Worse still, many papers in the landscape analysis literature omit a definition of the landscape [29, 57, 109, 161].

In order to consolidate the literature and provide a rigorous definition, the notion of a problem landscape is defined in Definition 2.5. The term "fitness" typically refers to maximisation, however since minimization of f is assumed throughout this thesis (meaning solutions with lower f values on the landscape are deemed "fitter"), it is rather improper to use the term "fitness landscape" to describe an optimization problem's landscape. Hence

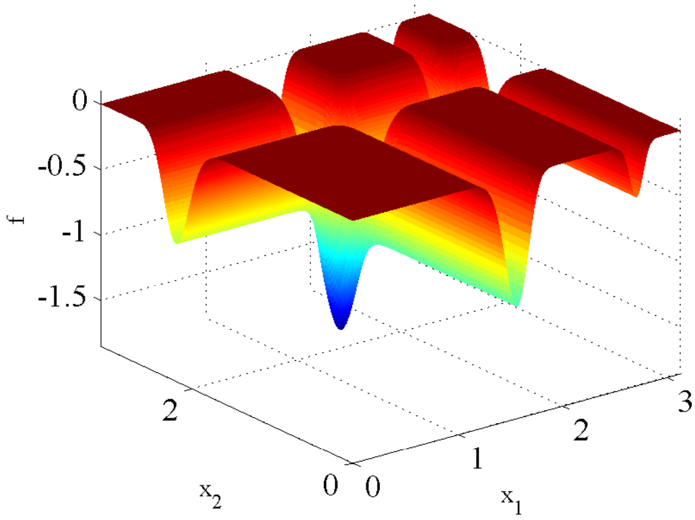


Figure 2.1: Problem landscape of the Michalewicz function defined over $\mathcal{S} = [0, \pi]^2$.

to avoid confusion between fitness landscapes from evolutionary biology and fitness landscapes in the (minimization) optimization context, this dissertation denotes the latter as *problem landscapes*.

Definition 2.5 (Problem landscape.)**.** *Given an optimization problem, let $\mathcal{S}$, f and d denote the search space, objective function and suitable distance function respectively. The* ***problem landscape*** *is the tuple:*

$$\mathcal{L} = (\mathcal{S}, f, d) \tag{2.2.1}$$

Definition 2.5 is consistent with landscape definitions in [114, 120, 122, 130, 138]. As defined in Equation 2.1.2, the distance d is used to provide a notion of neighbourhood. The candidate solutions and neighbourhood relation together form a connected graph, where vertices represent the solutions, and edges correspond to the distance between *immediate* neighbours (i.e. N where ϵ is arbitrarily small). The landscape is then formed by mapping the objective function values to each vertex (i.e. candidate solution) in the graph. Continuous problem landscapes are very intuitive, and are essentially a continuous space defined by $\mathcal{S}$, with the addition of an extra dimension for the "surface" defined by the objective function values. Figure 2.1 illustrates the problem landscape of the two-dimensional Michalewicz function (defined in Table A.1 of Appendix A), which shows multiple local minima and a single global optimum.

2.2.1 Common Landscape Terminology

The problem landscape framework provides a convenient abstraction for analysing and characterising optimization problem features and properties. Notions naturally used to describe two and three dimensional landscapes are often applied to problems of arbitrary dimensionality. For example, terms like "peaks", "valleys", "ridges", "funnels" and "plateaus" are used extensively in the problem landscape vernacular, with often little or no formalisation of their meaning (e.g. [30, 122, 128]).

Arguably one of the most widely used landscape descriptors is *modality*, which refers to the number of local optima (resembling modes) in the landscape. Highly multimodal landscapes with large transitions in fitness are generally described as *rugged*, while problems with few modes and small transitions in fitness are known as *smooth*[3] [74]. Ruggedness is a qualitative (and hence subjective) notion, and so numerous definitions have been proposed in the literature to quantify ruggedness. For example, Weinberger [199] measure ruggedness via the autocorrelation and correlation length of objective function values conducted over a random walk, Palmer [126] suggests defining landscapes as rugged if the number of local optima scales exponentially with a measure of problem size, and Vassilev et al. [190] quantify ruggedness via an entropic measure of the variety of fitness fluctuations in a random walk. Each of these measures are bounded, however the point at which a smooth problem becomes a rugged problem remains unclear.

Funnels have been used extensively in the potential energy surface literature, although like ruggedness, explicit definitions are vague and vary [103]. For example, Doye [45] defines a funnel as "a region of configuration space that can be described in terms of a set of downhill pathways that converge on a single low-energy structure or a set of closely-related low-energy structures". The exact algorithm to compute the downhill pathway is not provided, nor is the degree to which low-energy structures are considered "close" formally defined. Abstractly, a funnel is a region of $\mathcal{S}$ where the objective function generally decreases monotonically towards a single local optimum [103], and this definition is adequate for the purpose of the work in this thesis.

Ridges and *valleys* are commonly used to describe plateaus consisting of local maxima and minima respectively. A *massif central/big valley* landscape structure refers to a clustering of the distribution of local optima in the landscape, such that the local optima closer to the global optima generally have better objective function values [181]. Landscapes where the

[3] Confusingly, "smoothness" in this context is unrelated to the definition of smoothness in Section 2.1

global basin of attraction is small in comparison to the local basins of attractions are known as *deceptive*, as search heuristics with an element of greediness will often be "lured" into the local basins, and are hence "deceived" by the landscape structure.

The topological concepts and notions of ruggedness, ridges, valleys, plateaus and funnels provide an imagery with which to better visualise and understand the complexity of optimization problem landscapes. However, many of the concepts lack rigorous definitions and explicit quantitative measures. In addition, it is not entirely clear how the topological notions scale with dimensionality, and in particular, whether the landscape descriptors are useful or even exist in high dimensions [129]. Indeed, a recent survey on the prevalence of local optima and saddle points in high-dimensional, non-convex continuous error functions (from statistical physics, random matrix theory and neural network theory) concluded that there is a proliferation of saddle points, *not* local minima as is often thought (local minima with high error are exponentially rare) [41]. Furthermore, saddle points and local optima in high dimensional problems are often surrounded by plateaus and regions of negative curvature. From a more philosophical perspective, Provine [132] argues that if the sole purpose of a landscape metaphor is to provide intuition and aid understanding, then the use of high dimensional landscapes, for which there is little intuition, is of little help. Indeed, it seems very unlikely that the landscape descriptions derived from rudimentary geometry are appropriate for high D, where the possible number of variable permutations increases exponentially with D, resulting in a large increase in the types and complexities of landscape structures. There is clearly a need for landscape descriptors that are based on the landscape data, rather than intuition. Many data-driven landscape measures, properties and features have been proposed in the literature, and a review of these is conducted in Chapter 3.

2.3 Summary

This chapter defined the global optimization problem, as well as related concepts such as the notion of a neighbourhood, local and global optima, plateaus and basins of attractions. The fitness landscape framework originating from evolutionary biology was reviewed as a model for understanding the interaction between solutions, their variables and their respective objective function values. In addition to defining a consolidated notion of problem landscapes, explanations of commonly used terms in the landscape vernacular were reviewed. While many of the terms were developed from two and three dimensional geometric intuitions, they are frequently used to describe landscapes of arbitrary dimensionality. Issues regarding the suitability of the terms in high dimensional spaces were discussed, and an argument was made for the development of problem features and properties that are not derived from two and three dimensional landscape intuition.

CHAPTER 3

Landscape Analysis

Know how to solve every problem that has been solved.

Richard Feynman

The notion of an optimization problem landscape, $\mathcal{L} = (\mathcal{S}, f, d)$, given in Definition 2.5 (Section 2.2), provides a framework from which features summarising a landscape's structural information can be derived. Features defined using the landscape notion are dependent on f, $\mathcal{S}$ and d, meaning multiple landscape definitions, and hence features, are possible for a given optimization problem. The following chapter reviews the ability of problem landscape features and analysis techniques proposed in the literature to analyse, describe and characterise optimization problems. Where possible, features are defined and assessed under the generalised landscape notion, $\mathcal{L}$.

The analysis of problems is often closely associated with algorithm behaviour and performance, and so a discussion into the relevance and consequences of the No Free Lunch theorem, NP-completeness and problem difficulty is given in Section 3.1. The abilities and limitations of existing problem landscape features to characterise combinatorial and continuous problems are assessed in Section 3.2. Section 3.3 outlines and compares the challenges that are unique to characterising combinatorial problems, versus those that are unique to continuous problems. Section 3.4 reviews the efficacy of landscape features to quantify the similarity between problems. Related landscape analysis from the geography, ecology, biology, chemistry and physics literature is discussed in Section 3.5. Section 3.6 concludes the chapter with a summary of the limitations and shortcomings of existing landscape analysis techniques to characterise and compare problems.

3.1 No Free Lunch, NP-Completeness and Problem Difficulty

The No Free Lunch (NFL) theorem states that *all* algorithms perform equally well on average, across *all* possible problem instances [106]. The NFL theorem essentially means that for each problem that an algorithm instance performs well on, there is a problem for which the algorithm instance performs poorly on. While this may seem like a formidable obstacle in the pursuit of developing algorithms to efficiently solve optimization problems, the NFL theorem means that there are sets of problems that *particular* algorithms are better-suited to solving than other algorithms. Hence, understanding the relationship between algorithms and problems is of significant importance.

The relevance of the NFL theorem in practice is dubious; only discrete functions for which algorithms do not re-sample (i.e. re-visit) solutions are considered in the theorem, and so it is often argued that the problems and algorithms considered in practice do not represent the problems and algorithms to which NFL applies [201]. English [51] showed that the vast majority of *all* problems are highly incompressible functions, while problems of practical interest are generally compressible. Consequently, it is often argued that the problems encountered in practice represent a small proportion of all problems [51, 201]. Evidently, empirical benchmarking results, such as algorithm performances in continuous benchmarking competitions [72, 179], show that the performance of algorithms varies considerably across problems. A major goal in optimization research is therefore to determine and understand the problems that particular algorithm instances are well-suited to (and conversely, not well-suited to).

The NFL theorem does not hold for all problems encountered in practice, including NP-complete (decision) problems [201] like the Travelling Salesman Problem (TSP) and Number Partitioning Problem (NPP) (as described as optimization problems in Appendix A). NP-complete problems are generally perceived as difficult because there is no known polynomial-time algorithm to optimally solve them. However, many solvers can optimally solve large NP-complete problem instances in practice (e.g. Concorde is able to solve many large, real-world TSPs [42]). This is because NP-completeness refers to the worst-case complexity of an entire *class* of problems, and so the complexities of individual *instances* within the class can vary. Indeed, certain instances of NP-complete problems are *known* to be "easy" or "hard" for exact solvers [34]. Specifically, by identifying and controlling key problem features in the Travelling Salesman, Number Partitioning, Graph Colouring and Satisfiability problems, phase transitions are exhibited in the computational resources used by exact solvers [1, 22, 207]. The identification of such control features has traditionally required a deep level of problem understanding, and is therefore highly problem-specific.

Given that the performance of algorithms varies between problem instances, there has been increasing interest in developing features/properties to characterise problems, and to relate these features to the behaviour and performance of algorithms [77, 119, 162, 164]. While such features are frequently related to algorithm performance, it is important to emphasize that the features themselves do not imply problem difficulty. In other words, landscape features aim to measure/characterise pertinent structures, and it is the presence and interactions of such structures that affects algorithm performance. For example, consider Fitness Distance Correlation (FDC), which measures the extent of correlation between solutions' f values and their distance to a given reference solution (usually the closest global optimum) [82]. FDC is often reported (and indeed criticised [5]) as a measure of problem difficulty, however Jones [82, pp 178] originally argued that FDC is a *feature* of the landscape, and that a "difficult" feature for one algorithm to navigate may be "easy" for another algorithm to navigate.

A major aim of problem landscape analysis, and indeed this thesis, is to develop features that are able to describe, characterise and distinguish problem structures. Consequently, this thesis focuses on the development of *algorithm-independent* (but landscape-dependent) features and analysis techniques. Therefore, the following review of the landscape analysis literature evaluates the ability of proposed features to adequately characterise landscapes, independent of any particular algorithm (and hence without regard to notions of "difficulty").

3.2 Landscape Features

In comparison to continuous optimization problems, a considerable amount of problem analysis has been performed on combinatorial optimization problems. For example, the analysis of the TSP has resulted in a large number of features that are typically based on domain-specific knowledge and have been shown to contribute to problem difficulty [60, 173, 187, 206, 207]. In particular, Cheeseman et al. [34] and Ridge and Kudenko [141] show that increasing the standard deviation of the distances between cities for randomly generated TSP instances increases problem difficulty for a number of algorithms. Features such as these are very problem-specific and in most cases cannot be easily transferred to other problem classes. Instead, problem-specific features have been developed separately for other combinatorial problems classes such as Graph Colouring, Boolean Satisfiability, Time-tabling and Knapsack problems. More comprehensive reviews of combinatorial landscape and problem-specific features and techniques can be found in [130, 139, 163, 181].

Problem-specific features are very insightful and useful, however they do not allow comparisons between black-box problems or problems between different classes. No domain knowledge is available in the black-box scenario, and so for these problems, analysis is restricted to the candidate solutions, $\mathbf{x}$, from the feasible search space, $\mathcal{S}$, and their respective objective function values. However, complete enumeration of the search space is often impractical due to the finite, but very large number of candidate solutions. Hence, problem landscape analysis techniques typically employ random, statistical or other sampling methods to examine a set of solutions of interest (and/or their objective function values) from a landscape. Features based on the landscape metaphor are inherently very general and can also be applied to problems of different classes and/or non-black-box problems.

This section reviews the landscape analysis literature, with a particular emphasis on features based on finite samples of $\mathcal{S}$ and f. The review is certainly not exhaustive, but rather highlights fundamental issues and concerns with existing problem landscape analysis features. Readers interested in more comprehensive reviews on problem landscape analysis can consult the reviews in [130, 163, 171, 181].

3.2.1 Topological, Minimum Embedding and Fractal Dimensions

Arguably the most simple feature of an optimization problem is its *topological dimension*, D (also known as the *number of degrees of freedom*), defined as the number of variables in a candidate solution. Because D is solely related to $\mathcal{S}$, it does not capture any information regarding f, and is therefore a poor problem characteristic. Furthermore, D may not adequately reflect the *minimum embedding dimension*, defined as the smallest dimensionality for which an equivalent representation of the problem can be embedded within. Consider the function $f(\mathbf{x}) = x_1$, where $\mathbf{x} \in \mathbb{R}^D$; f is essentially a one dimensional function embedded in a D-dimensional space. Hence, its topological dimension is D, while its minimum embedding dimension is at most 1. While the minimum embedding dimension can be determined using dimensionality reduction techniques [93], it has not been used as a problem feature in the context of landscape analysis.

The *fractal dimension*, D_f, is a related notion of dimensionality that has been proposed to characterise the structural complexity in landscapes [167, 200]. There are multiple definitions in the literature (see [182] for a review), however all aim to capture the general notion that:

$$\text{bulk} \sim \text{size}^{D_f} \tag{3.2.1}$$

where *bulk* represents a measure of the landscapes volume/mass/information and *size* represents a linear distance (e.g. the diameter of the landscape). D_f quantifies how the structural detail of the problem changes with scale, and resulting values can be non-integer. Landscapes with smooth, gradual transitions in f over $\mathcal{S}$ yield fractal dimensions close to their topological dimension, while the fractal dimension of rugged landscapes will be larger [182]. Many of the techniques used to estimate D_f suffer from implementation issues that make them impractical in high dimensions (e.g. exponential growth in the sampling required to produce an estimate) [65, 182]. Power law analysis has been used to indicate the *presence* of fractal structures in certain combinatorial optimization problems [167, 200], however this analysis essentially yields a yes/no label (for the presence/absence of fractal structure) and is unable to quantify the degree or amount of fractal structure.

3.2.2 Local Optima and Basins of Attraction Based Features

The *number*, *size* and *distribution* of local optima are commonly employed to quantify how "rugged" a landscape is [181]. Determining the exact number and location of local optima involves exhaustive enumeration of $\mathcal{S}$, and is therefore infeasible for most problems of practical interest. Instead, estimates are often made using the set of distinct solutions resulting from local search algorithms restarted at multiple locations in $\mathcal{S}$ [143].

Gamier and Kallel [56] show that obtaining an accurate estimate of local optima using multiple local searches is computationally expensive; assuming n uniformly distributed basins of attraction, $O(n \log n)$ local searches are needed if the basins are equally sized, and $O(n^2)$ local searches are required if they are uniform randomly sized. Basins may still be missed in practice, and so the resulting estimate is a lower bound on the actual number of local optima. Given that n is unknown a priori in the black-box scenario, the number of local searches and their respective initialisation locations are often made as large as practically feasible [113, 136].

Despite the intense computational effort required to identify local optima, the number and distribution of local optima throughout $\mathcal{S}$ are frequently used to analyse both combinatorial and continuous optimization. For example, Boese et al. [16] and Reeves [136] show that certain instances of TSP problems contain a "big valley" structure that can be exploited by heuristics, while Alyahya and Rowe [6] recently observed correlation in the number of global optima in Number Partitioning Problem instances with a phase transition in the performance of an exact solver.

In continuous optimization, the shape, volume and distribution of basins of attractions have predominately been used to characterise artificial benchmark problems, such as the Black-Box Optimization Benchmarking (BBOB) problem set [30, 120, 113]. Basins of attraction are estimated similarly to local optima; multiple local searches are conducted, and the solutions visited during each local search are assigned to the resulting local optimum's basin. In an effort to reduce the intense computational effort required to determine the basins of attraction for a given problem, Muñoz et al. [120] utilise repeated local searches on $\mathcal{L} = (\mathcal{S}', f, d)$, where $\mathcal{S}'$ is a sample of 2000 solutions from $\mathcal{S}$. Because the local optima and basins of attraction are estimated from such a low-resolution landscape, the locations of local optima are likely to be erroneous or missed entirely, and the estimates of the basin sizes and their distributions throughout $\mathcal{S}$ are unlikely to be reliable. Muñoz et al. [120] consider only 2-D problems, and so the efficacy and practicality of their approach in higher dimensions is unclear.

3.2.3 Probabilistic Based Features

The *fitness distribution* (also known as the *density of states*) for a given problem landscape summarises the probabilities of each distinct objective function value in the landscape. Information regarding the candidate solutions and their relationship with their respective f values is not utilised. In practice, fitness distributions are constructed via density estimation of sampled f values [145]. The fitness distribution is non-unique; two problems with vastly different landscapes will produce identical fitness distributions if their sets of sampled f values are identical (regardless of where in $\mathcal{S}$ the f values occur). Because fitness distributions (and statistical measures of the distributions such as skewness) are based purely on f, they are applicable to both combinatorial and continuous optimization problems. Borenstein and Poli [18] utilise the fitness distributions of solutions generated from a Genetic Algorithm and random sampling to qualitatively analyse the one-max and needle-in-a-haystack problems. Similarly, Mersmann et al. [113] propose using the skewness, kurtosis and modality of the fitness distribution are as features for characterising problems. Results for the BBOB [72] problem set suggest that the skewness, kurtosis and modality are more accurate at classifying the BBOB problems than modality-related features such as the number of local optima.

Smith et al. [161] propose a set of four features (E_a, E_b, E_c and E_d), collectively called *fitness evolvability portraits*, that characterise both the ruggedness and neutrality of combinatorial and continuous landscapes. Fitness evolvability portraits aim to characterise the fitness of neighbourhoods from the perspective of solutions where $f(\mathbf{x}) = t$. Obtaining such solutions in practice is computationally expensive; solutions are sampled randomly and only accepted if their objective function value is equal to t. E_a measures the expected probability of a neighbouring solution to have "better" fitness than the current solution. Similarly, E_b estimates the expected fitness of neighbours, while E_c and E_d measures the expected fitness of the top and bottom p percentile of the neighbouring fitness respectively. Because E_a, E_b, E_c and E_d consider the neighbourhoods of solutions with fitness t, structural relationships defined over larger intervals than the neighbourhood are not considered. The expectation of all neighbourhoods' fitnesses across *all* solutions of fitness t also limits the capacity for the fitness evolvability portraits to capture localised structure.

E_a, E_b, E_c and E_d are dependent on the value of t chosen, and Smith et al. [161] suggest using a range of different t values to obtain a broad analysis of the problem. However the selection of an appropriate range of t values is difficult in the black-box scenario, where the existence of objective function values equal to t is not guaranteed. The size and distribution of the sample as well as the neighbourhood function are also important considerations for the application of fitness evolvability portraits. For continuous problems, Smith et al. [161] suggest a uniform random neighbourhood function (of unspecified range) and either uniform random sampling or algorithm trajectories to obtain a representative sample. Curiously, the authors ignore their own advice and analyse highly simplistic 1-D continuous problems by discretising the search space into a grid of equally spaced solutions, where directly adjacent grids are considered neighbours. Consequently, the efficacy of the fitness evolvability portraits for analysing high dimensional optimization problems is unknown.

3.2.4 Correlation Based Features

Correlation-based statistics of random walks in $\mathcal{S}$ are also commonly used to quantify ruggedness. The *random walk correlation function*, $\rho(s)$, measures the linear correlation of the objective function values of a sequence of solutions separated by a specified distance,

s [199]. That is, given a sequence of solutions $\mathcal{S}' = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$, their respective objective function values, $\mathcal{F} = \{f(\mathbf{x}^1), \ldots, f(\mathbf{x}^n)\}$:

$$\rho(s) = \frac{1}{(n-1)\,\sigma_F^2} \sum_{i=1}^{n-1} \left(f\left(\mathbf{x}^i\right) - \mu_F\right)\left(f\left(\mathbf{x}^{i+1}\right) - \mu_F\right) \tag{3.2.2}$$

where μ_F and σ_F^2 are the mean and variance of F and $s = d\left(\mathbf{x}^i, \mathbf{x}^{i-1}\right)$ for $i > 1$.

The *correlation length* reflects the largest distance between two solutions such that their objective function values are statistically significant [199]. The most common definition of correlation length is:

$$\xi = \frac{-1}{1 - \rho(1)} \tag{3.2.3}$$

The use of $\rho(1)$ in Equation 3.2.3 originates from time series analysis, where observations are sampled at precise intervals in time (i.e. observation $\mathbf{x}^i$ is 1 time interval from $\mathbf{x}^{i+1}$ and 2 time intervals from $\mathbf{x}^{i+2}$). A random walk with a fixed step size in continuous space will yield solutions at a variety of distances (and hence intervals) apart, and so a more appropriate definition of the correlation length is [170, 171]:

$$l = \sum_{s=0}^{\infty} r(s) \tag{3.2.4}$$

Alternatively, Reeves and Rowe [138] define the correlation length as the largest s before $\rho(s) \leq 0$. There is no consensus in the literature as to which definition is preferable, and so the variation in definitions hinders consistent and accurate comparisons of correlation lengths. By definition, correlation length summarises the linear correlation between objective function values, and so landscapes containing non-linear structure are inadequately characterised. Furthermore, potentially valuable information regarding the relationship between $\mathcal{S}$ and f is ignored by these features. Despite known issues, the autocorrelation and correlation length continues to be widely used to characterise problems [74, 124].

Fitness Distance Correlation (FDC) is a popular landscape feature that measures the extent of correlation between objective function values and their distance to a given reference solution (usually the global optimum) [82, 83]. Specifically, given a sample of solutions, $\mathcal{S}'$, their respective objective function values, $\mathcal{F}$, and a set of distances, $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, from each solution in the sample to the reference solution, the FDC coefficient is defined as:

$$FDC = \frac{C_{\mathcal{F}\mathcal{D}}}{\sigma_{\mathcal{F}}\sigma_{\mathcal{D}}} \tag{3.2.5}$$

where

$$C_{\mathcal{F}\mathcal{D}} = \frac{1}{n}\sum_{i=1}^{n}\left(f\left(\mathbf{x}^{i}\right) - \mu_{\mathcal{F}}\right)\left(d_i - \mu_{\mathcal{D}}\right) \tag{3.2.6}$$

and the mean and standard deviation of $\mathcal{D}$ are denoted by $\mu_{\mathcal{D}}$ and $\sigma_{\mathcal{D}}$, while the mean and standard deviation of $\mathcal{F}$ are denoted by $\mu_{\mathcal{F}}$ and $\sigma_{\mathcal{F}}$.

FDC was originally defined for the analysis of combinatorial problems and has been applied to a wide variety of problems including Travelling Salesman Problems [46], Capacitated Vehicle Routing Problems [183] and exam time-tabling problems [124]. In the continuous optimization context, Gallagher [54] calculated FDC for the error surface (i.e. problem landscape) of the training problem for a multi-layer perceptron neural network [54, 55]. For the specific learning task considered (student-teacher model), the global optimum was known, however this would not normally be the case for a neural network training problem. Solutions for the calculations were sampled from within a specified range around the global optimum. Wang and Li calculate FDC in the context of evaluating a continuous NK-landscape model and on some standard test problems [198]. The Congress on Evolutionary Computation (CEC) 2005 benchmark test suite and BBOB problem sets have both been analysed using FDC [57, 122, 189]; the CEC 2005 and BBOB problems have positive FDC coefficients, and Müller and Sbalzarini [122] conclude that FDC alone is not a sufficient feature for characterising the CEC 2005 problems.

While FDC is arguably one of the most commonly used problem landscape features, it has notable limitations (see Tomassini et al. [184, pp 217-219] for a review). For example, certain problem properties, such as non-linear scaling of the objective function, are known to affect FDC's reliability [123]. In addition, the traditional use of FDC uses each solutions' nearest global optimum as a reference point, and such knowledge is not typically known in many practical or black-box situations. Instead, a single global optimum is typically approximated using the sample's best solution [57, 122, 124], however the effect of this substitution on the adequacy of the resulting FDC estimate is not investigated or discussed in the literature (and hence Section 4.5 investigates this issue in further detail).

3.2.5 Information Based Features

Vassilev et al. [190] introduced three methods - *information content*, *partial information content* and *information stability* - for characterising combinatorial optimization problems via the sequence of objective function values resulting from random walks. Information content aims to quantify the ruggedness of the landscape through an entropic measure of objective function value fluctuations. Adjacent f values in the sequence are classified based on whether their difference in f (within a level of precision, ϵ) is increasing, decreasing or constant. Hence, information regarding the amount of change or the positioning of solutions is ignored. Specifically, given a sample of candidate solutions, $\mathcal{S}' = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$, let $S(\epsilon) = (s_1, \ldots s_{n-1})$ denote the sequence of symbols $\mathrm{s_i} \in \{-1, 0, 1\}$ where, for a given $\epsilon \geq 0$:

$$s_i(\epsilon) = \begin{cases} -1, & (f(\mathbf{x}^i) - f(\mathbf{x}^{i+1})) < -\epsilon \\ 1, & (f(\mathbf{x}^i) - f(\mathbf{x}^{i+1})) > \epsilon \\ 0 & \text{otherwise} \end{cases}$$

The information content is then:

$$IC(\epsilon) = -\sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]} \tag{3.2.7}$$

where $p, q \in \{-1, 0, 1\}$ and $P_{[pq]}$ is the probability of the substring pq in S^1. Resulting information content values are in $[0, 1]$, and highly rugged landscapes (where $pq = (-1, 1)$ or $pq = (1, -1)$) yield values close to $log_6 2 \approx 0.3869$.

Partial information content attempts to characterise the degree of modality in the landscape. The fluctuations in f values of neighbouring solutions in the sequence are counted and normalised by the length of the sequence. Formally, let $S' = (s_1, \ldots s_m)$ denote the substring of S such that $s_j \neq 0$ and $s_j \neq s_{j-1}$ $\forall$ $j > 1$. The partial information content is defined as

$$\begin{aligned} PIC(\epsilon) &= \frac{|S|}{|S'|} \\ &= \frac{n}{m} \end{aligned} \tag{3.2.8}$$

The substring S' essentially consists of an alternating series of -1's and 1's, representing fluctuations between decreasing f and increasing f. The authors suggest that because S' records the number of fluctuations in the sequence of f-values, the number of local optima in the landscape can be estimated from the partial information content. This is however

[1] The logarithm is base 6 because there are $\binom{3}{2} = 6$ combinations of pq ($p \neq q$) using the alphabet $\{-1, 0, 1\}$.

erroneous; objective function fluctuations can occur for saddle points (which are not local optima), while multiple basin crossings may count the same optima multiple times, thus inflating the estimate. Thus, partial information is a poor indicator for the number of local optima.

The last of the methods proposed by Vassilev et al. [190] is *information stability*, described as the largest difference in f between neighbouring solutions in the walk:

$$IS(\epsilon) = \max \left| f(\mathbf{x}^i) - f(\mathbf{x}^{i-1}) \right| \quad \forall\, i > 1 \tag{3.2.9}$$

The entropic measures of objective function values proposed by Vassilev et al. [190] analyse and operate solely on a simplistic representation of the objective function values of candidate solutions, sampled during a random walk. Notably, the measures do not utilise the information afforded by the candidate solutions, such as their relationship with f, or to each other.

Malan and Engelbrecht [109] estimate information content on seven continuous benchmark problems in 1 and 30 dimensions. Instead of using a neighbourhood function to generate solutions along the random walk, random walks with an increasing random step size are used to generate samples. Thus, the initial samples capture structure at small scales, while the samples at the end of the walk capture structure between large steps. This type of walk is particularly biased; the initial area is examined quite thoroughly, while the remaining areas of the landscape are neglected. The authors state that the choice in walk is an attempt to resemble the search path resulting from an individual in a population based algorithm. However, as discussed by Smith et al. [161], algorithm trajectories can easily be used instead of random walks. Indeed, Muñoz et al. [120] used samples generated by instances of a (1+1) Covariance Matrix Adaptation Evolution Strategy (CMA-ES), Particle Swarm Optimization (PSO) and random search to estimate variants of information content and partial information on 2-D continuous problems from the BBOB problem set. By using a technique to reduce bias in the samples, the CMA-ES and PSO samples yielded information content measures within 5% relative error of the measures derived from random sampling.

The information measures proposed by Vassilev et al. [190] have inspired additional, complementary features. Borenstein and Poli [21] adopt an approach similar to information content in order to characterise a variety of combinatorial problems and predict the performance of a Genetic Algorithm. Steer et al. [175] proposed four "secondary" features based on the information content and partial information content. The secondary features have been estimated on three 10-D combinatorial benchmark functions [175] as well 2-D continuous problems from the BBOB problem set [120]. The "secondary" features have been applied to only a small set of problems, and so their ability to characterise optimization problems remains largely unknown.

A very different approach to capturing the information in a landscape is taken by Borenstein and Poli [19], who argue that to adequately characterise landscapes, both the quality and quantity of information available must be considered. The authors propose the *information landscape*, which is a matrix, M, used to compare solutions sampled from $\mathcal{S}$. Entries $m_{i,j}$ are assigned probability values indicating the likelihood that $f(\mathbf{x}_i) > f(\mathbf{x}_j)$. If two information landscapes are constructed using the same sample of solutions, then the average difference between entries can be used as a comparative measure between problems. From this, one can compute the average difference between any given landscape to a random landscape (where all entries in M are 0.5). This value is normalised and indicates the *degree of information* (independent of the sample used), and so it can be used to characterise problems. However, as the authors remark, some potentially useful information, such as the actual objective function values, is not captured by information landscapes.

3.2.6 Other Features

Dispersion

Originally proposed as a continuous problem metric, *dispersion* [103] measures the average distance between pairs of high quality solutions, therefore indicating the degree to which high quality solutions are concentrated/clustered. Quality is determined by sampling n solutions and using truncation selection to retain the fittest tn solutions, where $t \in (0, 1]$. Let $\mathcal{S}' = (\mathbf{x}^1, \ldots, \mathbf{x}^m)$ denote a sample of $m = \lfloor tn \rfloor$ candidate solutions ordered by their objective function values, i.e. $f(\mathbf{x}^i) \leq f(\mathbf{x}^j)$ where $1 \leq i \leq j \leq m$. The dispersion of $\mathcal{S}'$ is:

$$\text{dispersion}(\mathcal{S}') = \frac{1}{m-1} \sum_{i=1}^{m-1} \left(\frac{1}{m-i} \sum_{j=i+1}^{m} d(\mathbf{x}^i, \mathbf{x}^j) \right) \tag{3.2.10}$$

where d is an appropriate distance function.

As dispersion is purely based on the notion of distance between points, it is applicable to both combinatorial and continuous problems. Dispersion has been used to study the performance of algorithms relative to particular problems (and their structure); CMA-ES, hybrid PSO/CMA-ES algorithms, Pattern Search methods and Local Search have been analysed on a number of benchmark functions [103, 121, 202]. The dispersion values of 2, 5, 10 and 20 dimensional problems from the BBOB problem set have also been used in the feature-set of an algorithm prediction model [118, 119]. Garden and Engelbrecht [57] also calculated dispersion values for artificial and benchmark problems (including the CEC 2005 and BBOB problem sets), however their experiments resulted in *negative* dispersion values. Dispersion is by definition **strictly** positive (since it is the expectation of distance values), and so the results in [57] are questionable.

Dispersion makes only limited use of the objective function values of solutions via the value of t used to truncate the sample. Furthermore, the existing methodology currently used to estimate dispersion, namely the use of Euclidean distance in conjunction with a uniform random sample, can lead to convergent dispersion values (Section 4.4 investigates this issue in further detail and proposes a number of improvements to the methodology).

Anisotropy

A landscape is isotropic if sub-regions of the landscape are statistically indistinguishable from the entire landscape [172]. The *coefficient of anisotropy* measures the amount of anisotropy by comparing a summary statistic of the landscape to a sub-region of interest [172]. Landscapes may be isotropic in regard to one sub-region, while anisotropic in another, and so repeated estimates of anisotropy on different sub-regions are analysed to obtain an overall estimate of landscape anisotropy.

Pitzer and Affenzeller [131] suggest quantifying anisotropy through estimating landscape features, such as autocorrelation or information content, on sub-regions and comparing the variation in the resulting features. Assuming the chosen landscape features are adequate characteristics of landscape structure, the features will yield consistent values for isotropic landscapes, whereas anisotropic landscapes may yield varying values. The degree of anisotropy is heavily dependent on the summary statistics utilised, and Pitzer and Affenzeller [131] caution that anisotropic landscapes can be falsely deemed as isotropic if the features are inadequate.

Epistasis

The degree of epistasis in a problem indicates the amount of non-linearity and interdependency between the variables [138]. The complex interactions between variables in highly epistatic problems result in rugged, heterogeneous landscapes. Davidor [43] proposed *epis-*

tasis variance as a measure of epistasis based on statistical measures of discrepancy between the objective function and a linear model. It has been shown that while epistasis variance is aimed at measuring epistasis, it actually measures the *absence* of epistasis [123]. Epistasis variance has numerous well documented flaws and limitations [130]. One such flaw is that the epistasis variance for constant functions and first-order functions is 0, despite the structural difference between constant and first-order functions. Another important limitation is that epistasis variance cannot differentiate between different orders of non-linearity between variables. To overcome this, Rochet [144] proposed *graded epistasis* and *graded epistasis correlation*, however these too suffer from many of the issues attributed to epistasis variance. [151] proposed four measures of epistasis (significance, entropic epistasis, mean significance and mean entropic epistasis) using an information-theoretic approach. The four measures have seen little use in the problem analysis literature; a small set of combinatorial problems were originally analysed and recently the measures have been used in the feature-set for an automated algorithm prediction technique [119].

Fitness Clouds

A *fitness cloud* is a two dimensional visualisation of the fitness of a sample of solutions versus the fitness of their neighbours [38]. The resulting visualisation indicates the potential for increases in objective function values of neighbouring solutions in the landscape. The minimum, maximum, mean and standard deviation of the f values of the neighbouring solutions summarise the fitness cloud information. Similarly, Vanneschi et al. [188] propose the *negative slope coefficient*, a metric based on the fitness cloud data that can be used to quantify the evolvability of the problem. For convenience, Collard et al. [38] analyse a single neighbour for each solution. As previously discussed, choice of the neighbourhood function, number of neighbours to generate and which neighbour to select are important, non-trivial considerations for the application of fitness clouds and negative slope coefficient analysis on continuous problems.

3.3 Combinatorial vs Continuous Feature Development

In a continuous search space, topological landscape features conceptually similar to the combinatorial case can be defined mathematically (as suggested in Pitzer and Affenzeller [130]), but evaluating these features on a real problem instance is problematic. Contrary to combinatorial problems, each solution in continuous space has an infinite number of neighbours in theory, and a finite but extremely large number in practice due to finite-precision representation of floating-point numbers. Hence, combinatorial problem features that are reliant on neighbourhood information - namely *autocorrelation* [199], *correlation length* [170], *fitness evolvability portraits* [161], *fitness clouds* [38] and variants of *information content* [175, 190] - must introduce additional assumptions and parameters to be used in a continuous space, such as the size and distribution of the neighbourhood, as well as methods for adequately sampling the neighbourhood. Common recommendations and approaches include sampling from a (bounded) uniform neighbourhood distribution, discretising the space [161], varying the neighbourhood size [109] and utilising algorithm trajectories [120]. However, the validity of these assumptions and their effects on empirical results are not well understood.

Another significant difference between combinatorial and continuous landscapes is tied to the *distance* between points in the solution space (using some appropriate metric). For a combinatorial landscape, the minimum possible pairwise distance will occur between a point and one of its neighbours. There will also be a finite set of possible distance values between all pairs of candidate solutions. For a continuous landscape, the minimum distance between points can be made arbitrarily small (in practice until the limit of precision is reached) and the number of possible distance values is infinite. Consider a combinatorial problem with binary representation, $\mathcal{S} = [0,1]^D$. To solve the problem is to determine whether each variable $x_i \in \mathbf{x}$ should take the value 0 or 1. A distance metric can be defined between points in the solution space (e.g Hamming distance), however there is no notion of the *scale* of x_i. For a continuous problem however, finding an appropriate scale for each x_i is critical (e.g. does the objective function vary in a significant way with changes in x_i of order 10^3? 10^{-3}? 10^{-30}?). Problem landscape techniques that originate from the assumption of discrete $\mathcal{S}$, such as autocorrelation, correlation length and information content, do not capture such information because it is not relevant for the combinatorial case.

3.4 Measuring Problem Similarity

There are many applications, including the development of algorithm portfolios [95] and automated algorithm predictors/selectors [15, 77, 119], that inherently rely on notions of problem similarity. For example, when faced with a new problem to solve, a logical first step in solving the problem is to determine whether a similar problem has been solved before.

The comparison between problems can be difficult due to the complexity of problem definitions, difference in notion, and the use of domain-specific terminology. Fortunately, the notion of the problem landscape provides a common framework for specifying and analysing problems. It follows that landscape features are often used by proxy to quantify problem similarity [162, 164]. Because individual features are generally developed to capture *specific* problem structures of interest, the structures that are essential in differentiating problems can be missed. Consequently, numerous examples exist where two structurally different problems yield the same features [5, 122, 123].

Given the limitations of individual landscape features, features have been combined into feature *sets* or *ensembles* in an attempt to gain greater discrimination between problems [77, 95, 162]. The underlying premise is that an ensemble of features captures more of the information in the problem than a single feature alone, and hence ensembles are more powerful discriminators. Quantifying problem similarity via feature ensembles requires a broad range of features, essentially capturing *all* information in the problem. Explicit quantification of the ability of feature ensembles to adequately differentiate problems remains a largely unexplored research area and is analogous to feature selection in machine learning [69].

Empirically, large feature-ensembles (e.g. ensembles with over 40 features) have been used to differentiate problems with moderate success. For example, Hutter et al. [77] used 43 features (based on basic statistics, local optima measures and graph-based metrics) to predict the running time of two algorithms on unseen problem instances. Smith-Miles and Tan [164] combined over 40 scalar features of Travelling Salesman Problems (TSP) into a *feature vector*, and the similarity between two TSPs was defined via a distance function between feature vectors. In the continuous setting, Mersmann et al. [113] used an ensemble of over 50 continuous features to predict the classes of problems in the BBOB problem set. In related work, Bischl et al. [15] used the same feature ensemble calculated on the BBOB problem instances to select well-performing algorithms from a portfolio. The existing work has shown encouraging results for small sets of problems, however the methodologies all rely on a large number of features, which can be computationally expensive to obtain for large sets of problems. Hence, the practical feasibility of applying feature-ensembles, as well as their resulting ability to characterise problems, remains relatively unknown for larger, more diverse problem sets.

Algorithm performance has also been used to quantify and compare problem similarity [117]. In this approach, an algorithm (or suite of algorithms) is applied to each problem, and the similarity between problems is derived via the similarity in algorithm performance. This approach relies heavily on the choice of algorithms, algorithm parameters and performance measures, and selecting these a priori is difficult in practice.

3.5 Landscape Analysis in Other Scientific Disciplines

The landscape metaphor and related concepts originate from the physical landscape encountered in everyday life, formed by the complex interactions between physical structures such as mountains, plateaus, valleys, ridges and cliffs. Landscapes are studied in many areas of science including geology, ecology, biology, chemistry and physics, and yet much of this literature remains unexplored in evolutionary computation (and more generally, optimization).

A plethora of landscape features and properties have been developed in the landscape ecology literature. Such landscape features are studied for their effect on ecological processes, including species' hunting/foraging patterns [147], breeding patterns [165] and migration patterns [39]. Landscape features are typically calculated from a spatial model (e.g. linear regression) of a survey/sample of "patches" (i.e. points) on a physical landscape of interest. Landscape features generally fall under two broad categories; geographical features (e.g. tortuosity, elevation and fractal dimension) [142] and ecologically-specific features (e.g. relative richness, dominance and connectivity) [185]. Geographical and ecological landscape features are highly specialised towards the analysis of 2-D and 3-D landscapes, and consequently, are generally not applicable to the analysis of high dimensional landscapes. While some formulations may be amenable to generalisation to higher dimensions (e.g. elevation), it is unlikely that they are able to characterise the structural complexities of high dimensional problem landscapes.

The notion of a landscape is frequently used in biology, chemistry and physics to model and analyse the relationship between atomic or molecular positions/configurations and their corresponding potential energy [197]. The potential energy effectively forms a surface over the space of feasible configurations, and hence is referred to as *potential energy surface* or *energy landscape*. Potential energy surfaces of practical interest are often high dimensional and contain many local optima [197]. Local minima on the surface correspond to states/configurations of low energy, which are of great importance in many applications including catalyst design and spin glass models. Indeed, Wales [197] state that the main focus of potential energy surface analysis is to determine "the influence and accessibility of a number of local optima". The focus of such landscape features is solely on local optima and basins of attraction, and are thus of limited use in their ability to adequately describe, characterise and discriminate optimization problem landscapes. Nevertheless, many landscape features used widely in optimization - such as barriers [9], autocorrelation and correlation length [199] - originate from the analysis of potential energy surfaces. A comprehensive review of potential energy surface features suitable in the optimization context can be found in [171].

3.6 Summary

Regardless of whether problem features originate in the combinatorial or continuous domain, there are some important issues that limit the ability of the features to characterise and differentiate black-box optimization problems. Namely;

- The reliance on problem-specific information that is not available for black-box problems.
- The inability of features to utilise all information available in the black-box setting.
- The filtering/compression of landscape information into a single scalar value.
- The adaptation of combinatorial features to continuous problems (and vice versa).

Arguably the most pertinent issue is the failure of existing techniques to utilise all information available in the black-box scenario. This stems from the design paradigm traditionally used in the development of new landscape analysis techniques. Designers typically aim to capture a particular problem structure of interest. For example, the fitness distribution describes the probability of particular objective function values, and no other structures or topological features within the landscape are intended to be captured. With a given structure/topological property of interest, designers then decide how best to capture the structure (e.g. what type of sample should be used as well as how the sample should be filtered and analysed). Herein lies a major issue; by only considering a certain aspect of landscape structure and by filtering the information available, information that describes (and potentially differentiates) a landscape may be lost. Furthermore, many techniques compress the information gleaned into a single scalar value (e.g. correlation length) representing the structural feature of interest (e.g. ruggedness). Such compression leads to even more information loss.

In summary, there is significant need for developing more powerful analysis techniques for black-box optimization problems. In Chapter 5, a new approach is proposed to study the characteristics of a landscape independent of any particular algorithm. Importantly, the approach utilises all black-box information available, makes no assumptions regarding the structure within landscapes, and does not target specific predefined landscape properties.

Chapter 4

Sampling in Continuous Spaces

It is a capital mistake to theorize before one has data.

Arthur Conan Doyle

The enumeration of an optimization problem's search space is often infeasible for a number of reasons. In the continuous context, the search space can never be enumerated as there are an infinite number of solutions. Assuming a certain level of precision, the search space is finite but may still be too large to enumerate (and this is also the case for many combinatorial problems). Consequently, the application of many landscape analysis techniques in practice is reliant on finite samples of solutions and/or their respective f-values.

This chapter reviews existing literature for sampling continuous problems for the purpose of landscape analysis. The sampling literature is vast, and so the review mainly focuses on techniques appropriate for continuous landscape analysis. Important limitations and considerations are discussed, and a sampling methodology particularly suitable for the analysis of high dimensional continuous problems is proposed. Case studies are also presented to illustrate the effects of different sampling methodologies (and their respective settings) on the theoretical and empirical behaviour of two widely used landscape analysis techniques; dispersion and fitness distance correlation. The chapter concludes with a summary of the contributions.

4.1 Methods and Techniques

Many landscape analysis techniques derive features/properties from finite samples of solutions and/or objective function values. The ability of these techniques to capture and describe important landscape properties is therefore heavily dependent on the sample. If crucial landscape structures are missing from the sample, the resulting features will inadequately characterise the problem. Therefore the choice in sampling methodology is extremely important, and may affect the accuracy of the landscape analysis.

Sampling and experimental design is a major component in applications from a variety of fields, including statistics [91], engineering [158] and economics [101], and as a consequence, the sampling literature is vast and highly application-driven. The measurement of interest typically influences the type of sample required, and the sample is then generated using a relevant technique. For example, sampling strategies satisfying certain uniformity properties can be employed to produce a sample with even coverage [101, 158].

In the context of continuous optimization problem analysis, the choice in sampling strategy is often motivated by the objectives and/or requirements of the problem analysis technique, simplicity of the strategy, as well as the bounds and constraints of the problems. Constraints are typically handled by generating solutions until a feasible solution is found [108]. When no information regarding the problem is known a priori, it is natural to sample the search space, $\mathcal{S}$, uniformly and (preferably) without prejudice. Thus in the following, a focus is given to sampling without prejudice from unknown spaces/distributions, which is particularly relevant in black-box optimization where landscape knowledge is limited. Techniques specifically aimed at sampling from a target/known distribution, including Markov Chain Monte Carlo sampling [91], are not explored in this thesis.

Arguably one of the simplest uniform sampling methods is *systematic* sampling (also known as grid search) [166], where solution variables are sampled systematically at a given distance apart. The resulting sample resembles a D-dimensional grid in $\mathcal{S}$. For unbounded problems, suitable bounds must be introduced. A variant of systematic sampling has been recently used to sample and analyse the 2-D BBOB problem set; search spaces are discretised into 10×10 "cells", with 1000 solutions randomly distributed over the cells [87]. While systematic sampling is useful in low dimensions, the sample size required increases exponentially with D. Furthermore, because solutions are sampled at precise increments, important structures within highly regular problems may be concealed *between* sample points. For example, a systematic sample of a periodic function will fail to capture interesting structures between sample points.

An approach similar to systematic sampling is to sample solutions *uniform randomly* from $\mathcal{S}$. Once again, bounds must be introduced for unbounded $\mathcal{S}$. Uniform random sampling is less restrictive than systematic sampling; solutions are randomly located and hence structure is sampled sporadically. Uniform random sampling is fast and simple to implement, and as a result it has been used widely in the landscape analysis literature [103, 122, 160, 161] and also more generally in numerical analysis and simulation [63, 91].

While both systematic and uniform random sampling produce uniform samples of $\mathcal{S}$, a large number of samples are required to achieve good coverage of $\mathcal{S}$, including sampling the dependence and independence between variables (dimensions). Low discrepancy sampling (also known as quasi-random sampling) are a class of sampling techniques that produce samples with similar properties to uniform random samples, but contain a minimal number of points [91]. This is achieved by targeting specific areas of $\mathcal{S}$, while obtaining (measurable) uniformity.

One widely used low discrepancy sampling technique is *Latin Hypercube* (LH) sampling. LH sampling generates n solutions by first dividing $\mathcal{S}$ into n^D equally sized hypercubes, and then placing each of the n solutions into a hypercube such that all other hypercubes in axis-alignment are empty [110]. *Orthogonal* sampling is an extension of LH sampling with a constraint that the solutions are distributed evenly throughout $\mathcal{S}$, meaning all sub-regions of $\mathcal{S}$ must have an equal density of solutions. Numerous other methods to produce low discrepancy samples exist, including *Sobol*, *Halton* and *Fuare* sampling (see [91] for further details). LH sampling has been widely used in experimental design, machine learning and optimization to determine parameter (and hyper-parameter) settings [12]. It is particularly useful in applications where the cost of evaluating f is high, as it ensures a wide variety of parameter value combinations, at a reasonable computation cost. In the context of landscape analysis, both Mersmann et al. [113] and Muñoz et al. [118] have recently utilised LH sampling to calculate various landscape features of BBOB problems spanning between 2 to 20 dimensions.

To illustrate the coverage of the aforementioned sampling techniques, Figure 4.1 displays 4 solutions sampled from $\mathcal{S} = [0, 1]^2$ using systematic, uniform random, LH and orthogonal sampling. In this example, uniform random and systematic sampling neglect particular sub-regions of $\mathcal{S}$, while LH and orthogonal sampling provide good coverage of $\mathcal{S}$, using the same number of solutions.

The above sampling methodologies provide an unbiased sample, but require rectangular bounds on $\mathcal{S}$. In contrast, random walks are a class of sampling techniques that can be used to sample both bounded and *unbounded* problems. They explore a space by generating an initial solution, $\mathbf{x}_0 \in \mathcal{S}$, and calculate subsequent solutions based on a step from the current solution. Random walks generally differ by the *direction* and *size* of the steps between solutions. Combinatorial problems can be sampled via random walks by selecting solutions

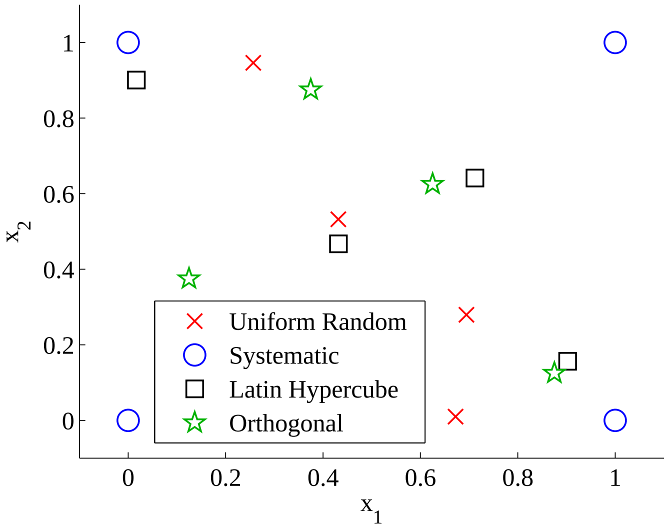


Figure 4.1: Four solutions sampled from $\mathcal{S} = [0, 1]^2$ using uniform random, systematic, Latin hypercube and orthogonal sampling.

from the current solution's neighbourhood [74, 190]. The notion of a neighbourhood can vary between combinatorial problems, however neighbourhoods are always finite. In contrast, neighbourhood relations in continuous problems are generally based on continuous intervals. Consequently, random walk steps are often taken in uniformly random directions, with the step size randomly sampled from a specified distribution (such as uniform or Gaussian [108]). Numerous variations of continuous random walks have been proposed and used in the context of continuous landscape analysis. In [109], a random walk with randomly sized, yet increasing, steps is utilised to estimate information content. Similarly, [107] use random walks with axis-aligned, fixed-size steps to calculate a variety of features including gradient estimates and dispersion. As previously mentioned, sampling at a predetermined interval can also miss important structures (occurring between steps), and so the fixed-step walks are limited. In an attempt to address this limitation, [108] proposed the use of multiple *smaller* random walks (anisotropic, uniform random sized steps), initialised at specific regions of $\mathcal{S}$ as the basis for landscape analysis techniques focused on ruggedness, smoothness and neutrality. Their proposed walk is biased and takes steps specifically away from the initialisation solution, as well as away from the search space boundary once it is reached. Such guidance in direction produces a highly biased and computationally expensive walk that may miss important structures near the initialisation and boundary areas.

Random walks can also be biased to selectively explore regions of interest. For example, an *adaptive* random walk samples solutions of increasing fitness (i.e. steps are only taken when they lead to a better solution) [85], while *neutral* random walks sample paths of consisting of neutral neighbouring solutions (i.e. steps between a solution and its furthest neutral neighbour are taken) [140]. Random walks initialised at local optima and simulating recombination and mutation have also been used to calculate FDC on NK landscapes in order to gain insight into the behaviour of a memetic algorithm [114]. The diversity of walks used in the problem analysis literature exemplifies the fact that there is no single preferred walk, and that choice in walk is largely guided by the underlying feature/measure of interest.

The solutions visited by algorithms throughout search have also been widely used to analyse the structural features of optimization problems [121, 161]. There are many advantages to this type of approach: algorithm data is available for many problems, and the problem can be both solved and analysed at once. In addition, the behaviour and performance of algorithms can be analysed in conjunction with the problem structures, potentially leading to novel insights [103]. However, algorithm trajectories are highly biased; structures unexplored by one algorithm (and hence largely absent in the resulting sample) may be imperative to the performance of another. In light of these issues, Muñoz et al. [120] developed methods to reduce the bias in estimates of information content and basin of attraction measures based on algorithm data. These methods were shown to be comparable with the same features calculated using an unbiased, uniform random sample for two dimensional problems. The computational effort involved to remove bias restricts the techniques to low dimensional (e.g. $D = 2$) problems, and so the bias-reduction technique developed by Muñoz et al. [120] is limited in practice.

Analysis into the effect of the sampling technique on the reliability of landscape features is rarely performed, and a recent investigation by [119] suggests that properties can be significantly affected by the sampling technique. In addition, despite the importance of sampling, there are few explicit guidelines to follow and so critical choices, such as the sample size, are left to the discretion of practitioners [84]. The next section discusses and investigates the effects of two sampling considerations that can drastically alter sampling adequacy: the sample size and the distance metric.

4.2 Important Considerations

As previously mentioned, there are an infinite number of solutions in a continuous optimization problem, and a finite, but potentially enormous, number of solutions when a specified level of precision is assumed. Therefore, complete enumeration of $\mathcal{S}$ is largely impossible, and so a representative sample must contain *enough* solutions to adequately reflect the general structural features and trends in the problem. However, herein lies an extremely important consideration; how many solutions constitutes enough?

Assuming an unconstrained, unbounded continuous problem, $\mathcal{S} = \mathbb{R}^D$, and so the volume of the search space, and hence number of solutions at a specified precision, increases exponentially with dimensionality. Likewise, the number of solutions increases exponentially with D for rectangular bounds. The exponential growth rate is extremely problematic in practice because the number of solutions must increase exponentially with D in order to conserve the desired sampling density. For example, consider sampling $\mathcal{S} = [-1, 1]^D$ systematically with solutions distanced at increments of 0.1. For $D = 1$, merely 21 samples are required, however $21^{100} \approx 1.6670 \times 10^{132}$ solutions are required for $D = 100$. To put this into a practical computing perspective, the sampling density in 100 dimensions requires *at least* 430 bits to map each solution to a unique identifier, and hence $21^{100} \times 430 \approx 10^{134}$ bits are required to store all of the solutions. In comparison, Lloyd [99] derived that an "ultimate laptop" operating at the *physical limits* of both speed and memory has a total storage capacity of approximately 10^{31} bits.

In the context of landscape analysis, sampling is used to obtain a characteristic set of solutions from which to derive and analyse landscape features. The general consensus in the landscape analysis literature is that an adequate sample should yield robust (i.e. non-varying) features. Scaling sample sizes exponentially with D is obviously infeasible for large D, and so in practice sample sizes are generally made as large as practically possible, and the variability of the features is analysed to ensure the sample size is adequate [108, 118, 161]. Sizes of 10^3, 10^4 and 10^5 have been widely used for problems where $D \leq 100$ [74, 109, 118, 190]. Sample size has also been scaled linearly with D; [119] use $10^3 D$ samples to derive an ensemble of features, while [108] use $10^4 D$ samples. However, due to the exponential increase in volume, it is highly unlikely that a sample of 10^5 or $10^4 D$ solutions can adequately sample a 10-D, let alone 100-D, problem. After all, assuming merely 5 points are required to sample $[b_l, b_u]$, then 5^{100} points are required to maintain the same sampling density for $[b_l, b_u]^{100}$.

There is no definitive answer or recommendation as to how large the sample size should be, and the reasoning or motivation behind the sample choice is rarely discussed in the literature. Indeed, the work of Malan and Engelbrecht [108] and Müller and Sbalzarini [122] are exceptions; both state that their use of $10^4 D$ samples is motivated by the number of function evaluations used in a continuous optimization competition, where a maximum of

$10^4 D$ evaluations are used for $D = 10, 30$ and 50 [179]. They justify this by arguing that the main objective of optimization is to solve problems, and so computational effort devoted to problem analysis should not surpass the effort required to solve problems. This is a highly simplistic and narrow viewpoint; the knowledge gained from a greater understanding of problem structures (and how they relate to algorithm behaviour) does not just give insight, and hence solve, the problem at hand, but it can also be used to develop better algorithms for other problems. Hence, in this thesis, sample sizes are chosen to cover as much of the landscape as is practically feasible.

Another highly important sampling consideration is the choice of distance metric. While there are numerous distance metrics to choose from, many practitioners rely on canonical metrics, such as the well-known Hamming distance. Instead, [181] recommends that the distance associated with a landscape should be *coherent*, meaning that that it is related to the neighbourhood used by an algorithm of interest. Of course, this assumes that an algorithm is of interest, which is not the case if landscape analysis is used as a pre-processing step to aid practitioners in determining a suitable algorithm. In addition, assuming a particular algorithm is of interest, certain neighbourhoods may not have an immediately obvious or even practically feasible distance that is coherent (e.g. 2-opt for TSPs [181]). Hence, many practitioners simply rely on previously used canonical distances, and in doing so, further proliferate the acceptance for potentially undeserved distances.

The selection of an appropriate distance function can also be problematic when applying analysis techniques originating from one domain to another. For instance, many problem landscape analysis techniques originating from combinatorial optimization utilise distance functions that are inappropriate in continuous spaces, and so practitioners often substitute a metric that is (or at least seems) more appropriate. For example, Fitness Distance Correlation (FDC) was originally proposed using Hamming distance, however it is generally used with Euclidean distance in the continuous setting [54, 120, 122]. Problem analysis techniques originating in the continuous domain also utilise Euclidean distance [103]. The motivation behind the use of Euclidean distance is rarely discussed, and given that there are other metrics available, [3] speculate that its popularity (in the context of high dimensional database and indexing) stems from its traditional use in two and three dimensional spatial applications. However, as outlined by Theorem 4.1 below, the distance metric can impose important limitations on the resulting sample, and hence limitations on features derived from the sample.

Theorem 4.1 (Beyer et al. [13] and Aggarwal et al. [3])**.** *Given a sample of points $\mathcal{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\} \subset \mathbb{R}^D$, a query point, $\mathbf{x}^q \in \mathbb{R}^D$ and the L_p norm (denoted $\|.\|_p$):*

If $\lim_{D\to\infty} var\left(\frac{\|\mathcal{X}\|_p}{E[\|\mathcal{X}_p\|]}\right) = 0$, *then*

$$\frac{\max(\|\mathbf{x}^q, \mathbf{x}^i\|_p)}{\min(\|\mathbf{x}^q, \mathbf{x}^i\|_p)} \to 1 \tag{4.2.1}$$

As a consequence of Theorem 4.1, the notion of proximity becomes ill-defined as dimensionality increases for certain combinations of sampling distributions and distance metrics. Hence, for applications where a notion of proximity is fundamental, including landscape analysis techniques like dispersion and FDC, it is important to ensure that the combination of sample distribution and distance metric does not yield the convergent behaviour in Theorem 4.1.

There are a broad range of sampling distributions and distance metrics for which the theorem holds [13]. Specifically, it holds for the L_p norm when a sample of points is drawn independently and identically distributed (i.i.d.) (across dimensions) from a distribution with finite variance [3, 13]. Hence, a uniform random sample in conjunction with Euclidean distance (the L_2 norm) will result in the convergent behaviour of (4.2.1) as dimensionality increases. To illustrate how quickly the distance ratio in (4.2.1) converges, $\frac{\max(\|\mathbf{x}^i - \mathbf{x}^j\|)}{\min(\|\mathbf{x}^i - \mathbf{x}^j\|)}$ (where $\mathbf{x}^i \neq \mathbf{x}^j$) was calculated for solutions sampled randomly from $\mathcal{U}[0,1]^D$, where $D = 1, 2, \ldots, 30, 40, 50$. Four different sample sizes - specifically, $100D, 1000D, 10000D$ and $1000D^2$ - are investigated, and for each sample size at each value of D, 30 different samples are generated in order to obtain 30 estimates of the distance ratio. Figure 4.2 shows the mean and standard deviation (as error bars) of the 30 distance ratios for each sample size as dimensionality increases. Since $\frac{\max(\|\mathbf{x}^i - \mathbf{x}^j\|)}{\min(\|\mathbf{x}^i - \mathbf{x}^j\|)}$ is non-negative, error bars yielding negative values are omitted from the figure.

The results in Figure 4.2 show that for all four sample sizes, the distance ratio is converging towards 1 as D increases, and that the rates of convergence are comparable. The rates of convergence are initially quite fast; at $D = 1$, the maximum distance (for each sample size) is over four orders of magnitude larger than the minimum distance, however after only 20 dimensions, the maximum distance (for each sample size) is less than a single order of magnitude larger than the smallest distance. In addition, as D increases, the variance of the distance ratios between the sample sizes also becomes less significant. For example, the distance ratios for the sample sizes range between approximately 5.1×10^4 and 1.9×10^8 at

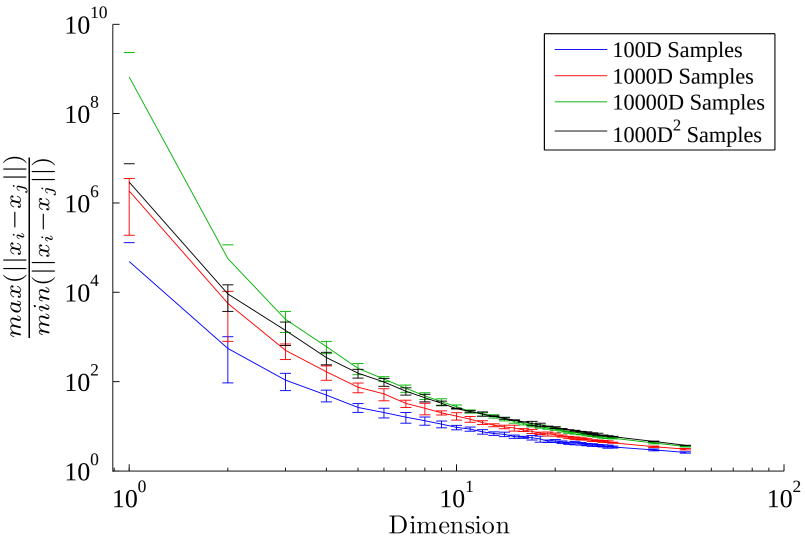


Figure 4.2: Behaviour of $\frac{\max(\|\mathbf{x}^i - \mathbf{x}^j\|)}{\min(\|\mathbf{x}^i - \mathbf{x}^j\|)}$ for $100D$, $1000D$, $10000D$ and $1000D^2$ points randomly sampled from $\mathcal{U}[0,1]^D$, where $D = 1, 2, \ldots, 30, 40, 50$.

$D = 1$, and only 2.6 and 3.7 at $D = 50$. The rates of convergence in Figure 4.2 also appear to slow as D increases, with the rate of convergence of the $1000D^2$ sample's distance ratio slowing more (and earlier) than the linear variants'. Figure 4.2 also shows that the standard deviation of the distance ratio generally decreases as dimensionality increases. This is rather intuitive; (4.2.1) indicates that as dimensionality approaches infinity, all points in the sample approach the same distance apart. Hence, as dimensionality increases, one expects decreasing variance in the distance values obtained (and hence the distance ratio).

To summarise, the sample size affects the distance ratio for small D (in our experiments, $D \leq 10$), however, as D increases, the distance ratio quickly converges to small values that tend towards 1, regardless of sample size. While the use of a sample size scaled quadratically with dimensionality appears to slow the convergence (compared to the linearly-scaled sample sizes), the reduction in convergence rate is slight and does not prevent the initial, fast decline of the distance ratio. The experiments conducted above are certainly not exhaustive, and samples of larger (yet sub-exponentially scaled) sizes will likely yield slower convergence rates, with perhaps less dramatic initial declines in the distance ratio. However, any sample size that doesn't match the exponential growth of the volume of $[0,1]^D$ as D increases will ultimately suffer from the convergence behaviour in (4.2.1).

Since many continuous problem properties utilise Euclidean distance and are based on samples (of sizes that are scaled sub-exponentially with D) of uniformly distributed points, an obvious research question is whether these properties are affected by the behaviour in (4.2.1), and if so, how?

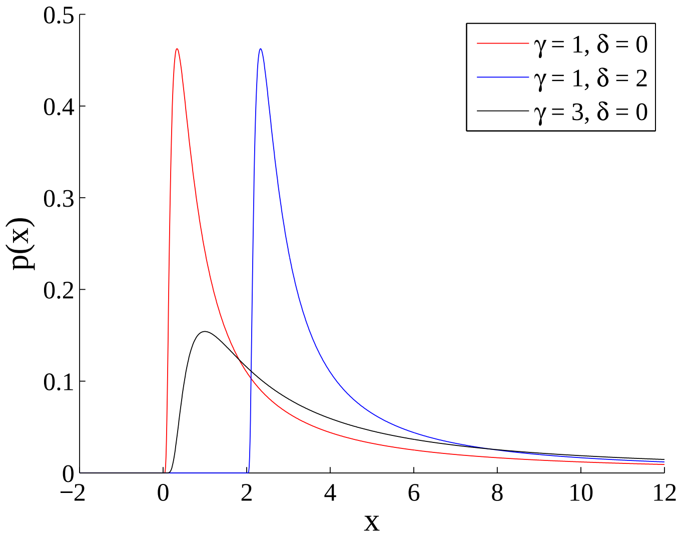


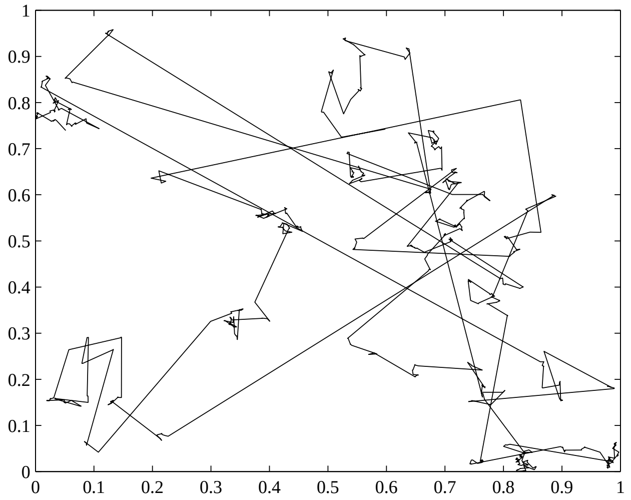
Figure 4.3: Example Lévy distributions.

4.3 Lévy Random Walks

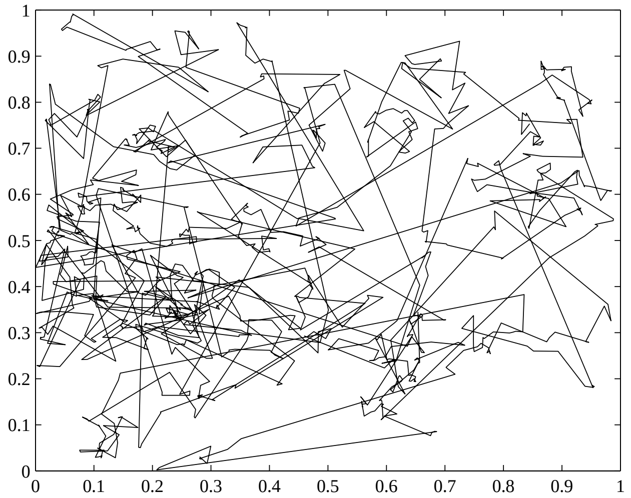
A Lévy random walk (also known as a Lévy flight) is a random walk where steps are taken in a random, isotropic direction and step sizes are sampled from a Lévy distribution [156]. Lévy walks have been observed in the hunting/foraging patterns of numerous animals, and is also an optimal strategy for sampling randomly distributed, sparse target sites [193]. This section outlines and proposes Lévy random walks for sampling continuous optimization problems. The application of Lévy random walks for sampling continuous optimization problems is novel, and the walks are subsequently used in the remainder of this thesis.

The Lévy distribution (also known as an Inverse Gaussian or Pearson V) is a continuous probability distribution for non-negative random variables. It is a member of the alpha-beta-stable distributions, where $\alpha = 0.5$ and $\beta = 1$ [91]. Lévy distributions are long-tailed, and are parameterised by two terms; scale (γ) and location (δ). Figure 4.3 shows three Lévy distributions and the effect of γ and δ.

By sampling step sizes from a Lévy distribution, the Lévy random walk has frequent small steps (dictated by γ and δ), and infrequent large steps, resulting in a sample with wide coverage of $\mathcal{S}$ and a variety of distances between solutions. δ essentially controls the minimum possible value, and in order to achieve steps of all sizes, it is set to 0 in all Lévy walks used throughout this thesis. The scale parameter is varied based on the size of the search space; spaces with a large area will need to have large γ to obtain steps that span the area. Figure 4.4 displays two example Lévy random walks of 1000 steps in $[0, 1]^2$ with $\delta = 0.001$ and $\delta = 0.01$. Experimental exploration of suitable γ values was conducted for all Lévy



(a) $\delta = 0.001$



(b) $\delta = 0.01$

Figure 4.4: Examples of Lévy random walks in $[0,1]^2$.

walks conducted in this thesis, and as a general rule, Lévy walks with $\gamma = 0.0005 \times \mathit{range}(\mathcal{S})$ provided adequate samples of high dimensional problems.

Assume a function exists to output D random variables from a Gaussian distribution with specified mean (μ) and standard deviation (σ): RandomGaussian(μ, σ, D). Similarly, assume a function exists to produce D random variables from a Lévy distribution with specified scale (γ) and location (δ): RandomLevy(γ, δ, D). The procedure used to conduct a Lévy random walk is outlined in Algorithm 4.1.

Like the aforementioned random walks, Lévy random walks are applicable to unbounded continuous optimization problems (the walk must simply be initialised at a feasible solution in the space). Bounded/constrained problems can also be sampled using a Lévy random

Algorithm 4.1 Levy Random Walk

```
Input:
    Search space, $\mathcal{S} \subseteq \mathbb{R}^D$
    Initial solution, $\mathbf{x} \in \mathcal{S}$
    Scale, $\gamma$
    Location, $\delta$
    Number of samples, $n$
 1: $\mathcal{S}'[1] \leftarrow \mathbf{x}$
 2: for $i \leftarrow 2$ to $n$ do
 3:     repeat
 4:         $\mathbf{g} \leftarrow \text{RandomGaussian}(0, 1, D)$
 5:         $\mathbf{direction} \leftarrow \frac{\mathbf{g}}{\|\mathbf{g}\|}$
 6:         $size \leftarrow \text{RandomLevy}(\gamma, \delta, 1)$
 7:         $\mathbf{x}_{next} \leftarrow \mathcal{S}'[i-1] + size \times \mathbf{direction}$
 8:     until $\mathbf{x}_{next} \in \mathcal{S}$
 9:     $\mathcal{S}'[i] \leftarrow \mathbf{x}_{next}$
10: end for
11: return $\mathcal{S}'$
```

walk, however the feasibility of proposed solutions must be satisfied (see line 8 of Algorithm 4.1).

To investigate whether the Lévy random walk exhibits a convergence of distances between solutions as dimensionality increases, the distance ratio, $\frac{\max(\|\mathbf{x}^i - \mathbf{x}^j\|)}{\min(\|\mathbf{x}^i - \mathbf{x}^j\|)}$ (where $\mathbf{x}^i \neq \mathbf{x}^j$), was calculated. Similar to the results in Figure 4.2, 30 different samples consisting of $1000D$ solutions were sampled from $\mathcal{S} = [0, 1]^D$ using a Lévy random walk, where $D = 1, 2, \ldots, 30, 40, 50$. Figure 4.5 shows the mean and standard deviation (as error bars) of the 30 distance ratios for each sample size as dimensionality increases. Since $\frac{\max(\|\mathbf{x}^i - \mathbf{x}^j\|)}{\min(\|\mathbf{x}^i - \mathbf{x}^j\|)}$ is non-negative, error bars yielding negative values are omitted from the figure.

The error bars in Figure 4.5 indicate that the standard deviation of the 30 distance ratios at each D is very low. Figure 4.5 show that unlike the uniform random sample in Figure 4.2, the distance ratios resulting from the Lévy random walk are not decreasing towards 0. The ratio is initially quite high (approximately 7.2735×10^6), as D initially increases, the ratio decreases to a minimum of 2.3977×10^4 at $D = 4$. However, at $D = 4$, the ratio stops decreasing and begins to increase slowly. This increase is likely due to the increasing diameter of $\mathcal{S}$, which in turn increases the largest possible Lévy step; because the minimum step is as close to (but not equal to 0) as precision allows, the distance ratio is increasing. The non-convergence of the distance ratio is a highly important result, as it shows that for applications where a notion of distance is imperative (e.g. many landscape analysis techniques), the Lévy random walk is favourable in comparison to uniform random sampling.

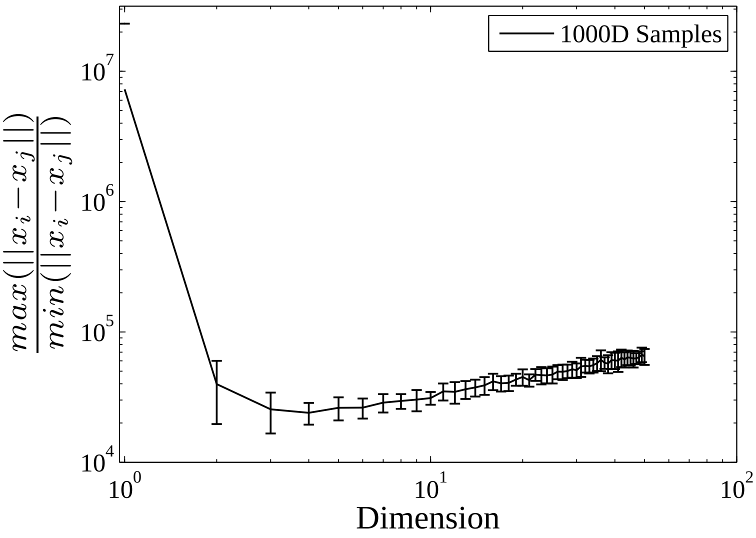


Figure 4.5: Behaviour of $\frac{\max(\|\mathbf{x}^i - \mathbf{x}^j\|)}{\min(\|\mathbf{x}^i - \mathbf{x}^j\|)}$ for $1000D$ points sampled from a Lévy random walk in $[0, 1]^D$, where $D = 1, 2, \ldots, 30, 40, 50$.

4.4 Case Study: Dispersion

Dispersion is a popular problem landscape feature that summarises the degree to which high-quality solutions are clustered in the search space. Lunacek and Whitley [103] originally calculated dispersion as the mean Euclidean distance between the fittest tn solutions from a uniform random sample of n solutions (see Section 3.2.6 for a review). While this methodology has been adopted by many practitioners, it is common to also normalise the dispersion by the diameter of the search space in order to allow equal comparison between problems (particularly of differing dimension) [57, 119, 121]. Resulting values of dispersion are in $[0, 1]$.

The following theorem presents an important result describing the limiting behaviour of dispersion as it is commonly employed (e.g. see [57, 103, 121]).

Theorem 4.2. *Given $t = 1$ and $\mathcal{S} = [0, 1]^D$, the dispersion of solutions sampled uniform randomly from $\mathcal{S}$ will converge to $\frac{1}{\sqrt{6}}$ as $D \to \infty$.*

Proof. Consider (independent) random variables $X_i, Y_i \sim \mathcal{U}[0, 1]$. Let $Z_i = (X_i - Y_i)^2$. Given that $E\left[X\right] = \frac{1}{b-a}\int_a^b x\mathrm{d}x$ for $X \sim \mathcal{U}[a,b]$:

$$\begin{aligned}
E\left[Z_i\right] &= E\left[(X_i - Y_i)^2\right] \\
&= E\left[X_i^2 - 2X_iY_i + Y_i^2\right] \\
&= E\left[X_i^2\right] - 2E\left[X_i\right]E\left[Y_i\right] + E\left[Y_i^2\right] \\
&= \int_0^1 x^2\mathrm{d}x - 2\int_0^1 x\mathrm{d}x\int_0^1 y\mathrm{d}y + \int_0^1 y^2\mathrm{d}y \\
&= \frac{1}{3}\left[x^3\right]_0^1 - 2\times\frac{1}{2}\left[x^2\right]_0^1 \times \frac{1}{2}\left[y^2\right]_0^1 + \frac{1}{3}\left[y^3\right]_0^1 \\
&= \frac{1}{3} - \frac{1}{2} + \frac{1}{3} \\
&= \frac{1}{6}
\end{aligned} \tag{4.4.1}$$

Hence $E\left[Z_i\right] = \frac{1}{6}$. Using Euclidean distance and normalising by $\sqrt{D}$ (i.e. the diameter of $\mathcal{S}$):

$$\begin{aligned}
\text{dispersion}(X,Y) &= \frac{1}{\sqrt{D}}\,E\left[\sqrt{\sum_{i=1}^{D} Z_i}\right] \\
&= E\left[\sqrt{\frac{1}{D}\sum_{i=1}^{D} Z_i}\right]
\end{aligned} \tag{4.4.2}$$

Using the Strong Law of Large Numbers, as $D \to \infty$:

$$\frac{1}{D}\sum_{i=1}^{D} Z_i \to E[Z_i]$$

Hence, as $D \to \infty$,

$$\begin{aligned}
\text{dispersion}(X,Y) &\to E\left[\sqrt{\frac{1}{D}\sum_{i=1}^{D} Z_i}\right] \\
&\to E\left[\sqrt{E\left[Z_i\right]}\right] \\
&\to E\left[\sqrt{\frac{1}{6}}\right] \\
&\to \frac{1}{\sqrt{6}}
\end{aligned} \tag{4.4.3}$$

■

The dispersion of an i.i.d. sample will converge to $\frac{1}{\sqrt{6}} \approx 0.4082$ as dimensionality increases, however because dispersion is typically based on a truncated sub-sample (i.e. $t < 1$), some problems may yield a non-i.i.d. sub-sample of fit solutions when truncation is used. Hence, the resulting dispersion values for these types of problems may not converge to $\frac{1}{\sqrt{6}}$.

The following investigation examines the behaviour of dispersion as dimensionality increases. To begin, the ability of dispersion to differentiate between two problems with drastically different distributions of fit solutions is examined. The two problems used are the Sphere function:

$$f_S(\mathbf{x}) = \sum_i^D x_i^2 \tag{4.4.4}$$

and the Reverse Sphere function:

$$f_{RS}(\mathbf{x}) = -f_S(\mathbf{x}) = -\sum_i^D x_i^2 \tag{4.4.5}$$

where $D = 2, \dots, 50, 100, 150, 200$. The Sphere function should have a low dispersion since the fittest solutions are relatively close together. In contrast, fit solutions in the Reverse Sphere function are widely distributed throughout $\mathcal{S}$ away from the origin, resulting in high dispersion. As previously discussed, the value of t may influence the behaviour of dispersion, and so for these experiments a range of values are investigated, specifically, $t = [0.0025, 0.01, 0.05, 0.1, 0.25]$. For each problem, 30 samples of $1000D$ solutions are generated randomly from $\mathcal{U}[-1, 1]^D$. The mean and standard deviation of the 30 resulting dispersions (at each value of D) for f_S and f_{RS} at the given thresholds is presented in Figure 4.6. Solid and dashed lines are used in Figure 4.6 to display the results for f_S and f_{RS} respectively.

Despite the different truncated selection thresholds, the dispersion for both f_S and f_{RS} appear to converge to $\frac{1}{\sqrt{6}}$. Figure 4.6 also shows that as t increases, the dispersion values are more tightly bound around $\frac{1}{\sqrt{6}}$. Importantly, the dispersion between these two functions becomes increasingly less discriminatory as dimensionality increases. That is, for two functions with drastically different distributions of fit solutions, the estimated dispersion values indicate the distributions are quite similar.

To further illustrate the behaviour of dispersion, the dispersion of the BBOB problem set (see Appendix A.1.2) at $t = 0.05$ was calculated using the same experimental setup and the results are shown in Figure 4.7[1]. Similar to the experiments above, 30 samples of $1000D$ solutions are generated randomly from $\mathcal{U}[-5, 5]^D$ for each problem. Lines in Figure 4.7 are

[1] Dispersion is calculated for $D = 2, \dots, 50, 100, 150, 200$ on BBOB problems generated with random seed equal to 1.

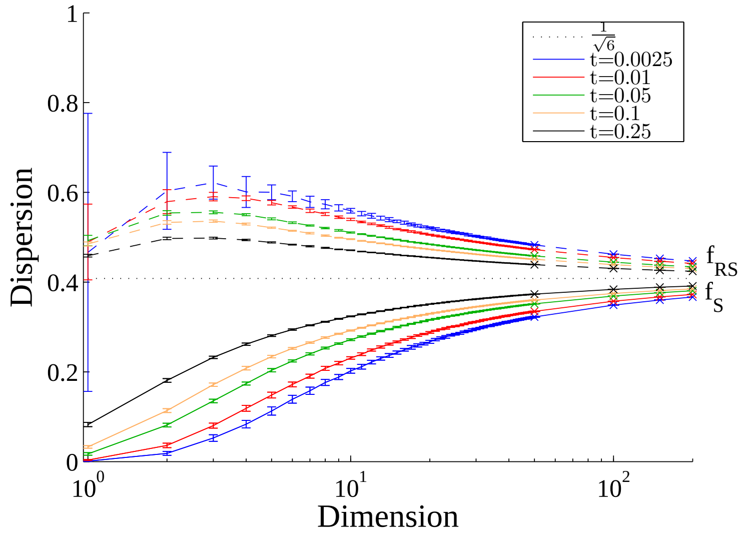


Figure 4.6: Behaviour of dispersion for the Sphere (solid lines) and Reverse Sphere (dashed lines) functions.

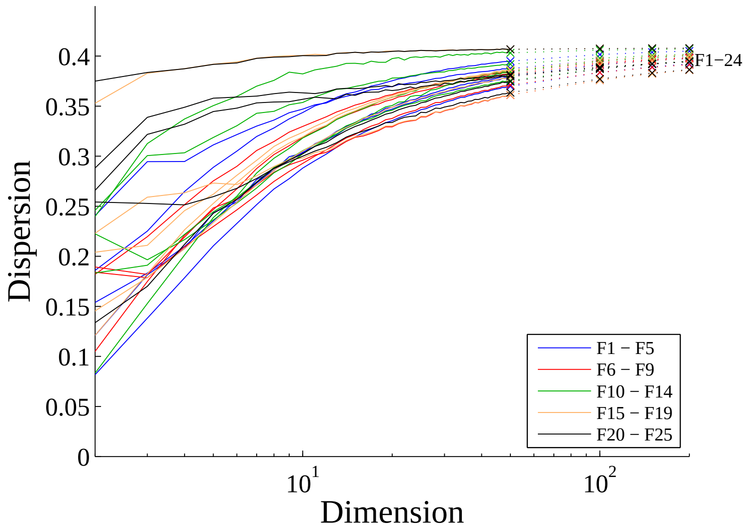


Figure 4.7: Dispersion for the BBOB problem set.

coloured/shaded according to each problem's classification within the benchmark set (see Table A.2). Because the dispersion values are quite similar across problems, error bars of the standard deviation of the 30 samples (per D) are not included as they obstructed the mean values. For most functions and dimensions, the standard deviation was typically around 6.5×10^{-4} (and at most 0.02). While dispersion discriminates between the different BBOB functions in low dimension, it is clear that dispersion becomes less discriminating and converges towards $\frac{1}{\sqrt{6}}$ as dimensionality increases.

The *difference in dispersion* was additionally proposed in [103] as the difference between the dispersion value when no selection is applied and the dispersion value when selection is applied. By Theorem 4.2, the dispersion of the full sample (when no selection is applied) converges to $\frac{1}{\sqrt{6}}$ as dimensionality increases. The results in Figures 4.6 and 4.7 indicate that for a number of problems of reasonable dimensionality, the dispersion at a variety of thresholds also seem to converge to $\frac{1}{\sqrt{6}}$. This suggests that as dimensionality increases, the difference in dispersion may converge to 0.

While dispersion has been used in a variety of applications, it is difficult to detect the convergent trend predicted by Theorem 4.2 within existing results reported in the literature. In [103], explicit dispersion values for only $D = 50$ problems are given, and so no comparisons with other dimensions can be made. However, when normalised by the diameter of the search space ($\sqrt{50}$), the values range from approximately 0.3041 to 0.4243, which is very similar to the results in Figure 4.7 for $D = 50$. Dispersion values are provided (or rather, difference in dispersion values) in [121], however only for 10, 30 and 50 dimensional problems. Despite having a sample of only 3 different dimensions, the results generally show the difference in dispersion values decreasing in magnitude towards 0 as dimensionality increases, in agreement with the argument above. While the dispersion values of the BBOB'10 problem set for 2, 3, 5, 10 and 20 dimensional problems were not explicitly reported in [120], subsequent analysis of their data shows that the dispersion values converge toward $\frac{1}{\sqrt{6}}$ as dimensionality increases.

To summarise, the current methodology used to calculate dispersion, namely the use of uniform random sampling in conjunction with Euclidean distance, has important limitations. These limitations have been shown theoretically and are clearly present on experiments conducted in this thesis, and in agreement with those in the literature. As a result, dispersion is severely restricted in its ability to adequately discriminate between problems.

4.4.1 Effects of Other Sampling Methodologies and Distances

The behaviour of the estimation of dispersion may be improved through a number of modifications to the methodology currently employed. This section focuses on three independent aspects of dispersion's implementation; the normalisation scheme, the distance metric and sampling strategy. For each aspect, a modification to dispersion is proposed and implemented, and the convergence of dispersion as dimensionality increases is assessed.

Modification 1: normalising by dispersion's bounds

One practical approach to improve the convergent behaviour of dispersion is by normalising the dispersion values by the "ideal" lower and upper dispersion bounds. That is, given a

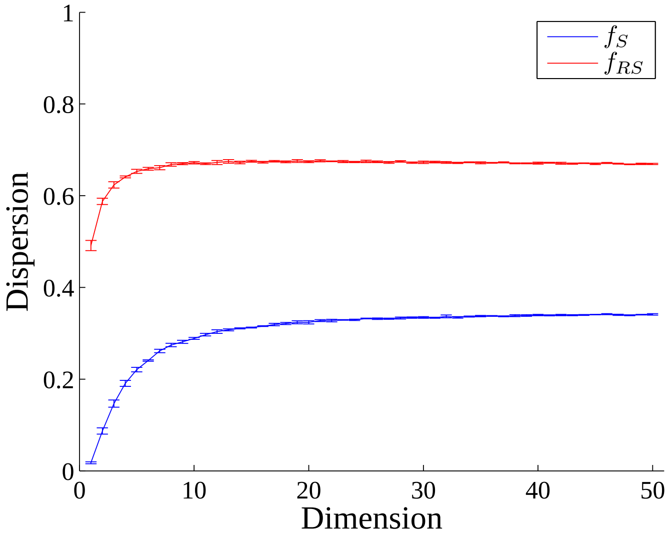


Figure 4.8: Bound-normalised dispersion for the Sphere (blue line) and Reverse Sphere (red line) functions.

sample of n solutions, there are $\frac{1}{2}n(n-1)$ possible distances (between pairs of solutions). From these, the distances of the fittest tn solutions are averaged in order to estimate the dispersion. However, from the $\frac{1}{2}n(n-1)$ distance values, the *tn smallest* distance values can be used to estimate the lowest (practically) possible value of dispersion. The same can be done for the *largest* values in order to obtain the largest (practically) possible value of dispersion. Then, the bounds can be used to normalise the problem's dispersion value.

Figure 4.8 shows the results of bound-normalising the Sphere and Reverse Sphere dispersion values, averaged over 30 samples of $1000D$ solutions (randomly sampled from $\mathcal{U}[-1,1]^D$) at each value of D. The threshold, t, was set to 5%, which is a typical value used in the literature [103]. The bound-normalised dispersion values in Figure 4.8 clearly discriminate between the two types of functions, and do not exhibit the convergence to $\frac{1}{6}$.

Figure 4.9 shows the results of bound-normalising the BBOB dispersion values, averaged over 30 samples of $1000D$ solutions (randomly sampled from $\mathcal{U}[-5,5]^D$) at each value of D. The standard deviations of these estimates for most functions and dimensions were generally very low (at around 2.4×10^{-3}), and were at most 0.02.

The dispersion values in Figure 4.9 are much more discriminatory than those in Figure 4.7. That is, the problems are well-separated and allow better categorisation. Furthermore, the convergence behaviour is no longer present; the dispersion values are consistent and stable as dimensionality increases. These results suggest that the bound-normalised dispersion is a significant improvement on the original dispersion methodology.

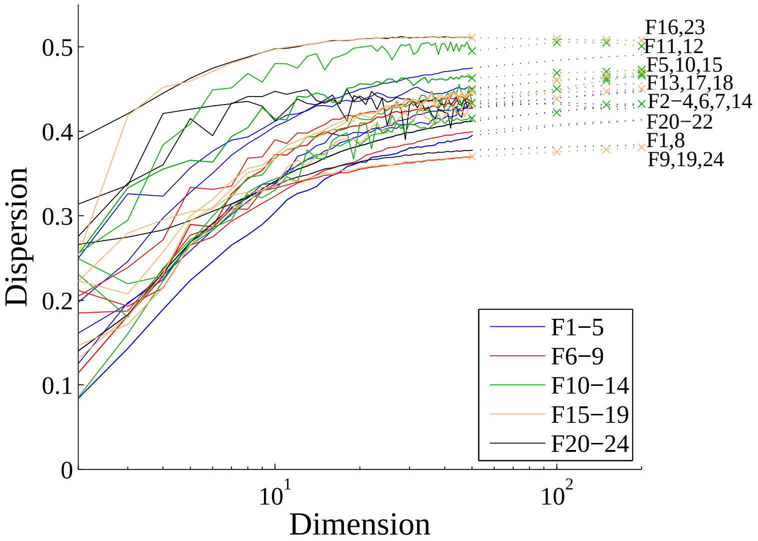


Figure 4.9: Bound-normalised dispersion for the BBOB problem set.

Modification 2: L_p norm with $p = 0.1$, $p = 0.5$ and $p = 1$

The rate of convergence of (4.2.1) has been shown to be sensitive to the value of p for the L_p norm [3]. In particular, lower (indeed, even fractional) values of p produced better contrast between the maximum and minimum distance than larger values of p. While this doesn't prevent the convergence behaviour, using an L_p norm with fractional p-values may at least improve discrimination between problems of modest dimensionality.

The following experiment calculates the dispersion of the Sphere, Reverse Sphere and BBOB problems using the L_p norm with $p = 0.1$, $p = 0.5$, $p = 1$ and $p = 2$ ($p = 2$ was also used to generate Figures 4.6 and 4.7). The aim of the experiment is to investigate whether low values of p in the L_p norm improve dispersion's behaviour. Figures 4.10 and 4.11 show the resulting dispersion values of the problems averaged over 30 samples of $1000D$ solutions (randomly sampled from $\mathcal{U}[-5,5]^D$ and $\mathcal{U}[-5,5]^D$ respectively) at each value of D. The standard deviation for most BBOB functions and dimensions was generally very low (at around 6.7×10^{-4}), and was at most 0.02.

Figures 4.10 and 4.11 display convergent behaviour for all four values of p. Lower values of p result in a slower convergence, although it seems by only a constant factor. In addition, the BBOB problems remain clustered together, with no significant improvement in separability compared to Figure 4.7. Hence while the results give an indication of the empirical differences that can be expected for different values of p, use of the L_p norm with small p values in general does not appear to significantly improve dispersion estimates.

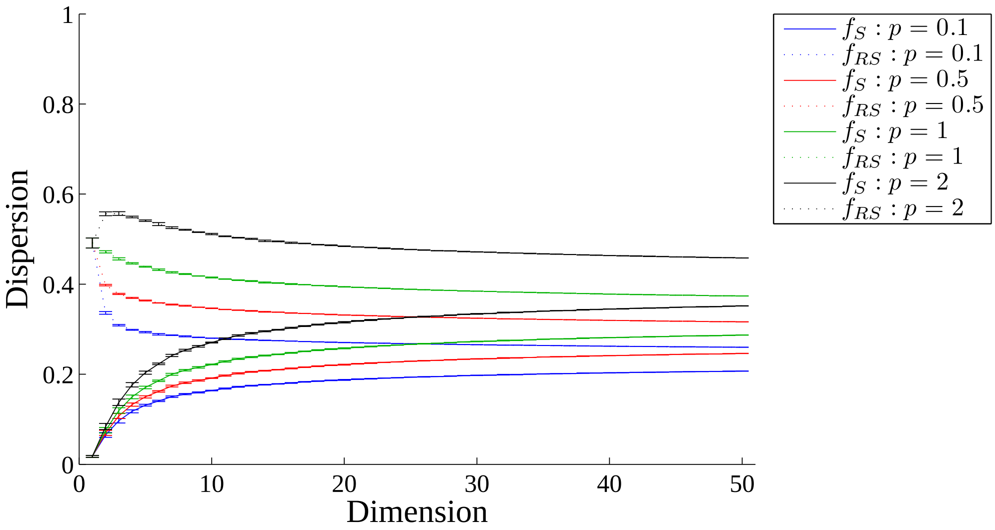


Figure 4.10: Dispersion using L_p norms where $p = 0.1$, 0.5, 1 and 2 for the Sphere (solid lines) and Reverse Sphere (dashed lines) functions.

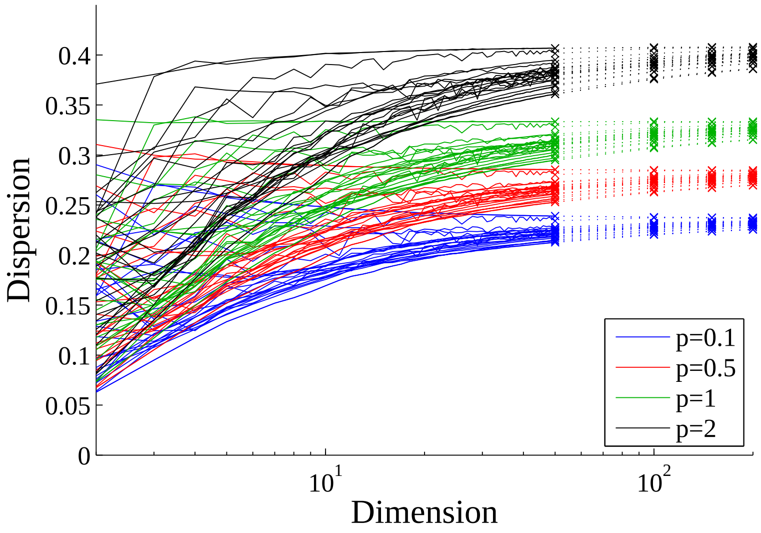


Figure 4.11: Dispersion using L_p norms where $p = 0.1$, 0.5, 1 and 2 for the BBOB problem set.

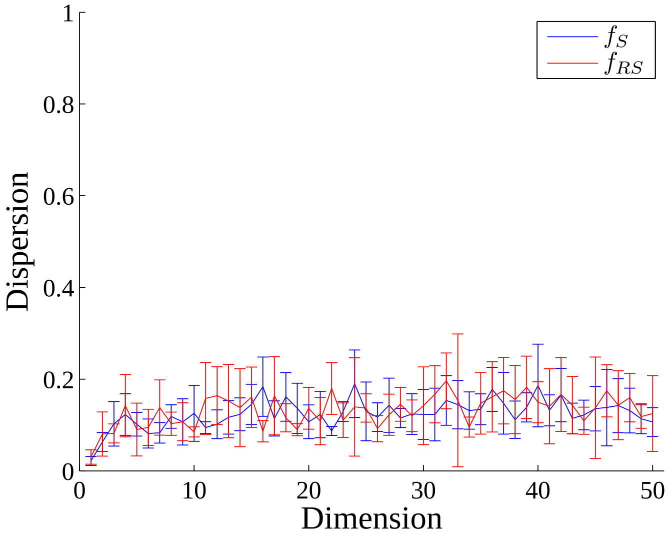


Figure 4.12: Dispersion using fixed-step (of size 10^{-2}) random walks for the Sphere (blue line) and Reverse Sphere (red line) functions.

Modification 3: Fixed-step Random Walks

Because (4.2.1) is dependent on the type of distance metric and sampling technique, using a distance metric and/or sampling technique where the theorem does not hold may improve the behaviour of dispersion. Here, a fixed-step isotropic random walk is used to sample $\mathcal{S}$, which is subsequently used to estimate dispersion. As the name suggests, a fixed-step random walk takes steps in a uniform random direction, of a specific size, α. While each solution is α away from the previous and next solutions in the walk, the isotropic nature of the walk means that a wide variety of distances are possible (e.g. the distance between a given solution and a solution 4 steps away is within $[0, 4\alpha]$). The aim of the experiment is to determine whether the dispersion estimates are affected when a fixed-step random walk is used to sample $\mathcal{S}$. For this experiment, $1000D$ steps of size $\alpha = 10^{-2}$ (Sphere and Reverse Sphere) and $\alpha = 1$ (BBOB) are taken to give wide coverage of the search space.

Figures 4.12 and 4.13 shows the mean dispersion values for the Sphere, Reverse Sphere and BBOB problems (averaged over 30 samples) at each value of D. The random walks for the BBOB problems in Figure 4.13 resulted in slightly larger standard deviations than the other improvements, however they were still quite low and generally remained constant at 0.04 (and at most 0.07).

Modification 4: Lévy Random Walks

Similar to the fixed-step random walk, the use of Lévy random walks may also impact on the estimation of dispersion. Figure 4.14 shows the dispersion of the Sphere and Reverse

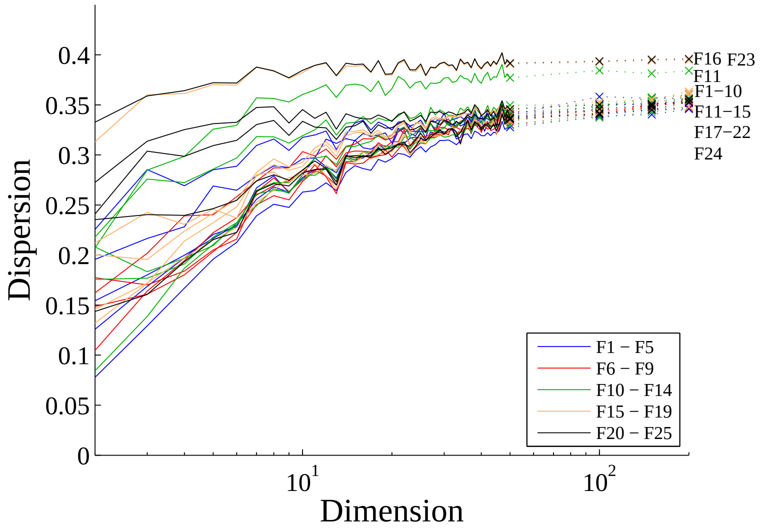


Figure 4.13: Dispersion using fixed-step (of size 1) random walks for the BBOB problem set.

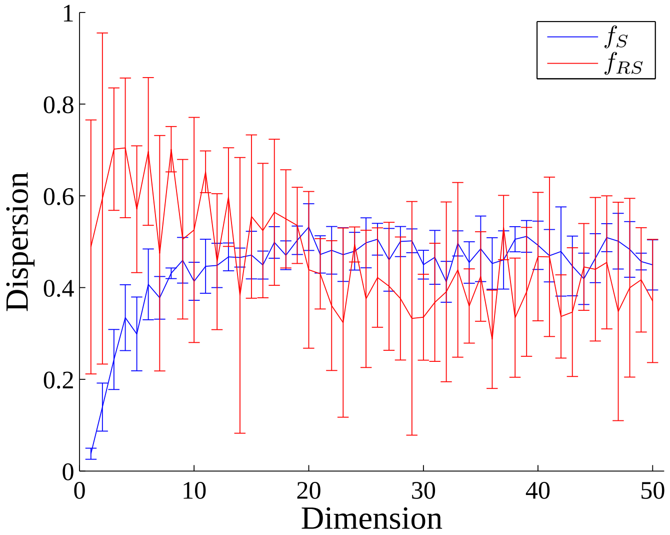


Figure 4.14: Dispersion using a Lévy random walk for the Sphere (blue line) and Reverse Sphere (red line) functions.

Sphere problems, calculated from samples of $1000D$ solutions, obtained using a Lévy random walk ($\gamma = 10^{-3}$, $\delta = 0$). Similarly, Figure 4.15 shows the dispersion of the BBOB problems calculated from samples of $1000D$ solutions, obtained using a Lévy random walk ($\gamma = 0.1, \delta = 0$).

While the dispersion values calculated for the Sphere and Reverse Sphere functions shown in Figure 4.14 offer limited improvements, the dispersion values in Figure 4.15 are a considerable improvement to the original dispersion values shown in Figure 4.7. In particular, the dispersion values from the Lévy random walk samples do not appear to converge, and they provide a wide range of values that differentiate the problems well and can therefore facilitate in problem classification/categorisation. The dispersion values in Figure 4.15

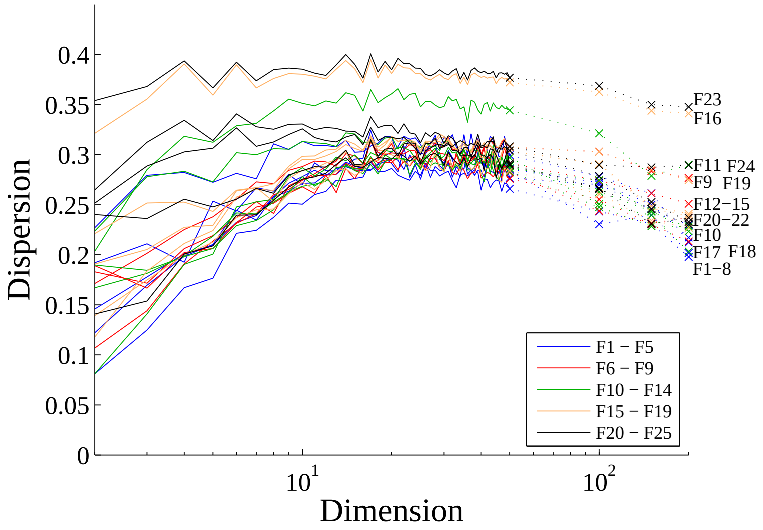


Figure 4.15: Dispersion using a Lévy random walk for the BBOB problem set.

are initially quite varied, and as D increases, all values initially increase. At approximately $D = 22$, the dispersion values stop increasing and appear to slightly decrease, however, the relative ordering of problems (by their dispersion values) is consistent across D.

4.4.2 Summary

The interaction between sampling strategy and distance metric is complex and can introduce interesting behaviour in a wide variety of applications. This case study examined the behaviour of the dispersion metric, which utilises uniform random sampling in conjunction with Euclidean distance. A theoretical argument was developed showing that using the existing methodology to calculate dispersion, the dispersion of the full sample will converge to $\frac{1}{\sqrt{6}}$ as dimensionality reached the infinite limit. Experimental analysis on the Sphere, Reverse Sphere and BBOB problems resulted in convergent dispersion values. Furthermore, dispersion was deemed incapable of adequately distinguishing between high and low dispersive structure. Importantly, it is the *existing* methodology behind dispersion that is flawed, rather than the actual concept of dispersion. Hence, in an attempt to improve dispersion, four independent modifications to its underlying methodology were proposed; bound-normalisation, the L_p norm with low p, a fixed-step random walk and a Lévy random walk. The modifications proposed are simple and do not add significant complexity or computational effort to dispersion's original methodology. Encouragingly, the bound normalisation, fixed-step random walk and Lévy random walk were shown to improve the convergent behaviour and increase separability between problems.

4.5 Case Study: Fitness Distance Correlation

Fitness Distance Correlation (FDC) measures the extent of correlation between the fitness values of a set of solutions and their distance to a given reference solution, $\mathbf{x}'$ (usually the global optimum). FDC was originally proposed in the combinatorial problem context, and is often used with distance functions based on specific algorithm move operators [5, 83]. However in the continuous optimization context, Euclidean distance is typically used [54, 120, 122].

As defined in Theorem 4.1 and illustrated in the case study for dispersion in Section 4.4, the use of Euclidean distance in conjunction with certain sampling schemes may not be ideal. The following case study investigates the degree to which FDC is effected by the use of Euclidean distance and a uniform random sample. Similar to Section 4.4, the consequences of convergent distances on FDC coefficients is theoretically analysed. Next, FDC is empirically examined using the Sphere function, Reverse Sphere function and BBOB problem set (over a variety of dimensions) to assess whether it is being affected. Then, different distance metrics and sampling methodologies are proposed and their effect on FDC is analysed.

The formulation for FDC (Equation 3.2.5) is heavily reliant on an accurate notion of distance between solutions sampled and the reference solutions. Theorem 4.1 in Section 4.2 essentially states that for a broad range of distributions of points, the contrast between the largest distance and the smallest distance becomes non-existent for certain distance metrics as dimensionality increases. For FDC calculated from a uniform random sample in conjunction with Euclidean distance, all distances in $\mathcal{D}$ will converge to a constant, c. Consequently, as dimensionality, D, increases,

$$\lim_{D\to\infty} \mu_D = c$$

and

$$\lim_{D\to\infty} \sigma_D = 0$$

Because $\lim_{D\to\infty} d_i = c \; \forall \; d_i \in \mathcal{D}$ and $\lim_{D\to\infty} \mu_{\mathcal{D}} = c$, the resulting value for $\lim_{D\to\infty}(d_i - \mu_{\mathcal{D}})$ is 0. Ignoring the contributions of $f(\mathbf{x}^i)$ and μ_F (as they will vary depending on f) and substituting $\lim_{D\to\infty}(d_i - \mu_{\mathcal{D}}) = 0$:

$$
\begin{aligned}
C_{\mathcal{FD}} &= \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}^i) - \mu_{\mathcal{F}})(d_i - \mu_{\mathcal{D}}) && (4.5.1)\\
\lim_{D\to\infty} C_{FD} &= \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}^i) - \mu_F) \times 0 && (4.5.2)\\
&= 0
\end{aligned}
$$

Consequently, in the infinite limit the FDC coefficient is:

$$\lim_{D\to\infty} FDC = \lim_{D\to\infty} \frac{C_{\mathcal{FD}}}{\sigma_{\mathcal{F}}\sigma_{\mathcal{D}}} \tag{4.5.3}$$

Because both the numerator and denominator of Equation 4.5.3 are 0 in the limit, the rate at which each converge to 0 will dictate the limiting value of FDC. This is difficult to derive analytically, since the terms $(f(\mathbf{x}^i) - \mu_F)$ and $\sigma_{\mathcal{F}}$ will affect the convergence rates, but are dependent on the objective function under consideration. Therefore, experimental analysis is conducted below to determine the convergence behaviour of FDC coefficients for a variety of problems.

It is important to stress that the above theory is only applicable to the scenario where FDC is calculated using a *reference point* from a distribution (as well as an appropriate sample of solutions and distance metric) satisfying the properties outlined in [3, 13]. For example, the theory applies if both the reference point and $\mathcal{S}'$ are sampled uniformly from $\mathcal{S}$. Conversely, if $\mathcal{S}'$ is sampled uniformly from $\mathcal{S}$, but the reference point is a specific point of interest (e.g. the origin, global optimum etc), the theory will not apply. Hence, FDC is only affected by the convergence theory discussed above if the reference point satisfies the conditions.

In this case study, two variants of FDC are calculated: $FDC_{\mathbf{x}^*}$, which uses the global optimum ($\mathbf{x}^*$) as the reference point (and hence should be unaffected by the theory), and $FDC_{\hat{\mathbf{x}}^*}$, which uses the best solution ($\hat{\mathbf{x}}^*$) in the sample as the reference point (and hence may be affected by the theory). The Reverse Sphere function has 2^D global optima (located in the corners of the hypercube) and so for each solution in the sample, the distance between it and its *closest* global optimum is used. The same experimental setup used in the dispersion case study (Section 4.4) is used in the following. Specifically;

- $D = 2, \ldots, 50, 100, 150, 200.$

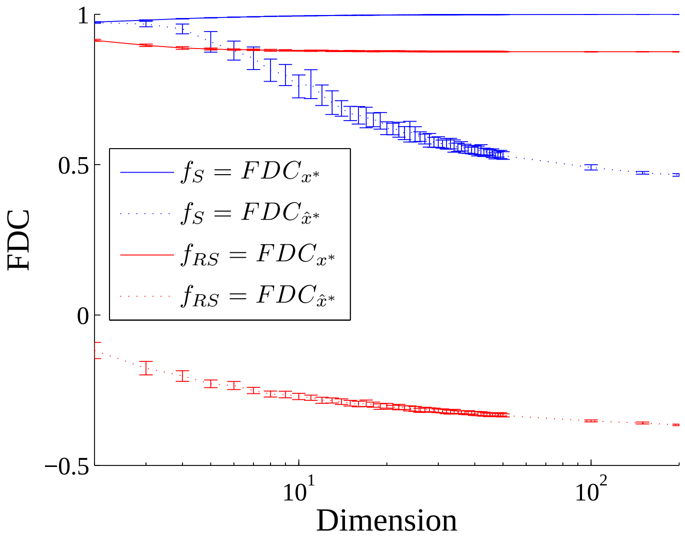
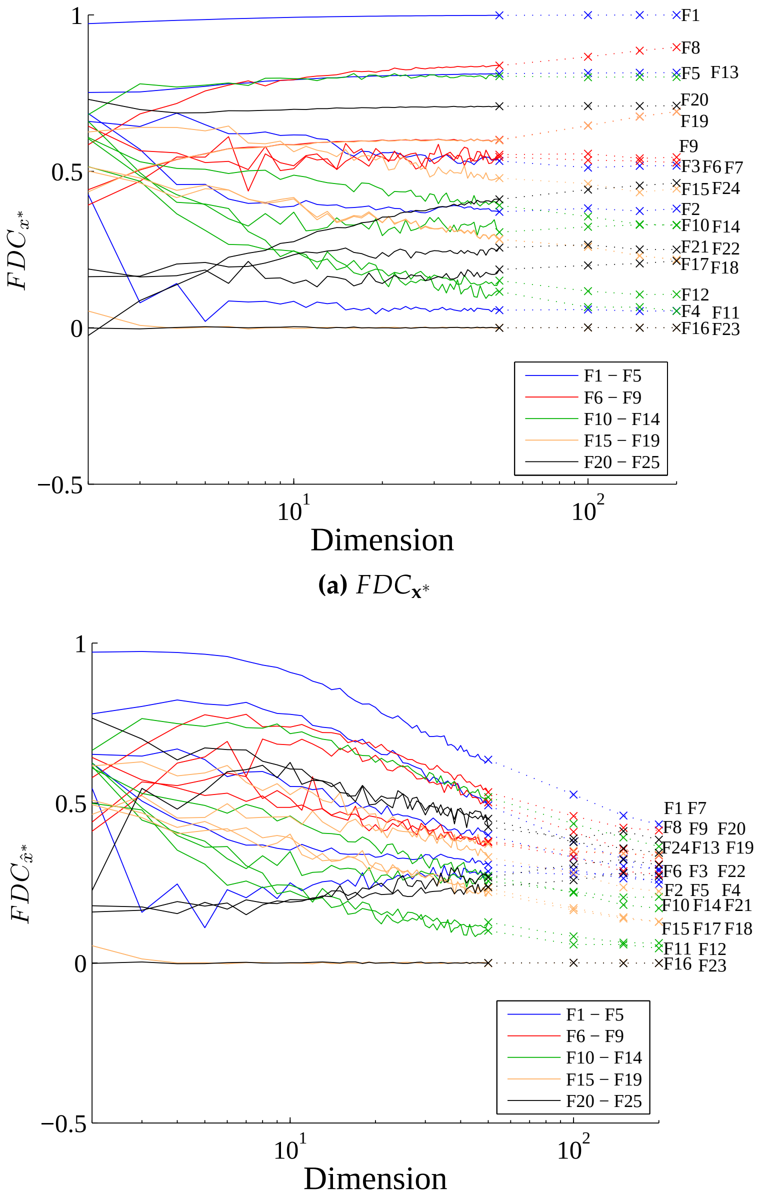


Figure 4.16: $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ for the Sphere and Reverse Sphere functions for $D = 2, \ldots,$ $50, 100, 150, 200$.

- $\mathcal{S} = [-1,1]^D$, $\mathcal{S} = [-1,1]^D$ and $\mathcal{S} = [-5,5]^D$, for the Sphere, Reverse Sphere and BBOB problem set respectively.
- For a given problem, 30 samples of $1000D$ solutions are generated uniform randomly from $\mathcal{S}$.

To begin, the FDC coefficients are calculated for samples of the Sphere (f_S) and Reverse Sphere (f_{RS}) functions defined in Equations 4.4.4 and 4.4.5. The mean and standard deviation (shown as error bars) of the 30 resulting FDC coefficients (at each value of D) for f_S and f_{RS} are presented in Figure 4.16. Blue and red coloured lines are used in Figure 4.16 to display the results for f_S and f_{RS} respectively, and $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ are drawn using solid and dotted lines respectively.

All FDC coefficients shown in Figure 4.16 have extremely small standard deviations between samples (the largest standard deviation was 0.0480). Both the Sphere and Reverse Sphere's $FDC_{\mathbf{x}^*}$ values remain quite constant as D increases. The calculation of $FDC_{\mathbf{x}^*}$ uses a reference point that is *not* in the sample. Because the distances are calculated between each solution in the sample and the reference point (not in the sample), the theory regarding the convergence of distance does not apply. Consequently, the $FDC_{\mathbf{x}^*}$'s shown in Figure 4.16 do not appear to be affected by D. In contrast, the $FDC_{\hat{\mathbf{x}}^*}$'s calculated for both f_S and f_{RS} is affected by increasing D. In particular, the $FDC_{\hat{\mathbf{x}}^*}$'s for the Sphere function seem to be greatly affected. While $FDC_{\mathbf{x}^*} \approx 1$ for f_S, its $FDC_{\hat{\mathbf{x}}^*}$'s begins to decrease at $D = 5$ and $FDC_{\hat{\mathbf{x}}^*}$ reaches a low of approximately 0.4658 at $D = 200$. The $FDC_{\hat{\mathbf{x}}^*}$ for f_{RS} also decreases as D increase,



(a) $FDC_{\mathbf{x}^*}$

(b) $FDC_{\hat{\mathbf{x}}^*}$

Figure 4.17: $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ (using L_2) for the BBOB problem set for $D = 2, \ldots, 50, 100, 150, 200$.

although the decrease is not as drastic as f_S. Specifically, $FDC_{\hat{\mathbf{x}}^*}$ for the Reverse Sphere function begins at approximately -0.1179, and decreases steadily until approximately -0.3653 at $D = 200$. Hence, the difference between $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ is approximately 0.5342 for f_S, and only 0.2474 for f_{RS}.

To further investigate the use of Euclidean distance with uniform random sampling to calculate FDC, Figure 4.17 shows the results of $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ calculated on the BBOB problem set (see Appendix A.1.2).

The $FDC_{\mathbf{x}^*}$ coefficients in Figure 4.17a are clearly invariant to D. However, similar to the experiments with f_S and f_{RS} shown in Figure 4.17, the $FDC_{\hat{\mathbf{x}}^*}$ coefficients decrease and appear to be converging as D increases. As previously mentioned, $FDC_{\mathbf{x}^*}$ uses a reference solution that is *not* in the uniform random sample, while $FDC_{\hat{\mathbf{x}}^*}$ uses a reference solution that is. Hence, the convergence theory is really only applicable to $FDC_{\hat{\mathbf{x}}^*}$, and Figure 4.17b confirms that the theory is relevant in practice.

4.5.1 Effects of Other Sampling Methodologies and Distances

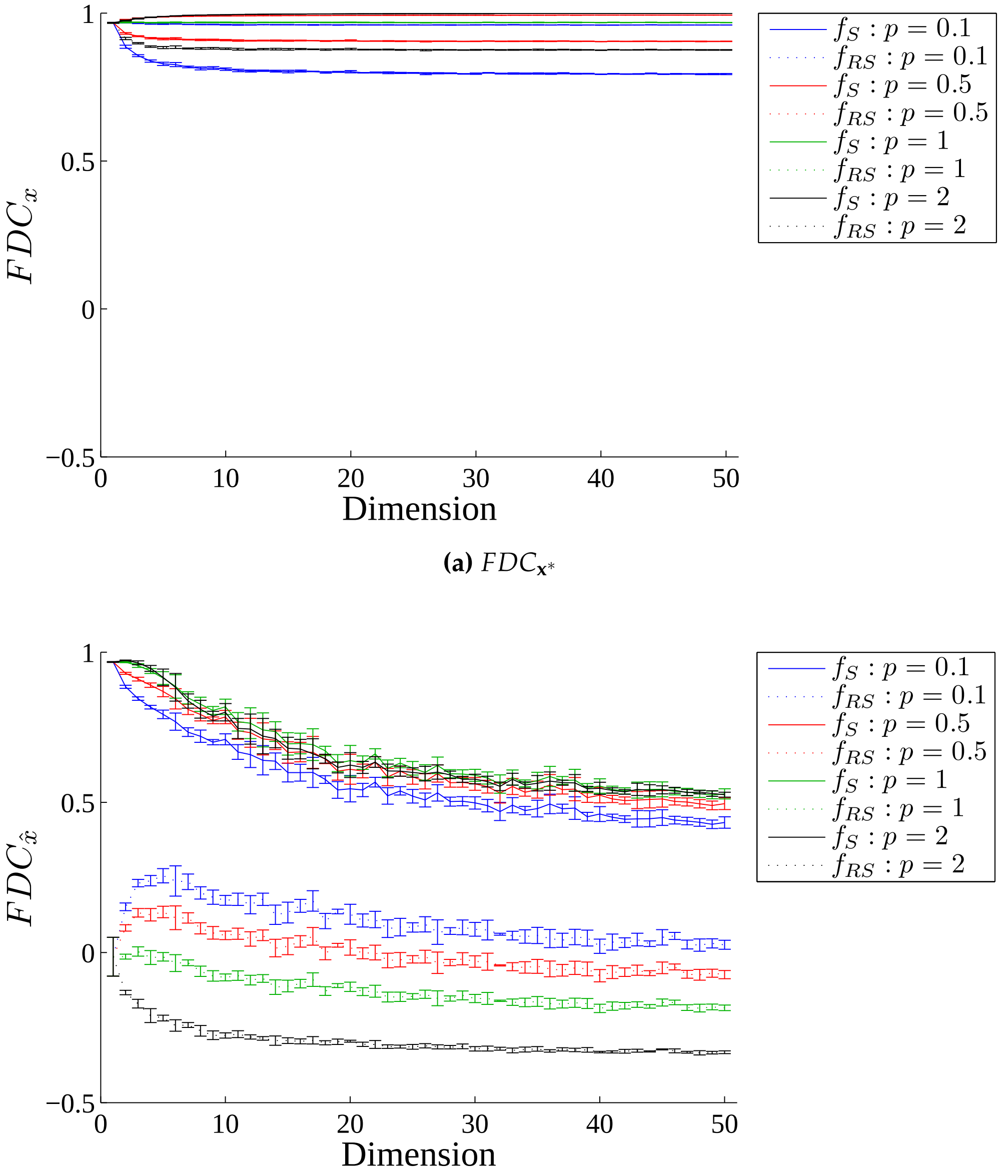
While the $FDC_{\mathbf{x}^*}$ coefficients calculated on the BBOB problems shown in Figure 4.17 were invariant to increasing dimensionality, the $FDC_{\hat{\mathbf{x}}^*}$ coefficients exhibited convergent behaviour. This section focuses on three different sampling methodologies and distance metrics that may slow or prevent the convergence of $FDC_{\hat{\mathbf{x}}^*}$ coefficients; 1) the L_p norm with small p values as a distance metric; 2) a fixed-step random walk; and 3) a Lévy random walk. For each proposed modification, $FDC_{\mathbf{x}^*}$ coefficients are also examined to ensure that the modifications to not adversely affect the coefficients.

Modification 1: L_p **norm with** $p = 0.1$**,** $p = 0.5$ **and** $p = 1$

The following experiment investigates the effect of using the L_p norm with $p = 0.1$, $p = 0.5$ and $p = 1$ in calculating the FDC coefficients of the Sphere, Reverse Sphere and BBOB problems (note that Figures 4.16 and 4.17 shows results for $p = 2$). Figures 4.18 to 4.21 shows the resulting FDC coefficients of the Sphere, Reverse Sphere and BBOB problems averaged over 30 samples of $1000D$ solutions (randomly sampled from $\mathcal{U}[-1,1]^D$ and $\mathcal{U}[-5,5]^D$ respectively) at each value of D.

The standard deviation for $FDC_{\mathbf{x}^*}$ on the BBOB functions was generally very low (average of 0.0396, 0.0434 and 0.0451 for $p = 0.1, 0.5$ and 1 respectively), and was at most 0.3729, 0.3868 and 0.3947 for $p = 0.1, 0.5$ and 1 respectively. The standard deviation for $FDC_{\hat{\mathbf{x}}^*}$ on most functions and dimensions was also generally very low (average of 0.0517, 0.0573 and 0.0606 for $p = 0.1, 0.5$ and 1 respectively), and was at most 0.3615, 0.3764 and 0.3854 for $p = 0.1, 0.5$ and 1 respectively.

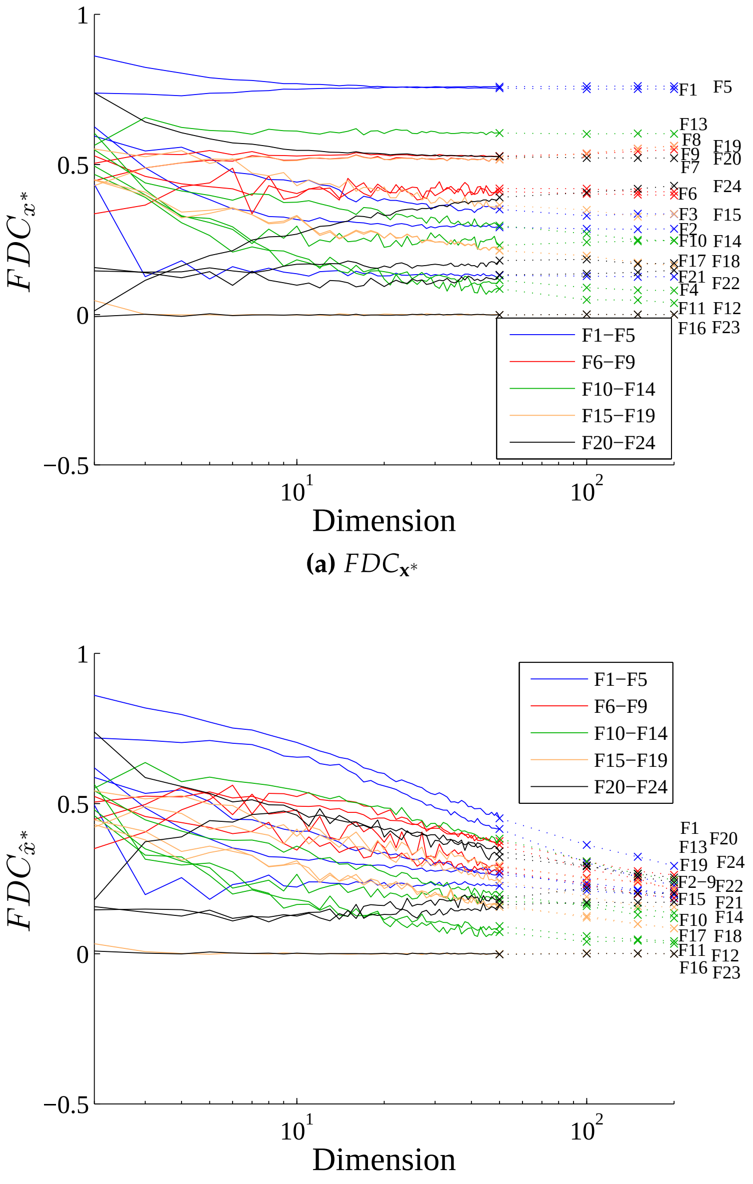
The results in Figures 4.18 to 4.21 show that the $FDC_{\mathbf{x}^*}$ coefficients are consistent across D, in contrast to the $FDC_{\hat{\mathbf{x}}^*}$ coefficients, which converge towards 0 as D increases. The non-convergence of the $FDC_{\mathbf{x}^*}$ provide further evidence that using a reference point that is not within the sample prevents convergence. The rate of convergence of the $FDC_{\hat{\mathbf{x}}^*}$ coefficients varies between the different p values. For $p = 0.1$, the $FDC_{\hat{\mathbf{x}}^*}$ coefficients shown in Figure 4.19 decrease very consistently in a straight line (and hence at a logarithmic rate, due to



(a) $FDC_{\mathbf{x}^*}$

(b) $FDC_{\hat{\mathbf{x}}^*}$

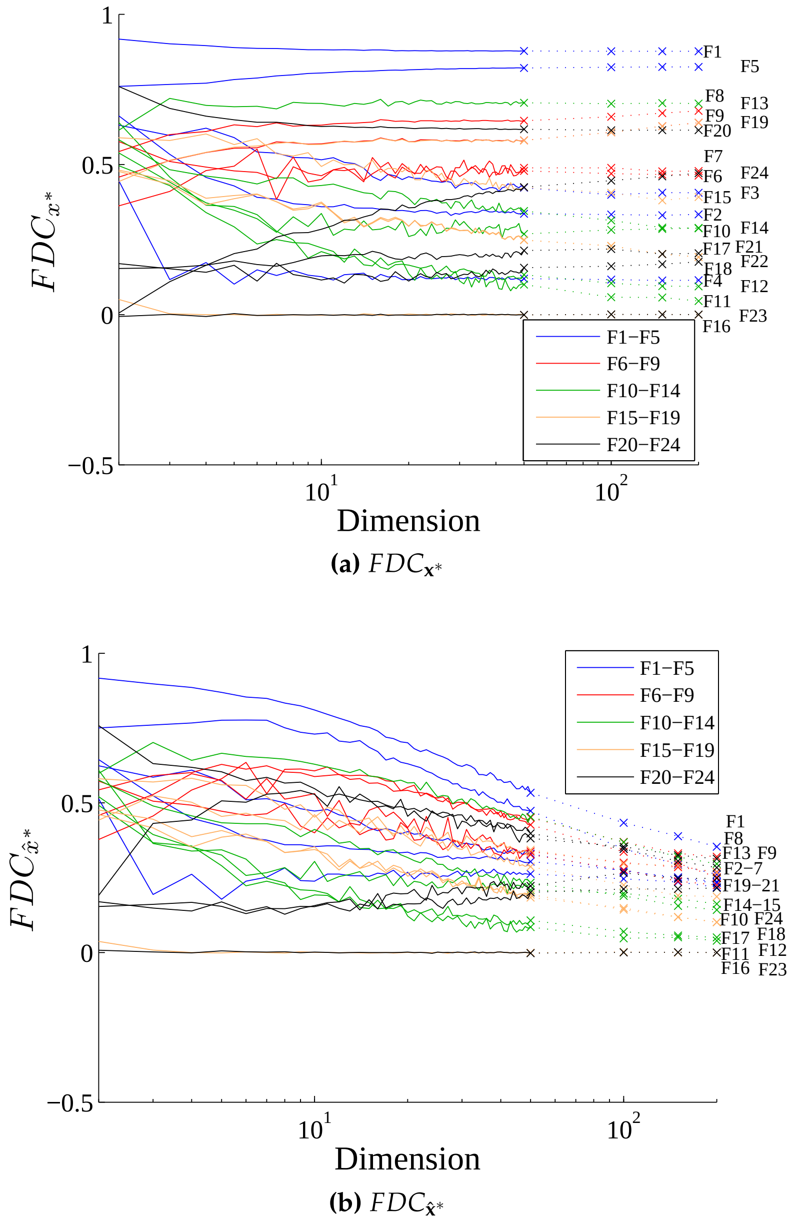
Figure 4.18: $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ using $L_{0.1}$, $L_{0.5}$, L_1 and L_2 for the Sphere and Reverse Sphere functions.



(a) $FDC_{\mathbf{x}^*}$

(b) $FDC_{\hat{\mathbf{x}}^*}$

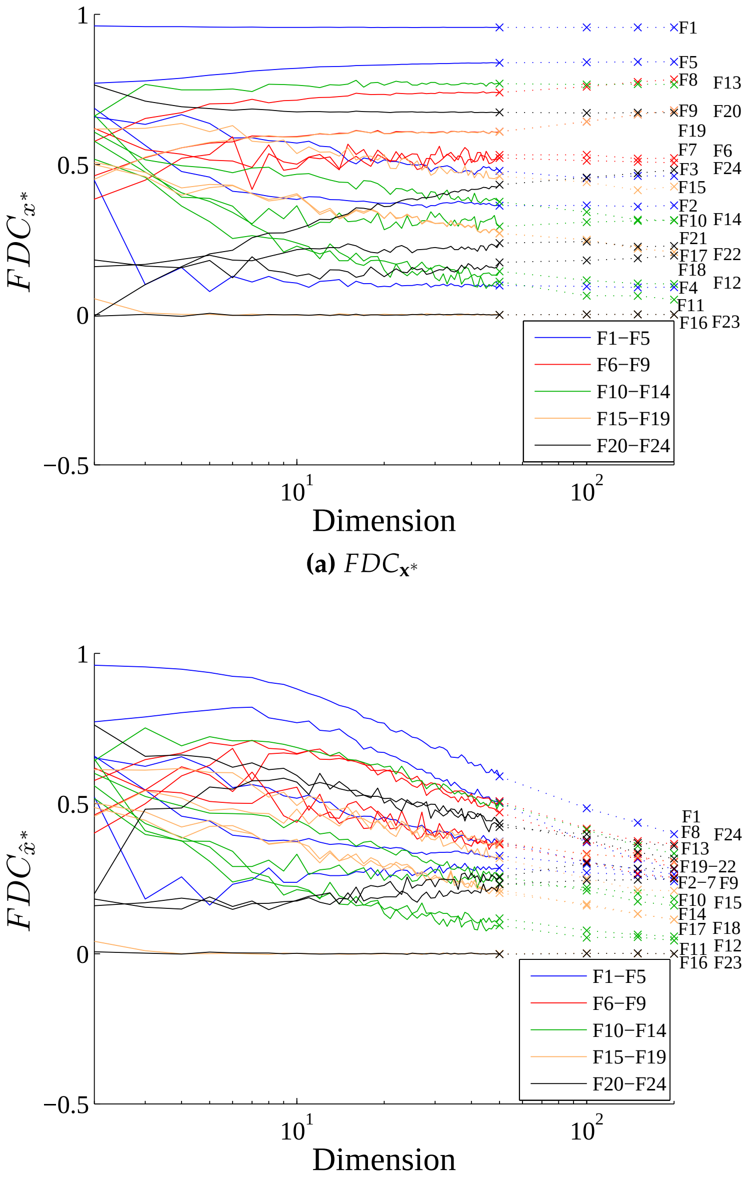
Figure 4.19: $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ using $L_{0.1}$ for the BBOB problem set for $D = 2, \ldots, 50, 100,$ 150, 200.



(a) $FDC_{\mathbf{x}^*}$

(b) $FDC_{\hat{\mathbf{x}}^*}$

Figure 4.20: $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ using $L_{0.5}$ for the BBOB problem set for $D = 2, \ldots, 50, 100, 150, 200$.



(a) $FDC_{\mathbf{x}^*}$

(b) $FDC_{\hat{\mathbf{x}}^*}$

Figure 4.21: $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ using L_1 for the BBOB problem set for $D = 2, \ldots, 50, 100, 150, 200$.

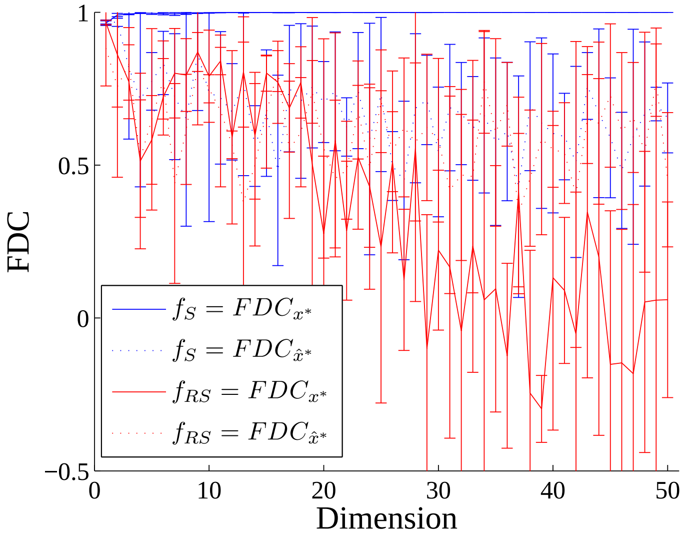
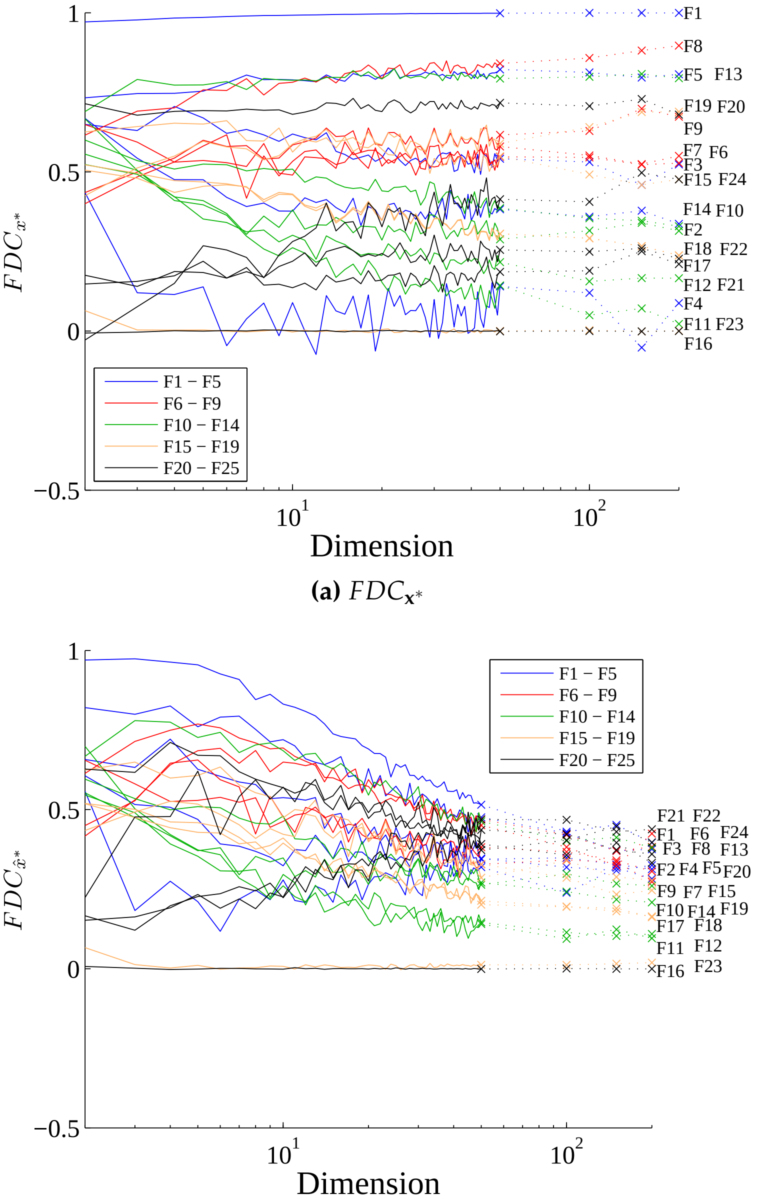


Figure 4.22: FDC using fixed-step (of size 10^{-2}) random walks for the Sphere and Reverse Sphere functions.

the logarithmic scaling of D). The range in $FDC_{\hat{\mathbf{x}}^*}$ coefficients for the problem set decreases from approximately [0,0.86] at $D = 1$ to [0,0.29] at $D = 200$. As p increases, the convergence curves change significantly. At $p = 2$, shown in Figure 4.17, the $FDC_{\hat{\mathbf{x}}^*}$ coefficients are quite constant from $D = 1$ to $D = 10$, and decrease sharply for $D > 10$. However, there is less of a change in the range in coefficients for the problem set; at $D = 1$, the problems' $FDC_{\hat{\mathbf{x}}^*}$ coefficients are in [0, 0.97], and at $D = 200$ they are in [0, 0.43]. Therefore, $FDC_{\hat{\mathbf{x}}^*}$ provides less discriminatory power as D increases. In summary, while the results in Aggarwal et al. [3] show that small (even fractional) values of p can slow the convergence, the use of a small p value in the L_p norm does not have a large effect on the convergence of $FDC_{\hat{\mathbf{x}}^*}$ in these experiments.

Modification 2: Fixed-step Random Walks

As shown in the dispersion case study (Section 4.4), fixed-step-size isotropic random walks may slow or prevent the convergence of distances. The following experiment investigates the effect of sampling via a fixed-step random walks on the FDC coefficients. For each D, 30 fixed-step random walks with $\alpha = 10^{-2}$ (Sphere and Reverse Sphere) and $\alpha = 1$ (BBOB) are used to calculate $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ for problem instances. Figures 4.22 and 4.23 show the resulting FDC coefficients averaged over the 30 samples at each value of D. The standard deviation for $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ on most functions and dimensions of the BBOB problems was generally very low (average of 0.0823 and 0.0955 respectively), and was at most 0.3966 and 0.3917 respectively.



(a) $FDC_{\mathbf{x}^*}$

(b) $FDC_{\hat{\mathbf{x}}^*}$

Figure 4.23: FDC using fixed-step (of size 1) random walks for the BBOB problem set.

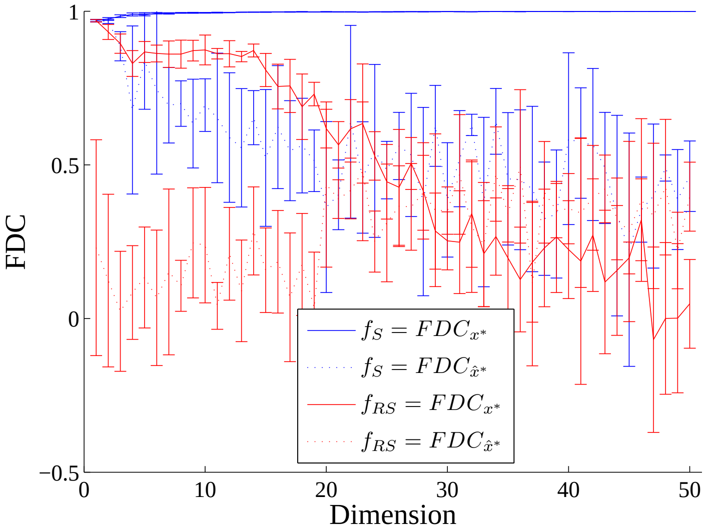


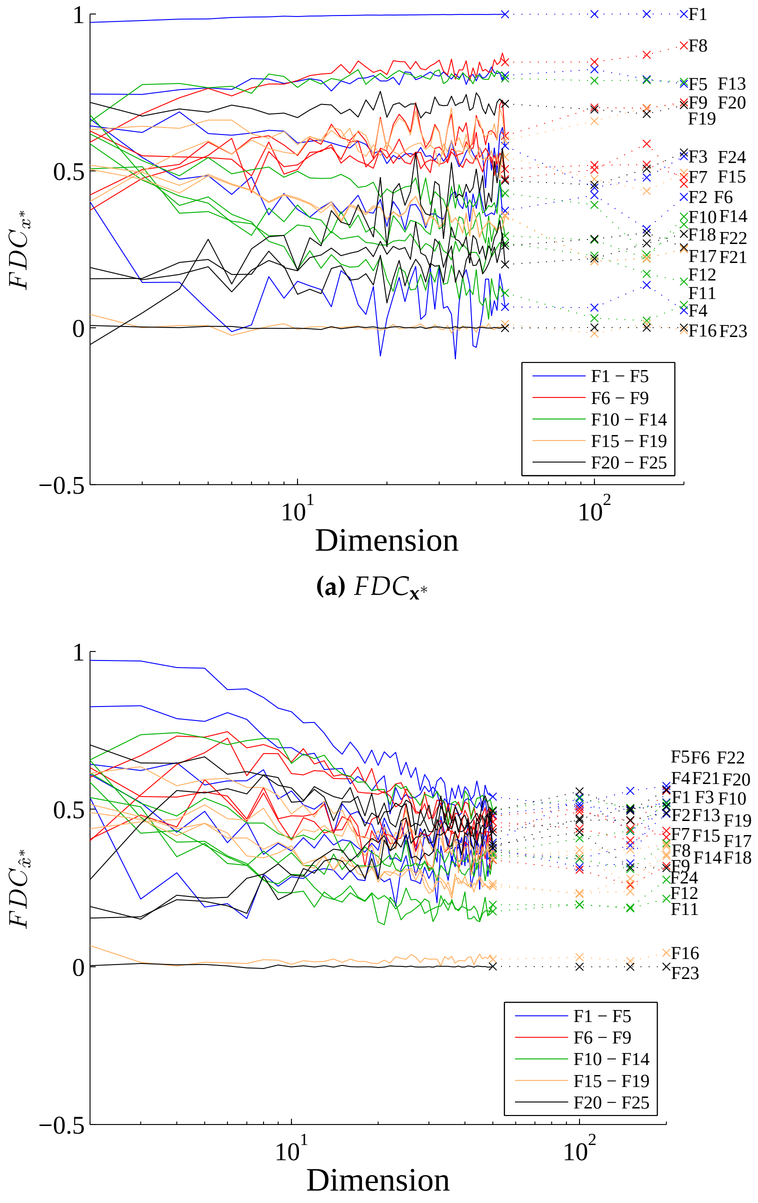
Figure 4.24: FDC using a Lévy random walk for the Sphere and Reverse Sphere functions.

The FDC coefficients in Figure 4.22 fluctuate considerably more between subsequent values of D than the FDC coefficients calculated using a uniform random sample (Figure 4.18). The standard deviations are also significantly larger than standard deviations of the FDC coefficients calculated using the uniform random sample. The FDC coefficients in Figure 4.23 also fluctuate considerably more between subsequent values of D than the FDC coefficients calculated using a uniform random sample (Figures 4.17 to 4.21). The FDC coefficients in Figure 4.23 show a similar trend to the FDC coefficients calculated using Euclidean distance (the L_2 norm) in Figure 4.17. Specifically, the $FDC_{\mathbf{x}^*}$ coefficients remain quite constant as D increases, while the $FDC_{\hat{\mathbf{x}}^*}$ coefficients range between [0, 0.97] at $D = 1$ to [0, 0.44]. To summarise, the FDC coefficients calculated from fixed-step random walks are comparable to the FDC coefficients resulting from a uniform random sample.

Modification 3: Lévy Random Walks

The following modification uses a Lévy random walk to obtain a sample of solutions from $\mathcal{S}$. At each value of D, 30 Lévy random walks with $\delta = 0$ and $\gamma = 0.1$ (Sphere and Reverse Sphere) and $\gamma = 0.1$ (BBOB) and are used to calculate $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$. The resulting FDC coefficients (averaged over the 30 samples at each D) are shown in Figures 4.24 and 4.25. The standard deviation for $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ for the BBOB problems for most functions and dimensions was generally very low (average of 0.1087 and 0.1212 respectively), and was at most 0.3961 and 0.3875 respectively.

The $FDC_{\mathbf{x}^*}$ coefficients shown in Figures 4.24 and 4.25a are very similar to the $FDC_{\mathbf{x}^*}$ coefficients resulting from the fixed-step random walks (see Figures 4.22 and 4.23). That is,



(a) $FDC_{\mathbf{x}^*}$

(b) $FDC_{\hat{\mathbf{x}}^*}$

Figure 4.25: FDC using a Lévy random walk for the BBOB problem set.

the coefficients fluctuate slightly, but are generally quite consistent as D increases. Modifications 1 (L_p norm with small p) and 2 (fixed-step random walks) yielded convergent $FDC_{\hat{\mathbf{x}}^*}$ coefficients, with little discriminatory power in high dimensions. The $FDC_{\hat{\mathbf{x}}^*}$ coefficients displayed in Figure 4.25b are less convergent than modifications 1 and 2. While the coefficients decrease between $D = 6$ and $D = 30$, the values remain quite constant (with minor fluctuations) for $D > 30$. In addition, the coefficients generally increase between $D = 150$ and $D = 200$. Overall, the Lévy random walks provides a wider variety of $FDC_{\hat{\mathbf{x}}^*}$ coefficients, that, in contrast to the other approaches, do not appear to converge as D increases.

4.5.2 Summary

The results in this case study show that using an explicit reference solution *not* taken from the sample, such as the global optimum ($\mathbf{x}^*$), yields FDC coefficients ($FDC_{\mathbf{x}^*}$) that are invariant to the dimensionality effects shown for dispersion in Section 4.4 and discussed in [3, 13]. In contrast, FDC coefficients ($FDC_{\hat{\mathbf{x}}^*}$) calculated by using a solution from the sample as the reference point resulted in decreasing/convergent values as dimensionality increased. The convergence of $FDC_{\hat{\mathbf{x}}^*}$ was particularly evident for uniform random samples in conjunction with the L_p norm. For this scheme, lower p values in the L_p norm resulted in poorer discrimination between the BBOB problems. Overall, the FDC coefficients calculated using Lévy random walks provided the best separability between the BBOB problems.

4.6 Summary

Sampling of high dimensional continuous optimization problems is an important but non-trivial task, and the resulting adequacy of the sample is highly influenced by the sampling methodology used, distance metric and sample size. This chapter reviewed and discussed the advantages and limitations of several well-established sampling methodologies, including systematic sampling, uniform random sampling, Latin Hypercube sampling, orthogonal sampling and random walks. The importance of sample size was discussed with respect to both theory and practice. Adverse effects from combining certain distance metrics with particular sampling methodologies was also theoretically and empirically analysed.

A major contribution of this chapter is the novel application of Lévy random walks as an effective sampling technique for high dimensional continuous problems. Experimental results on the Sphere, Reverse Sphere and BBOB problem set showed that the Lévy walk yielded dispersion and FDC estimates that, unlike uniform random sampling and fixed-step random walks, were not significantly affected by increasing dimensionality. Hence, given the comparable and often superior results to uniform random sampling and fixed-step walks (arguably the most widely used sampling techniques for continuous landscape analysis), Lévy random walks are used extensively in the experimental investigations in Chapters 7 and 8 of this thesis.

CHAPTER 5

Length Scales in Optimization

If you torture the data enough, nature will always confess.

Ronald Coase

A significant contribution of this thesis is to develop a framework that can be used to study the structural characteristics of a problem landscape, $\mathcal{L} = (\mathcal{S}, f, d)$, independent of any particular optimization algorithm. The following chapter proposes *length scale* as an important summary of optimization problem information. The chapter begins by defining length scale and describing several important properties. Small, intuitive problems are analysed using length scale to demonstrate how problem structure is captured and can be inferred. This chapter also introduces the *length scale distribution* as an important and useful summary of length scale information. Methods to estimate the length scale distribution are discussed, and properties of the distribution are outlined. The chapter concludes by reviewing concepts related to length scale.

5.1 Length Scale

Definition 5.1. *Let $\mathbf{x}^i, \mathbf{x}^j \in \mathcal{S}$ be two distinct ($\mathbf{x}^i \neq \mathbf{x}^j$) solutions with corresponding objective function values $f(\mathbf{x}^i)$ and $f(\mathbf{x}^j)$, and let $d : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$ be a distance function between solutions. The **length scale**, r, is defined as:*

$$r(\mathbf{x}^i, \mathbf{x}^j) = \frac{|f(\mathbf{x}^i) - f(\mathbf{x}^j)|}{d(\mathbf{x}^i, \mathbf{x}^j)} \tag{5.1.1}$$

The length scale intuitively measures the magnitude of objective function difference with respect to a step between two points in the search space[1]. The development of length scale

[1] The relationship between length scale and similar concepts in the optimization and wider literature, such as the *difference quotient* and *Lipschitz constant*, is discussed in Section 5.5.

is motivated by the analysis of continuous problems, however because r relies solely on the black-box information available from the problem landscape (i.e. $\mathcal{S}$, d and f), it is also applicable to discrete/combinatorial problems.

Contrary to the approaches previously used in developing landscape analysis techniques, length scale information makes no assumptions regarding the structure within landscapes, nor does it capture specific landscape properties. Rather, all information provided in the black-box setting (i.e. the ability to evaluate f at any point $\mathbf{x} \in S$) is captured and utilised, therefore implicitly capturing the landscape structure required to describe, characterise and differentiate problems.

Because the length scale is defined over distinct pairs of solutions, $d(\mathbf{x}^i, \mathbf{x}^j) > 0$, the denominator of Equation 5.1.1 can never be 0. However, no restrictions are made upon f, and so it is possible to obtain $r = \infty$ when $\left|f(\mathbf{x}^i) - f(\mathbf{x}^j)\right| = \infty$. Since length scale is defined as the change in magnitude of the objective function over a finite interval in the search space, its range is $[0, \infty]$.

5.1.1 Effect of the Distance Function

As has been discussed and demonstrated in Chapter 4, choosing a suitable distance metric is non-trivial and it can produce unpredictable and often surprising results. Generally, an appropriate distance metric preserves the geometric relationships between points, and a comprehensive survey of distances can be found in [44]. Because Euclidean distance is used in the vast majority of continuous optimization algorithms and analysis techniques, it is utilised throughout the analysis of continuous problems in this thesis. However precautions are taken with the sampling methodology to ensure that the distance between pairs of solutions within the sample does not converge in high dimensional problems (further details are deferred until Section 6.2 of Chapter 6). For the combinatorial problems analysed in this thesis, the distance is explicitly provided and is based on problem definitions, applications and analysis commonly adopted in the literature.

5.1.2 Analytical Length Scale Expressions

For situations where the objective function is known, it may be possible to derive an analytic expression for all length scales. Such expressions are amenable to analysis and inference regarding problem structure. The analytic expressions can also be used to obtain finite samples of length scales directly.

The following examples illustrate the derivation and utility of length scale expressions.

Example 5.1. *1-D linear objective function*

Given $f(x) = ax$ where $x, a \in \mathbb{R}$ and $d(x^i, x^j) = \left|x^i - x^j\right|$, the length scale between x^i and x^j is:

$$\begin{aligned} r &= \frac{\left|f(x^i) - f(x^j)\right|}{d(x^i, x^j)} \\ &= \frac{\left|ax^i - ax^j\right|}{\left|x^i - x^j\right|} \\ &= |a| \end{aligned}$$

For this problem, r captures the intuition that any step in $\mathcal{S}$ will be accompanied by a proportional change in f. This objective function is simple, but it is important to illustrate that the length scale of any finite set of samples from the search space (e.g. the points visited by an optimization algorithm) is invariant to the location(s) in $\mathcal{S}$ or the order in which the points were taken. The length scale of a D-dimensional neutral/flat landscape (i.e. $f(\mathbf{x}) = c, c \in \mathbb{R}$) is also a special case of a linear function where $r = 0$.

For most continuous problems, r will not be a constant over $\mathcal{S}$. In different regions of the space, the length scale value will depend on the local structure of the problem landscape from which $\mathbf{x}^i$ and $\mathbf{x}^j$ are drawn. Landscape structures such as varying slopes, basins of attractions, ridges and saddle points will affect the resulting length scale values. The 1-D absolute value function defined in Example 5.2 contains a single minimum at $x = 0$ that affects the resulting length scales.

Example 5.2. *1-D absolute value function*

Given $f(x) = a\,|x|$ where $x, a \in \mathbb{R}$ and $d(x^i, x^j) = \left|x^i - x^j\right|$, the length scale between x^i and x^j is:

$$\begin{aligned}
r &= \frac{\left|f(x^i) - f(x^j)\right|}{d(x^i, x^j)} \\
&= \frac{\left|a\left|x^i\right| - a\left|x^j\right|\right|}{\left|x^i - x^j\right|} \\
&= \begin{cases} a, & x^i, x^j < 0 \text{ or } x^i, x^j > 0 \\ a\frac{\left|\left|x^i\right| - \left|x^j\right|\right|}{\left|x^i - x^j\right|}, & x^i \geq 0, x^j < 0 \text{ or } x^i < 0, x^j \geq 0 \end{cases}
\end{aligned}$$

The length scale expression captures the linear nature of the function within each of the sub-domains $x \geq 0$ and $x \leq 0$, however steps *across* the minimum result in slightly more complex length scale values. Transitions directly across the minimum ($x^i = -x^j$) will result in a length scale of 0, and since $\left|x^i - x^j\right| \geq \left|\left|x^i\right| - \left|x^j\right|\right|$ ($x^i, x^j \in \mathbb{R}$), length scales across the minimum (where $x^i \neq -x^j$) will be within $(0, a]$.

The 1-D quadratic function offers slightly different structure to the 1-D absolute value function, but is also symmetric about a single minimum at $x = 0$.

Example 5.3. *1-D quadratic objective function*

Given $f(x) = ax^2$ where $x, a \in \mathbb{R}$ and $d(x^i, x^j) = \left|x^i - x^j\right|$, the length scale between x^i and x^j is:

$$\begin{aligned}
r &= \frac{\left|f(x^i) - f(x^j)\right|}{d(x^i, x^j)} \\
&= \frac{\left|ax^{i2} - ax^{j2}\right|}{\left|x^i - x^j\right|} \\
&= \frac{|a|\left|(x^i - x^j)(x^i + x^j)\right|}{\left|x^i - x^j\right|} \\
&= |a|\left|x^i + x^j\right|
\end{aligned}$$

Here, steps between points that are relatively close to the optimum result in relatively small length scales compared to the same-sized steps further from the optimum. This suggests that an algorithm needs to reduce the size of the steps it makes to successfully approach the optimum of this problem. Indeed, gradient descent algorithms are known as efficient and effective approaches for such problems because the gradient smoothly approaches 0 at the optimum.

The derivation of an analytical length scale expression is not always feasible; the length scale expression for certain problems may be too complex to reduce and/or interpret, while other problems may not have an accessible problem definition (e.g. a black-box problem). For these problems, a multiset of length scale values can be obtained by calculating r between pairs of solutions sampled from the search space. Insight into the structural nature of problems can be gained through statistical and information theoretical analysis of the length scale values obtained from a sample. In Chapter 6, a novel sampling methodology is proposed to obtain an adequate sample of r values, and new landscape analysis techniques are developed to characterise and distinguish problems based on the length scale samples.

5.2 Properties of Length Scale

From an information perspective, the structural regularities that fundamentally define a landscape are captured by distance, and are therefore invariant to isometric (distance preserving) mappings such as translation, rotation and reflection [20]. An *equivalence relation* [146] is defined (in terms of structure and information) between two problems if there is an isometric mapping between the search spaces as well as between the objective functions. For example, consider $f : \mathbb{R}^D \rightarrow \mathbb{R}$, and let $g(\mathbf{x}) = f(\mathbf{x} - \boldsymbol{\alpha}_1) + \alpha_2$ where $\mathbf{x}, \boldsymbol{\alpha}_1 \in \mathbb{R}^D$ and $\alpha_2 \in \mathbb{R}$. While both $\mathcal{S}$ and f have been translated (by $\boldsymbol{\alpha}_1$ and α_2 respectively) to define g, the structure of the landscape has not changed, and so from a landscape analysis perspective - and from the point of view of any reasonable optimization algorithm - f and g are equivalent (denoted $f \equiv g$). Because equivalent problems share the same structure, they should be characterised equivalently, and so it follows that problem characteristics with an invariance to isometric mappings are attractive. This is analogous to algorithm design, where algorithms are invariant to transformed, but equivalent problems (e.g. the invariance of Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [70]).

Length scale is the ratio of two distance functions, hence it is invariant to isometric mappings including translation, rotation and reflection. For uniform scaling of $\mathcal{S}$ by $\alpha \in \mathbb{R} \setminus \{0\}$, the length scale values are scaled by a factor of $\frac{1}{\alpha}$. Likewise, for scaling of f by α, the length scale values are scaled by α. Other transformations (e.g. shearing) have the potential to drastically alter the structure of the landscape, and length scale is sensitive to such transformations.

From Definition 5.1, if two length scales are identical, then the objective functions are equivalent between the pairs of solutions used to evaluate Equation 5.1.1. That is, given solution pairs $\mathcal{S}_a = \{\mathbf{x}^i, \mathbf{x}^j\}$ and $\mathcal{S}_b = \{\mathbf{x}^p, \mathbf{x}^q\}$, and functions $f_a : \mathcal{S}_a \to \mathbb{R}$ and $f_b : \mathcal{S}_b \to \mathbb{R}$, if:

$$\frac{|f_a(\mathbf{x}^i) - f_a(\mathbf{x}^j)|}{d(\mathbf{x}^i, \mathbf{x}^j)} = \frac{|f_b(\mathbf{x}^p) - f_b(\mathbf{x}^q)|}{d(\mathbf{x}^p, \mathbf{x}^q)} \tag{5.2.1}$$

then $f_a \equiv f_b$. Additionally, two nonequivalent functions will always produce non-equal length scales. Hence length scale is a very useful indicator of functional equivalence (over pairs of solutions) in the context of optimization. Assuming enumeration of all solutions (which is possible in the combinatorial case), two equivalent problems will yield identical length scales, and therefore, identical length scale distributions.

In the discrete/combinatorial context, the search space can be exhaustively enumerated and the length scale values calculated between all combinations of solution pairs. The resulting multisets of length scales are equivalent if and only if the problems are equivalent. Two equivalent functions will yield the same sets of length scales, while two nonequivalent functions will yield different sets of length scales. Thus, the length scale multiset completely and unambiguously describes and characterises a given landscape. In the continuous context, exhaustive enumeration is infeasible in practice, and so length scales are calculated from finite samples of the search space. Hence, two problems are likely to be equivalent if the multiset of length scales produced from the finite samples are equivalent. Problems with sub-regions of equivalence will have length scales in common, and so problem similarity can be measured by the degree to which problems share common length scales.

In practice, the length scale values calculated from a finite sample of solutions are highly dependent on the sample. For some problems, such as continuous or high-dimensional combinatorial problems, it is infeasible to sample all solutions in $\mathcal{S}$, and so it is possible to obtain two different length scale sets from two equivalent problems. This can occur when each of the samples captures a nonequivalent sub-space of the problem (i.e. the solutions and their respective objective function values differ), or when a different number of samples is taken for each problem. In addition, if the sub-space where two nonequivalent problems differ is not sampled, then it is also possible to obtain identical length scale sets for two nonequivalent problems. An intuitive example is comparing a constant (flat) objective function to a needle-in-a-haystack (NIAH) objective function; if the needle is not captured in the sample of the NIAH, then the two length scale sets (assuming equal size) will be identical. It is therefore imperative that the sampling methodology employed produces an adequate and representative sample of the problem. An adequate sample will yield length scales that are uniformly drawn from the true, underlying distribution of length scales. Further discussion is deferred until Section 6.2, where a novel sampling methodology and adequacy assessment criteria to obtain such a sample is outlined.

5.3 Length Scale Distribution

By considering r as a random variable, the length scales in a landscape can be summarised by their distribution. Continuous landscapes have an infinite number of solutions, and hence r is treated as a continuous random variable.

Definition 5.2. *Let r be a continuous random variable taking values from the set $R = [0, \infty)$ (i.e. $r \in R$). The **length scale distribution** is defined as the probability density function $p(r)$.*

To assist with readability, many of the following definitions and techniques are based on probability densities, rather than probability mass functions. In most cases, the definitions and techniques described have intuitive equivalents for probability mass functions, and hence are applicable for discrete/combinatorial problems. Precise definitions and formulae for the discrete/combinatorial case can be found in Appendix B.

The length scale distribution describes the probability of observing different length scale values for a given problem landscape. Consider Example 5.1 (1-D linear function) again. Since $r = a$, $p(r)$ is a Dirac delta function with a spike at $r = |a|$:

$$p(r) = \begin{cases} \infty \text{ if } r = |a| \\ \\ 0 \text{ otherwise} \end{cases} \tag{5.3.1}$$

where

$$\int_0^\infty p(r)\mathrm{d}r = 1 \tag{5.3.2}$$

It follows that the D-dimensional flat function also results in a Dirac delta function with a spike at $r = 0$. Figure 5.1 illustrates $p(r)$ for $f(x) = 2x$.

Now reconsider Example 5.3 (1-D quadratic function), where $r = |a| \left|x^i + x^j\right|$. Assuming uniform enumeration of $\mathcal{S}$, r is the absolute value of the sum of two independent, continuous uniform random variables (multiplied by the constant $|a|$). Introducing lower (b_l) and upper (b_u) bounds for f, let $Z = X + Y$, where $X, Y \sim \mathcal{U}[b_l, b_u]$ and are independent. The distribution for Z is triangular [66]:

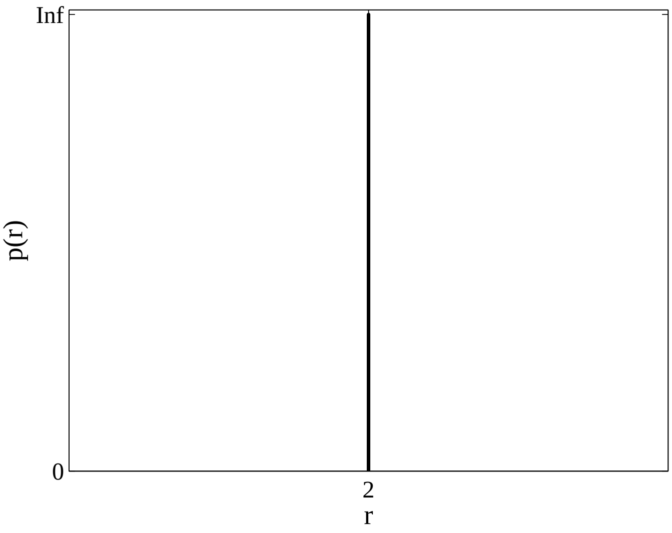


Figure 5.1: $p(r)$ for $f(x) = 2x$.

$$p_Z(z) = \begin{cases} \frac{z-2b_l}{(b_u-b_l)^2}, & 2b_l \leq z \leq (b_l + b_u) \\ \frac{2b_u-z}{(b_u-b_l)^2}, & (b_l + b_u) < z \leq 2b_u \\ 0 & \text{otherwise} \end{cases} \tag{5.3.3}$$

A density, p_s, can be produced by scaling another density, $p(x)$, by a factor of s via the relation $p_s(x) = \frac{1}{s}p(\frac{x}{s})$. Hence, the 1-D quadratic's length scale distribution is a "folded" triangular distribution:

$$\begin{aligned} p(r) &= \frac{1}{|a|}\left|p_Z\left(\frac{r}{|a|}\right)\right| \\ &= \frac{1}{|a|}\left(p_Z\left(\frac{r}{|a|}\right) + p_Z\left(\frac{-r}{|a|}\right)\right) \end{aligned}$$

Figure 5.2 illustrates $p(r)$ for $f(x) = 2x^2, x \in [-10, 10]$.

For simple problems where an analytical formulation of the problem is known, such as Examples 5.1 and 5.3, an expression for $p(r)$ can be derived. However, in most situations the analytical formulation of the problem is unknown or difficult to derive $p(r)$ from. In situations where analytical formulation is difficult or unknown, length scale values calculated from a sample of the landscape can be used to *estimate* $p(r)$. Probability density estimates are used in a wide variety of practical applications, and there exists many approaches and

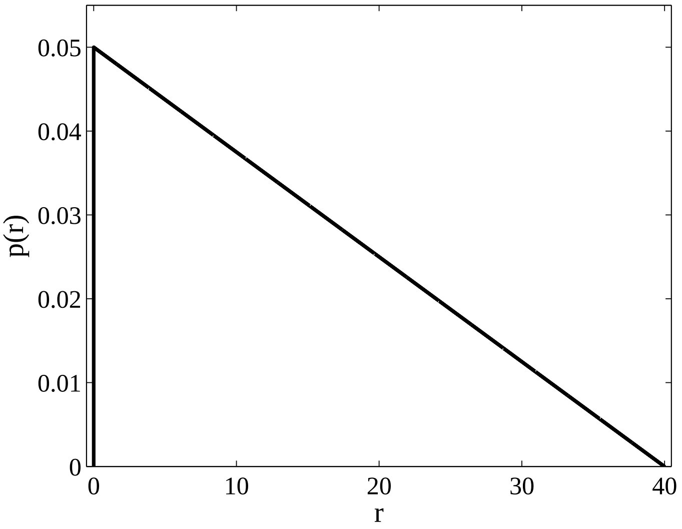


Figure 5.2: $p(r)$ for $f(x) = 2x^2, x \in [-10, 10]$.

techniques to derive accurate estimators. The length scale distribution, $p(r)$, is an important aspect of the framework presented in this thesis, and so the following section digresses in order to discuss and review relevant probability density estimation techniques.

5.3.1 Probability Density Estimation

In probability density estimation, various techniques and methods are used to estimate a probability density function for a given sample of data. The application of density estimation in this thesis is to estimate the length scale distribution from a univariate sample of length scale values. While numerous density estimation techniques exist for multivariate data (e.g. see [79, 150]), the following review focuses on density estimation of univariate data, $X = \left[x^1, x^2, \ldots, x^n\right]$. It is assumed that points in X are drawn independently from an unknown probability density, p, and the aim is to obtain an estimator, $\hat{p}$. There are three general approaches to density estimation; parametric, semi-parametric and non-parametric.

Parametric density estimation assumes that p is from a particular type or family of density functions, with corresponding parameters $\Theta = (\theta_1, \theta_2, \ldots, \theta_k)$. The aim is to choose the type of density and its corresponding parameters such that $\hat{p}$ models the data well. When little is known about the data, choosing a density family/type can be difficult, and so there exist a range of model selection techniques (e.g. Bayesian Information Criterion, minimum description length and structural risk minimisation) that can be employed to indicate the preferred density from a set of candidates [149]. However, these methods still require practitioners to specify a set of candidate models, from which the preferred model is chosen.

Maximum Likelihood Estimation or Bayesian Estimation are typically used to estimate the model parameters, Θ. In Maximum Likelihood Estimation, the parameters are assumed to be fixed and their values are chosen such that the probability of the data is maximised [79]. In contrast, Bayesian Estimators model (and hence estimate) the density's parameters with a probability distribution. Using this approach, a prior distribution, $p(\Theta)$, is assumed and a posterior distribution is estimated using Bayes' Theorem:

$$p(\Theta|X) = \frac{p(X|\Theta)p(\Theta)}{\int p(X|\Theta')p(\Theta')\mathrm{d}\Theta'} \tag{5.3.4}$$

Evaluation of $\int p(X|\Theta')p(\Theta')\mathrm{d}\Theta'$ can be difficult in practice for some families of distributions [4]. Parametric estimation typically models the data using a small number of parameters, and so it is advantageous in applications where storage of the density is an important consideration. However, specification of the data's parametric family can be very difficult and can lead to erroneous modelling when little or no information is known about the nature of the data.

A semi-parametric estimator models data using a combination of parametric and/or non-parametric components. Semi-parametric models are useful when there is no *single* density suitable for all of the data, rather, there are a number of densities, each suitable for a specific subset of the data. One popular semi-parametric estimator is the *mixture model*, which uses a weighted summation of component densities to produce a density estimate [4]. More specifically, given k component densities, $p(X|\mathcal{G}_i)$, such that $\sum_{i=1}^{k} p(\mathcal{G}_i) = 1$, the density estimate is:

$$p(X) = \sum_{i=1}^{k} p(X|\mathcal{G}_i)p(\mathcal{G}_i) \tag{5.3.5}$$

The component densities of the mixture model can be a mixture of parametric and non-parametric models.

In non-parametric density estimation, no explicit assumptions regarding the underlying distribution of X are made. Instead, there is an assumption that X is fundamentally "smooth", which essentially means that similar inputs are assumed to yield similar outputs [4]. This is quite a reasonable assumption, after all, inference and prediction is rather fruitless on noisy, random and erratic data. Non-parametric techniques are also known as *memory-based* or *lazy* algorithms, because their specification often requires the entire dataset, X [79]. In contrast, parametric models are simply defined by the density family and its corresponding parameters, which together are often much smaller than $|X|$. Despite their larger space requirements, non-parametric density estimators are advantageous in scenarios where little is known or can be assumed about the nature of X.

Because non-parametric density estimators do not assume an explicit distribution of the data, they are applicable to a wide variety of data. The work in this thesis utilises density estimation on univariate data only, and so techniques designed and better-suited to multivariate data are omitted from this review. One of the most well-known non-parametric density estimators is the *histogram*. Histograms discretise X into "bins" of specified width h, and the density is defined as [79]:

$$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} w\left(\frac{x - x^i}{h}\right) \tag{5.3.6}$$

where $x^i \in X$ and w is the weight function:

$$w(u) = \begin{cases} 1 & \text{if } |u| < 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{5.3.7}$$

Histograms are computationally very efficient to build and store, however they lack smoothness and may contain drastic fluctuations in probability between adjacent bins. Increasing h can improve the smoothness of the estimator, however this can also introduce more empty bins, and hence regions of 0 probability. The weight function acts as a harsh inclusion/exclusion operator; x's probability is based purely on points from X that are within a strict finite range (dictated by h) of x. To achieve smoothness, the range could be relaxed such that points very close to x contribute largely to its probability, and as the distance from x increases, the contributions gradually (and smoothly) decrease. This is the main idea behind *kernel density estimators* which use a *kernel* function, K, in-place of the weight function, w [79]:

$$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x^i}{h}\right) \tag{5.3.8}$$

where $h \in \mathbb{R}^+$ is known as the *bandwidth* and influences the smoothness of the estimate. For the purpose of density estimation, the kernel function, K, is non-negative and integrates to 1:

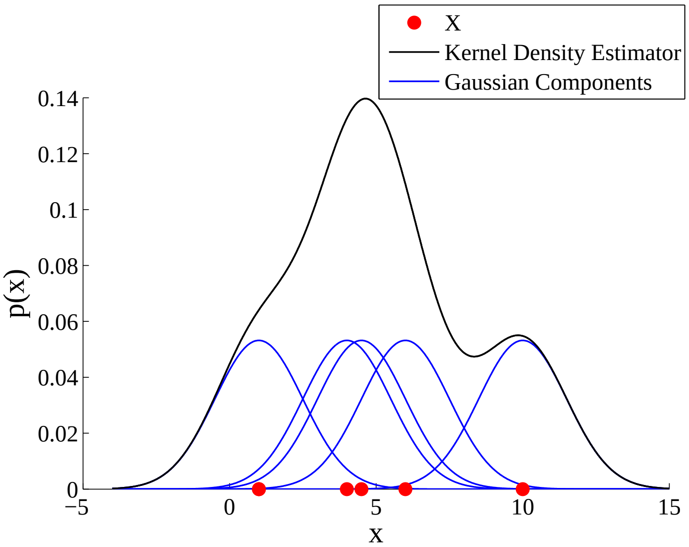


Figure 5.3: Kernel density estimator with a Gaussian kernel for $X = [1, 4, 4.5, 6, 10]$ and $h = 1.5$.

1. $\int_{-\infty}^{\infty} K(u) \mathrm{d}u = 1$

2. $K(u) \geq 0, \;\; \forall u \in \mathbb{R}$

K essentially dictates how the points nearby x influence its density value, and so it is often also symmetric about 0. A histogram is thus a kernel density estimator with a uniform distribution (with range equal to the bin width) as K. One of the most popular kernel functions used in practice is the Gaussian kernel [79]:

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp^{-\frac{1}{2}u^2} \tag{5.3.9}$$

Conceptually, kernel density estimation using a Gaussian kernel fits a (scaled) Gaussian component to each point in X (i.e. $\mu = x^i, \sigma = h$). The probability of x is then the sum of all the components' probabilities at x. Because Gaussian distributions have an infinite support, $\hat{p}$ also has infinite support. Figure 5.3 illustrates the contributions of the individual Gaussian components for $X = [1, 4, 4.5, 6, 10]$ and $h = 1.5$; nearby points largely contribute towards the density values, and the density is non-zero and generally quite smooth overall.

The computational running time of the length scale framework is dependent on the running time of the kernel density estimation. Thus in order to analyse the length scale framework's efficiency in later sections, the computational efficiency of kernel density estimation must first be reviewed. Computation of Equation 5.3.8 for a single evaluation point requires iterating over all n data points in X, and thus takes $O(n)$ time (assuming the evaluation of K takes $O(1)$ time). Hence, computing the density for a set of m evaluation points requires

$O(mn)$ time. Various modifications can be made to reduce the computational complexity. One commonly used approach reduces the running time to $O(mg)$ by "binning" the the points in X into a reduced set of g points that are equally spaced throughout the interval of X [135]. Further reductions in running time can be made using more sophisticated approaches, such as Fast Fourier Transforms and evaluating the density at only the binned points. While these can achieve significant improvements to speed, such modifications may not be suitable for certain applications and can affect the accuracy of $\hat{p}$.

The smoothness of $\hat{p}$ is highly influenced by the bandwidth, h. Large values of h correspond to a large overlap between kernels and thus yield a smooth density, while small values yield a sharper density that is more sensitive to fluctuations in the data. The selection of an appropriate bandwidth is a hard problem in itself, and there exist a wide range of well-established techniques that derive h from the input data and type of kernel function being used. Perhaps the simplest class of bandwidth estimation methods are the so-called "rule-of-thumb" methods [155]. In these methods, the bandwidth is derived from the bandwidth that is optimal for data originating from a particular type of distribution (e.g. Gaussian). While this does seem to counter the entire philosophy of non-parametric techniques (i.e. techniques that assume no particular form on the data), "rule-of-thumb" bandwidth selection can achieve satisfactory results when the data resembles some similarity to the assumed density. For example, if the data is unimodal, symmetric and thin tailed, then using a Gaussian kernel with a bandwidth that is optimal for a Gaussian will likely achieve satisfactory results. In practice, rule-of-thumb methods are generally avoided as the bandwidths chosen are sensitive to outliers and tend to over-smooth the data [79].

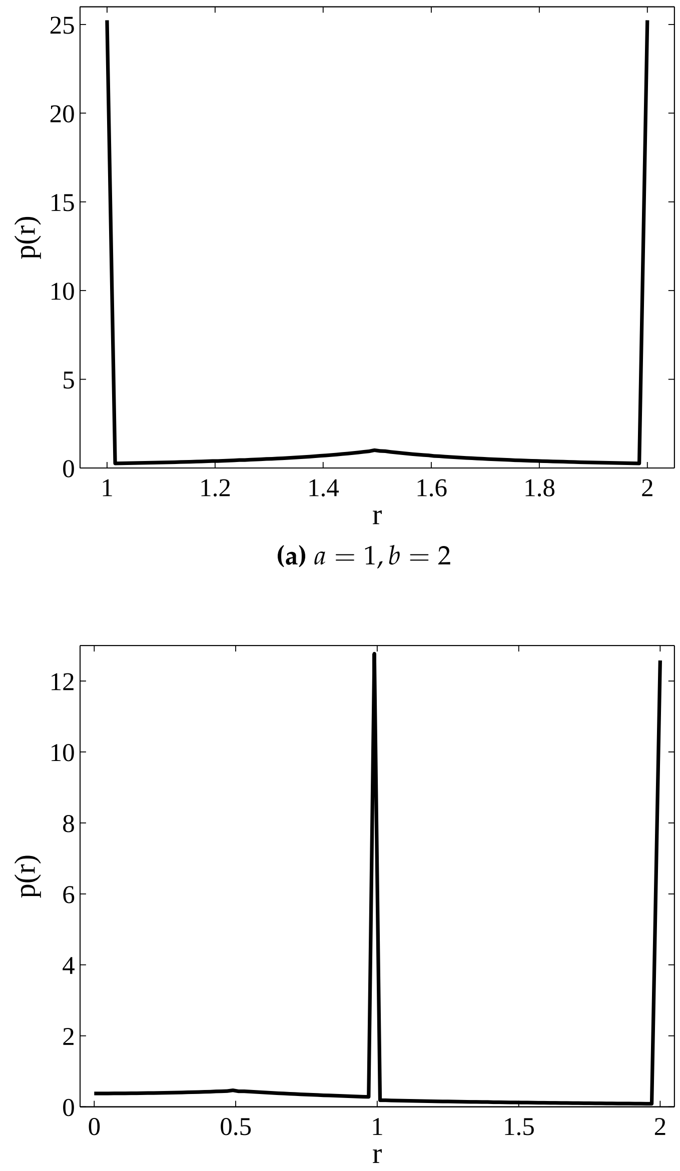
More accurate bandwidths can be derived by analysing and utilising the data itself, although such techniques are often computationally more costly than rule-of-thumb methods. In Leave-One-Out-Cross-Validation, n estimators are constructed by leaving out a single data point in turn. The bandwidth is then chosen such that the error (according to a particular criterion) between the n estimators is minimised [79]. Given that n different estimators are constructed, cross-validation methods are very computationally expensive and can be impractical for datasets where $n \gg 100$. An alternative class of bandwidth selection methods, known as "plug-in" methods, derive h via computations involving estimates of the density's *derivative functionals*. The estimation of the density derivative functionals are crucial to the success of these methods, and one of the most effective approaches is the "solve-the-equation" technique [79, 155]. Here, an initial bandwidth is substituted into a non-linear equation which is then solved and used to obtain an improved bandwidth estimate. This process is usually iterated multiple times, thus increasing the accuracy of h. The solve-the-equation plug-in method runs in $O(n^2)$, but for the specific task of estimating h for a univariate Gaussian kernel, $O(n)$ can be achieved using the *ϵ-exact* algorithm [135]. Experimental comparisons of the ϵ-exact algorithm and the original solve-the-equation method on synthetic and real-world data show that the ϵ-exact algorithm does indeed achieve $O(n)$ running time, with negligible differences in the accuracy of h.

Well-known non-parametric estimators that were not discussed above include nearest-neighbour estimators, variable-kernel estimators, adaptive kernel estimators, projection pursuit estimators, delta sequence estimators and orthogonal series estimators (see [78] for a review). While all are valid methods, kernel estimators are used throughout this thesis as they are particularly appropriate for length scale data, which is univariate with little or no information known regarding its underlying distribution. In addition to its appropriateness for length scale data, kernel density estimation is intuitive, well-established in the literature, widely used in practice and there are existing tools and algorithms to improve performance with negligible detriment to the estimator's accuracy. While kernel estimators are used for the majority of density estimation in this thesis, for small, illustrative examples where the accuracy of $\hat{p}$ is not imperative, histograms are used instead.

5.4 Properties of the Length Scale Distribution

The length scale is defined as the change in objective function value with respect to a step in the search space. For unknown f (i.e. black-box), $r \in [0, \infty]$, and so in general, $p(r)$ is supported over the semi-infinite interval $[0, \infty]$. This interval can potentially be restricted if additional function and domain information is known.

The majority of the length scale distributions of continuous problems in this thesis tend to be unimodal, long tailed distributions. While this phenomenon is observed throughout the continuous problems analysed in this thesis, it is certainly not true across all continuous problems. To illustrate this point, the following investigation contrives problems to produce multimodal length scale distributions. This is achieved by combining multiple 1-D linear functions, each defined at a separate partition of the domain. Individually, each function's corresponding $p(r)$ has a "spike" positioned at $|a|$, and so by combining functions with different values of a, a multimodal $p(r)$ can be obtained. For example, consider a 1-D piecewise function consisting of two linear functions:



(a) $a = 1, b = 2$

(b) $a = -1, b = 2$

Figure 5.4: Examples of multimodal length scale distributions.

$$f(x) = \begin{cases} ax, & x \leq 0 \\ bx, & x > 0 \end{cases} \tag{5.4.1}$$

where $a \neq b$. The subsequent length scale distribution has two modes positioned at a and b respectively. Figure 5.4 shows two realisations of Equation 5.4.1 where $\mathcal{S} = [-10, 10]$. Solutions were enumerated from $\mathcal{S}$ by increments of 0.01, the length scales between all solutions were calculated and $p(r)$ was estimated using a histogram with 100 bins.

This concept can be extended to produce piecewise functions consisting of n linear components. In the following, $\mathcal{S} = [0, 10]$ is partitioned into n equally sized pieces. The domain

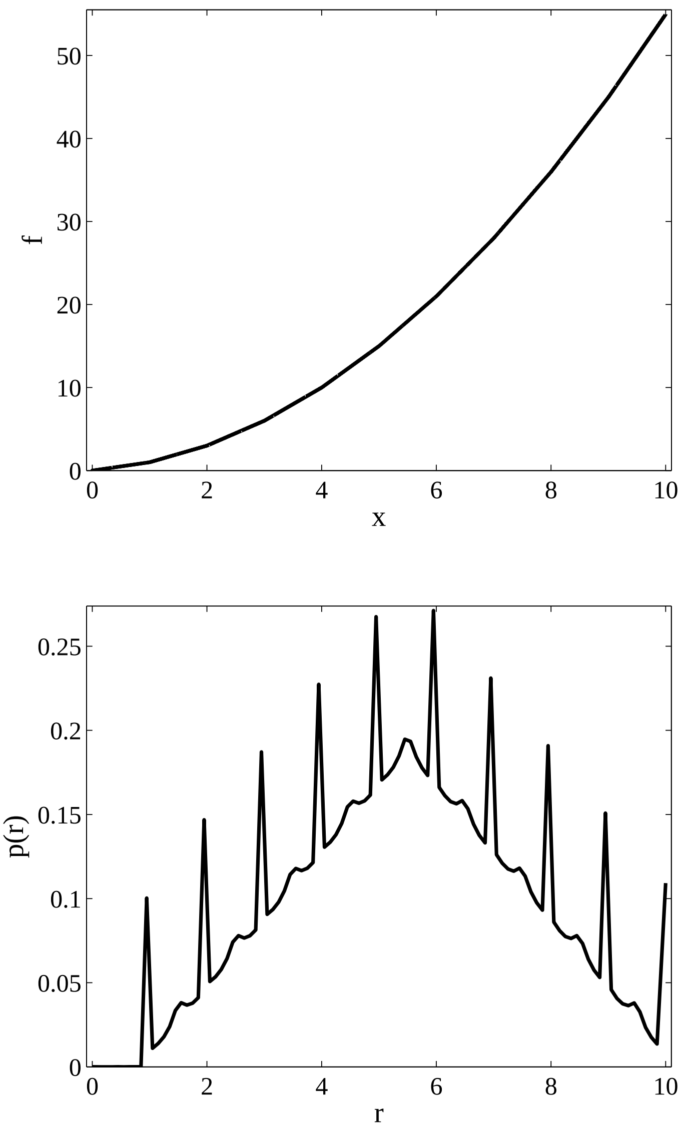


Figure 5.5: $n = 10$ with sequentially assigned pieces.

of piece i is $[\frac{10(i-1)}{n}, \frac{10i}{n}]$ and $f_i(x) = ix$. Figures 5.5 to 5.7 displays example functions and length scale distributions for the piecewise function where $n = 10, 30$ and 100.

For the sequentially-assigned functions (Figures 5.5b to 5.7b), increasing the number of pieces, n, initially increases the number of modes. However, as the number of modes increases, the length scales begin to reduce the distinction between modes. Eventually, a threshold is reached (here, at $n \approx 100$) where the modes are no longer distinguishable, and instead, the distribution resembles a triangular distribution. Indeed as n continues to increase, the distribution smoothly tends towards a triangular distribution with lower and upper bounds 0 and n respectively and single mode at $\frac{n}{2}$. This is actually quite intuitive; as n increases, f more closely approximates a quadratic function.

The examples above show that by assigning the piece functions sequentially to the do-

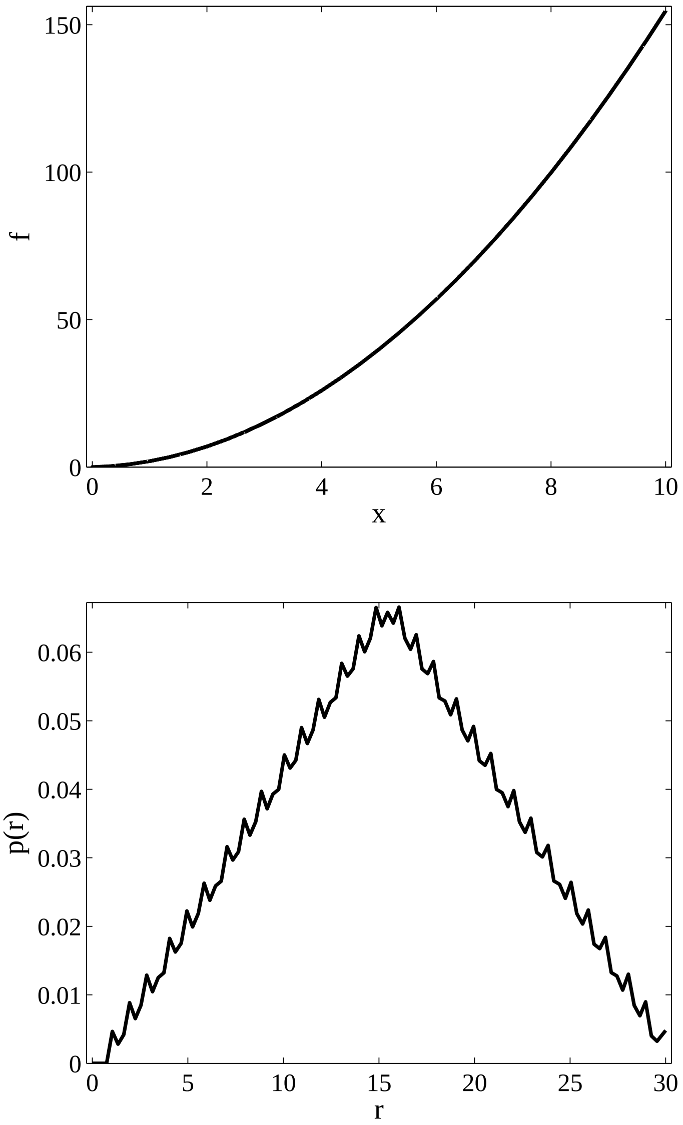


Figure 5.6: $n = 30$ with sequentially assigned pieces.

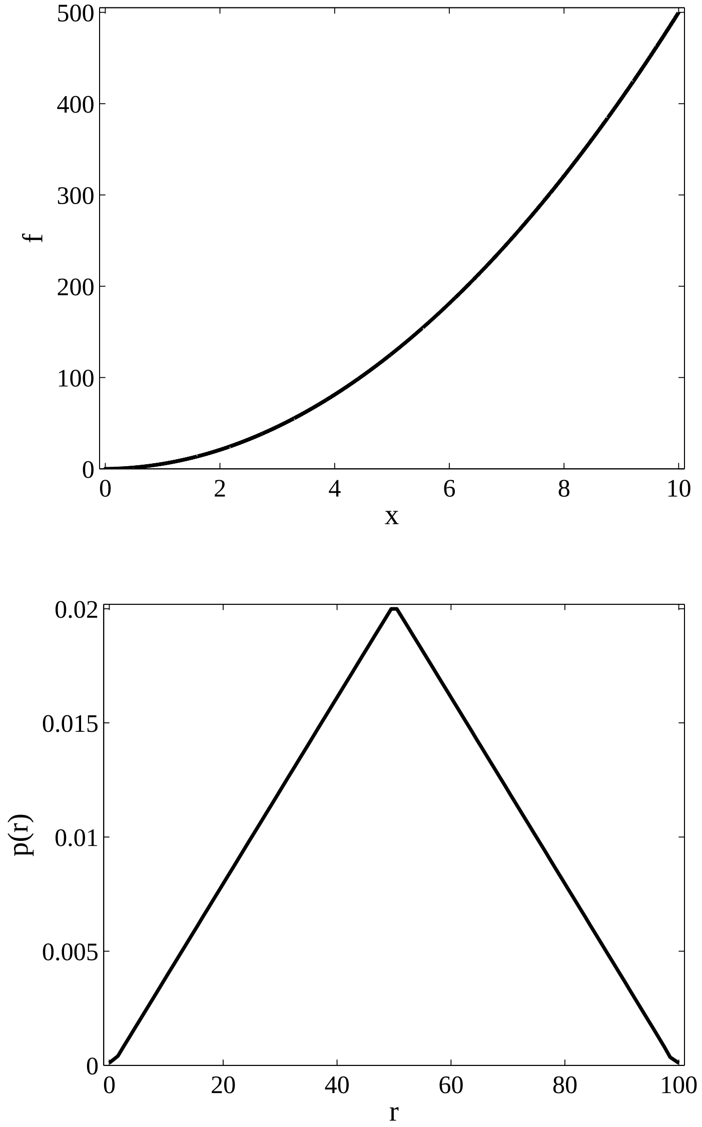


Figure 5.7: $n = 100$ with sequentially assigned pieces.

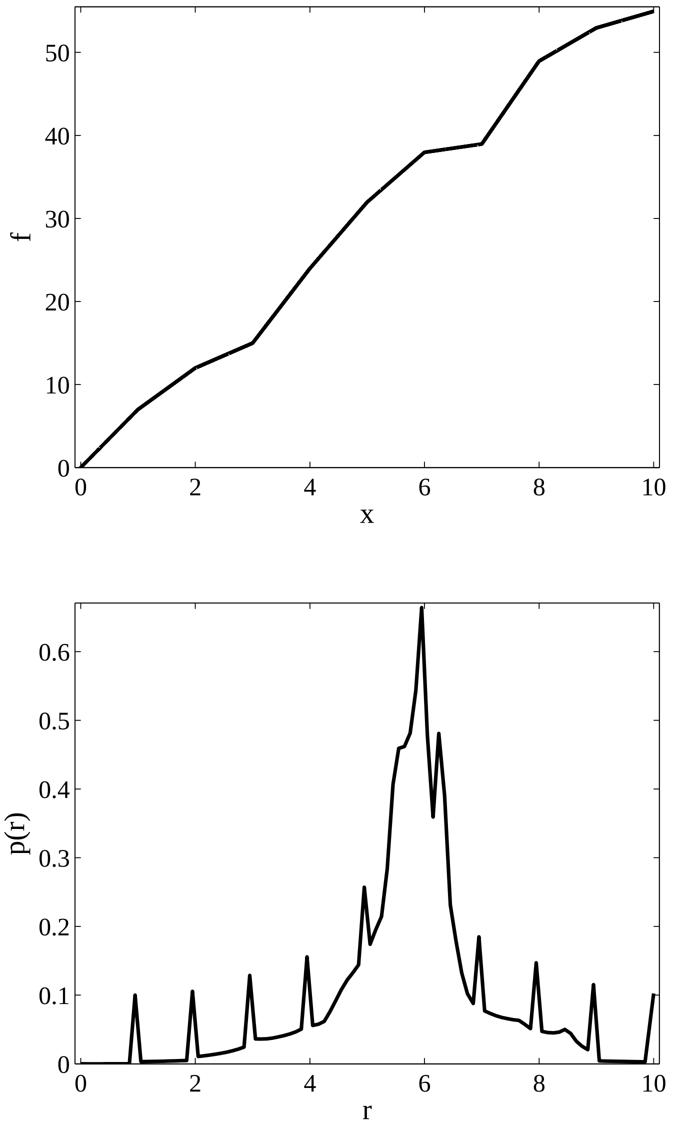


Figure 5.8: $n = 10$ with uniform randomly assigned pieces.

main, multimodal length scale distributions up to a certain number of modes can be produced. Next, the n linear piece functions are assigned *uniform randomly* along the domain. That is, n different linear pieces are produced ($f_i(x) = ix$) and each function is assigned a (uniform) random partition of the domain. As in the above examples, the pieces are combined such that the functional transitions between domain partitions has no discontinuities. Figures 5.8 to 5.10 shows the results of the piecewise functions and length scale distributions for these problems.

The randomly-assigned piecewise functions exhibit similar (yet more erratic) behaviour to the sequentially-assigned piecewise functions. Specifically, as n increases, the modes become less pronounced and are rendered indistinguishable by the remaining length scales. However, the randomly-assigned piecewise functions do not appear to produce triangular

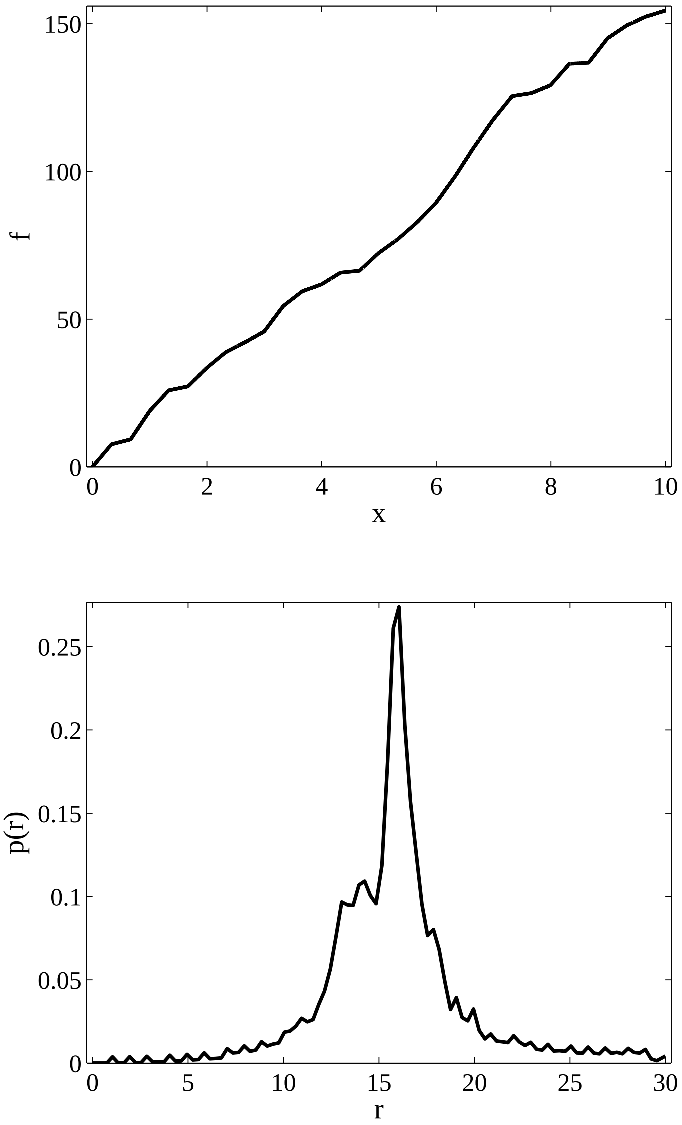


Figure 5.9: $n = 30$ with uniform randomly assigned pieces.

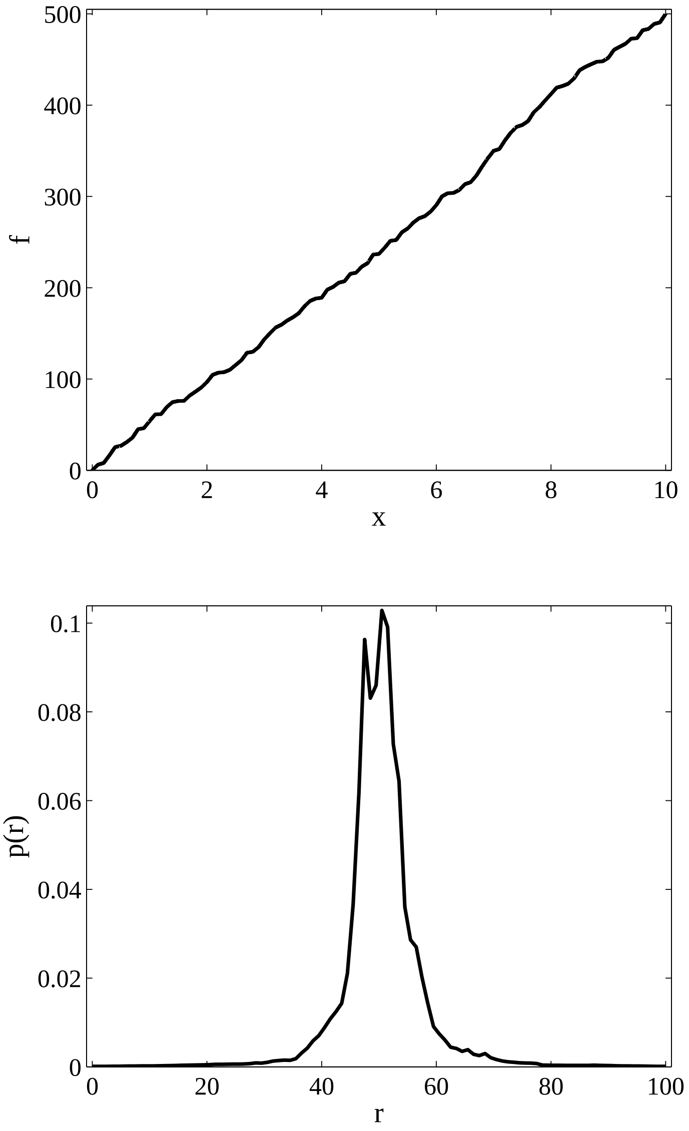


Figure 5.10: $n = 100$ with uniform randomly assigned pieces.

length scale distributions as n increases. Instead, the $p(r)$s tend towards a unimodal symmetric distribution with shrinking variance as n increases. As previously shown in Example 5.1, a 1-D linear function results in a $p(r)$ with a single mode positioned at the gradient of the linear function. The results in Figure 5.10 reflects the similarity of the piecewise function to the 1-D function; as n increases, f appears increasingly more similar to a 1-D linear function with small perturbations, and the resulting $p(r)$ in Figure 5.10b has a single, narrow mode.

5.5 Related Work

Length scale is related to a number of other existing concepts in the optimization and wider literature. While the length scale value in Equation 5.1.1 bears similarity to other concepts discussed below, it is fundamentally and uniquely concerned with capturing relative changes in the objective function at a wide variety of solution intervals, particularly for the purpose of analysing and understanding optimization problems.

The definition of r is related to the *difference quotient* (also known as *Newton's quotient* and is a generalisation of *finite difference* techniques) from calculus and numerical analysis. The difference quotient is defined as $\frac{f(x+h)-f(x)}{h}$ and used to estimate the derivative at x, as $h \to 0$ [125]. Numerous applications, including the implementation of gradient-based algorithms, utilise approximations of this form when the gradient of f is not available. Finite difference methods are widely used in the solution of differential equations, but are not typically used in the context of landscape analysis.

Length scale should not be confused with the derivative of a function. Fundamentally, length scale aims to capture the relative change in objective function value between solutions at a wide variety of distance intervals. In contrast, the derivative is defined at a single point, and while approximations (like the difference quotient) typically utilise pairs of points, the points are infinitesimally close to each other. Crucially, r is defined for problems where no notion of a derivative exists, such as combinatorial problems, black-box continuous problems and non-differentiable continuous problems.

Length scale is also related to the *Lipschitz constant*, defined as a constant, $L \geq 0$, where the Lipschitz condition $\left|f(\mathbf{x}^i) - f(\mathbf{x}^j)\right| \leq L\|\mathbf{x}^i - \mathbf{x}^j\|, \forall \mathbf{x}^i, \mathbf{x}^j$ is satisfied [152]. The ideal Lipschitz constant is the smallest L for which the Lipschitz condition holds, and functions satisfying the condition are known as Lipschitz continuous.

The definition of the Lipschitz constant is very similar to length scale, more specifically, the maximum length scale of a problem is equal to the ideal Lipschitz constant. However, the differences in the definitions of r and L define vastly different values; valid Lipschitz constants may overestimate the largest rate of change, while length scales will always underestimate or be equal to the largest rate of change. Consider again Example 5.3: the 1-D quadratic objective function (defined in Section 5.1). The length scales of this function are given by $|a|\left|x^i + x^j\right|$. For convenience, bounds $x \in [0, 1]$ are introduced $a = 1$ (hence $f = x^2$). Given that $x^i, x^j \in [0, 1]$, valid r values are in $(0, 2)$. For bounds $[0, 1]$, Lipschitz constants in $[1, \infty)$ satisfy the Lipschitz condition. Hence, values for r and L are very different for this simple function.

Lipschitzian optimization **algorithms** are a class of global optimization algorithms that use knowledge of Lipschitz constants to solve Lipschitz continuous problems [75]. A function's Lipschitz constant is not always known a priori (e.g. in the black-box scenario), and so various methods have been developed in order to estimate L from the landscape. Estimating L is itself a global optimization problem, and so heuristics are used to actively search the landscape for L. Heuristics vary between methods, with many calculating what are essentially length scales between solution pairs, and refining the search to solutions with large/promising length scales [10, 178, 204]. Because the aim is to accurately estimate L, only the supremum of the length scales along the trajectory is of interest and hence retained. The remaining sample of length scales are of no use in the Lipschitzian optimization context, and so they are never analysed and simply discarded. Consequently, notions and concepts similar to length scale analysis are absent from the Lipschitzian optimization literature. While the techniques used to estimate L illustrate potential approaches for sampling r, the sample is biased due to the heuristics used to search for L. Therefore, the resulting length scales do not present an accurate representation of the landscape and are of little use from a landscape analysis perspective. In summary, the Lipschitzian optimization community utilise length scales, however there is no evidence that the length scales have been analysed, and the methodologies used to generate samples of r are not suitable for landscape analysis.

To summarise, length scale analysis captures information regarding *all* rates of change, over a wide variety of intervals (distances) on the problem. While the concept of length scale is similar to the difference quotient and the Lipschitz constant, to the best of knowledge, the utilization of this information at *all* scales has not been previously performed in the context of optimization problem analysis.

5.6 Summary

Length scale, r, has been proposed as a fundamental property of optimization problems. In practice, length scale values can be calculated from a finite sample of candidate solutions and their objective functions values. The simplicity of length scale means that it can be readily calculated from samples of the landscape and/or algorithm search data. Length scale is invariant to isometric mappings including translation, rotation and reflection, and it is sensitive to scaling and shearing. Common r values indicates functional equivalence between the pairs of solutions over which the r values are calculated. Likewise, nonequivalent functions will yield different length scale values. Hence, r is an indicator for functional equivalence over pairs of solutions.

The length scale distribution, $p(r)$, is an important summary of length scale information. When the analytic form of f is known, an analytic expression for $p(r)$ may be derived. The expression for $p(r)$ facilitates analysis of landscape structure through direct interpretation and/or sampling. In practice, f is often unknown or difficult to derive an expression for $p(r)$. In this scenario, a finite multiset of r values can be obtained from sampling the problem landscape, and $p(r)$ can be estimated using a probability density estimator such as a kernel estimator or histogram. While the majority of length scale distributions analysed in this thesis are unimodal, long-tailed distributions, it has been shown that multimodal distributions are possible.

CHAPTER 6

Length Scale Analysis: Techniques and Practical Considerations

But the truth is, it's not the idea, it's never the idea, it's always what you do with it.

Neil Gaiman

The length scale proposed in Chapter 5 encapsulates the structural information in problem landscapes. This chapter proposes a new method to sample length scale values in practice, and techniques to subsequently analyse the length scale information. Section 6.1 reviews selected techniques from the visualisation, set-theory, clustering, statistics and machine learning literature, and unique procedures are developed to apply relevant techniques to length scale data. In Section 6.2 a novel methodology is proposed to obtain an adequate sample of length scales in practice. Important practical considerations, including the time and space complexities of the methods, are discussed in Section 6.3. The chapter concludes in Section 6.4 with a summary of its contributions.

6.1 Analysing Length Scales

For a given optimization problem, the length scales are a multiset of scalar values that contain important structural information. As argued in Section 5.2, the exhaustive multiset of length scales completely describes and hence identifies a problem. This section investigates the utility of length scale information by analysing the length scales to infer structural features of a problem landscape, as well as using the length scales to compare the structural similarity between problems. While this section mainly outlines relevant analysis techniques, experiments in Chapter 7 utilise the techniques developed in order to analyse sets of artificial, benchmark and real-world-like problems from both continuous and combinatorial optimization.

6.1.1 Heatmaps of Length Scale Values

Heatmaps are a commonly used technique to visualise the data from a two dimensional matrix. Typically, such matrices occur when a measurement is taken between the entities of two sets. To build a heatmap, the first set is enumerated (according to some order) along the x-axis, the second set is likewise enumerated along the y-axis, and the grid-points at coordinates (p, q) are coloured according to the measurement between the entities that p and q represent. Thus, for 1-D problems, the length scale values between explicit steps in the search space can be viewed by using a heatmap, where the x-axis corresponds to the start of the step, x^i, the y-axis corresponds to end of the step, x^j, and the colour of pixels at the coordinate (x^i, x^j) reflects the length scale value. This technique involves sampling the search space evenly within a bounded region. The constraints defined for constrained problems make suitable bounds, however bounds must be introduced for unconstrained problems.

To demonstrate the information afforded by a heatmap, consider Example 5.3 (the 1-D quadratic function). In this example, consider $f(x) = 2x^2, x \in [-10, 10]$ with x sampled incrementally: $S' = \{-10, -9.999, \ldots, 10\}$. Figure 6.1 shows the resulting heatmap of length scales between all 20001 solutions in S'. Pixels are coloured according to their value in the range of the length scales sampled, where black represents low values, red represents intermediate values and yellow/white represents high values.

Length scale is symmetric by definition, and so heatmaps will be symmetric across the diagonal drawn from the upper left corner and the lower right corner. The length scale value between a solution and itself is not defined, and so the leading diagonal (indicated by the blue line) in the heatmap is also not defined.

The heatmap in Figure 6.1 shows how length scales capture the symmetric structure of the quadratic function. Steps directly across the minimum (i.e. from x to $-x$ and vice versa) result in no change in f, and hence $r = 0$ (coloured in black). For this function, the largest length scales are at the edges of the search space (e.g. between 9.9 and 10), where the landscape is the steepest. Because of the precision used to enumerate the function, this is a step between 9.999 and 10 (or -9.999 and -10), which results in $r = 39.998$. Note that as the precision of the step size approaches 0, r asymptotically approaches the gradient, which is

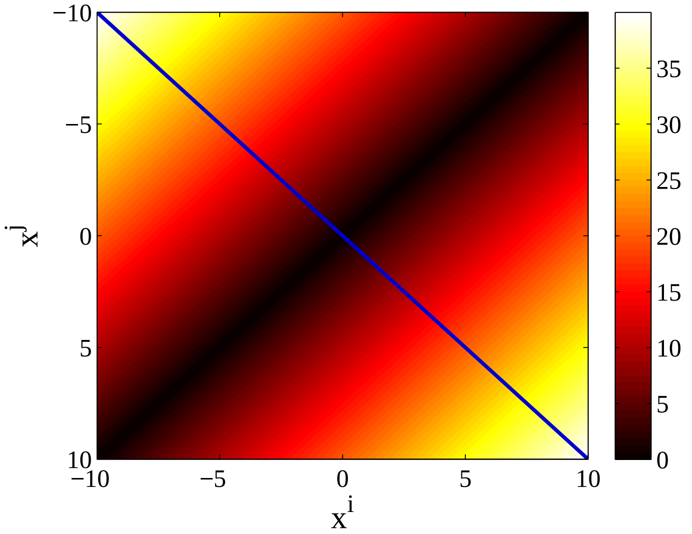


Figure 6.1: Heatmap of r for $f(x) = 2x^2, x \in [-10, 10]$. The blue line indicates where $x^i = x^j$, and hence r is undefined.

40 for the step on the edge of the quadratic. The heatmap shows that similarly sized steps towards the middle of the landscape (e.g. between 0 and 0.001) results in small length scales. Saddle points, and hence potential optima, can be observed when a length scale value of 0 occurs. For this problem, there is a clear optimum represented by the dark region. Overall, the change in length scales is very smooth, and reflects the smooth nature of the quadratic landscape.

To illustrate the richness of length scale information provided by heatmaps, the following artificial 1-D function is used and contains a variety of different topological features, including neutrality, linear slopes, convex basins of attraction and funnels.

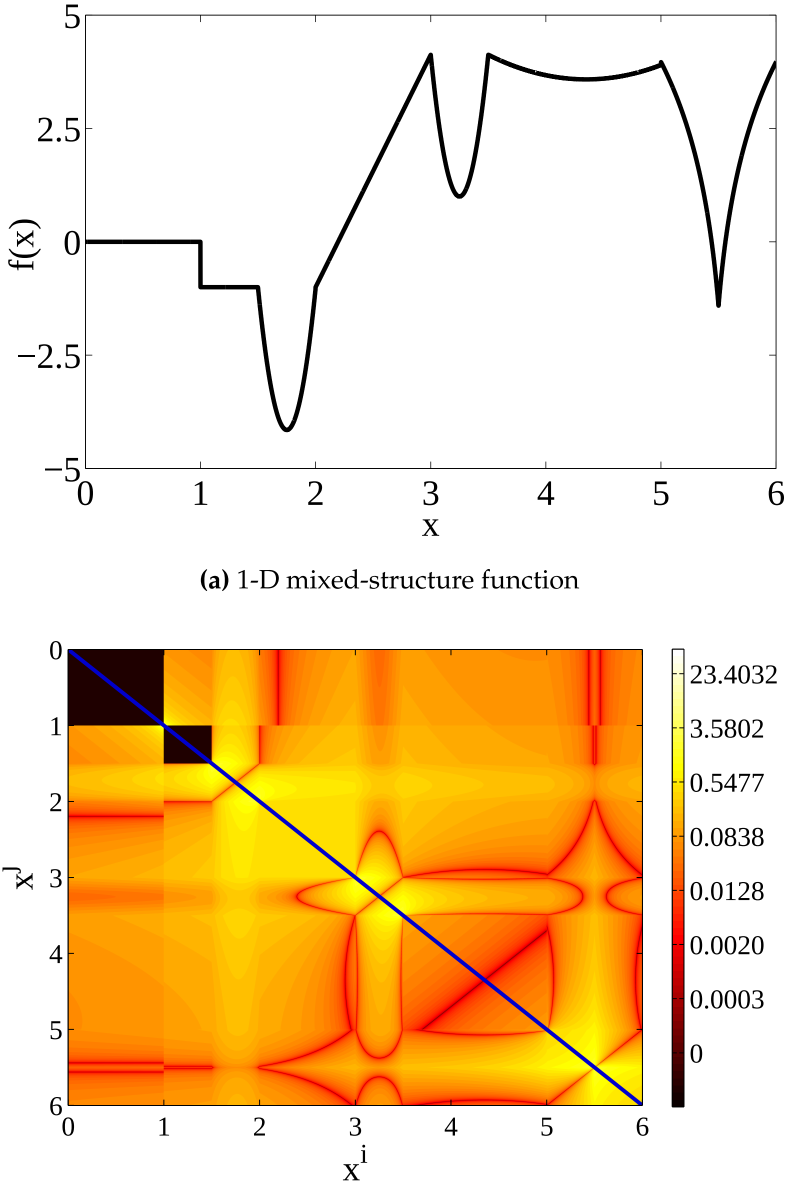
Example 6.1. *1-D "mixed-structure" function defined as follows and shown in Figure 6.2a.*

$$f(x) = \begin{cases} -1, & 1 \leq x < 1.5 \\ 50(x-1.75)^2 - 4.15, & 1.5 \leq x < 2 \\ 5.125x - 11.25, & 2 \leq x < 3 \\ 50(x-3.25)^2 + 1, & 3 \leq x < 3.5 \qquad \text{where } x \in [0,6] \\ 0.75(x-4.35)^2 + 3.583, & 3.5 \leq x < 5 \\ 3\log(|x-5.6|) + 5.5, & 5 \leq x < 5.5 \\ 3\log(|x-5.4|) + 5.5, & 5.5 \leq x \leq 6 \\ 0 & \text{otherwise} \end{cases}$$

Figure 6.2b shows the length scales calculated between pairs of points, x^i, x^j, at increments of 10^{-3} across the search space, $\mathcal{S} = [0,6]$. The values have been shaded using a logarithmic scale to better visualise magnitudes of change in r. The blue line across the leading diagonal indicates undefined length scales, where $x^i = x^j$.

It is clear from the length scales in Figure 6.2b that there are many different structures within the landscape. While it may not be immediately obvious what the structures are, the boundaries between them can be identified by the sudden transitions in shade/colour and pattern. For example, the colour change from black to light yellow at $x^i \approx x^j \approx 1.5$ on the diagonal of the figure indicates a large change in structure. Here, the change in structure is caused by the transition between the flat/neutral region ($1 \leq x < 1.5$) and the quadratic region ($1.5 \leq x < 2$). Likewise, the change in colouring at $x^i \approx x^j \approx 5$ shows a change in structure, caused by the transition between the quadratic region at $3.5 \leq x < 5$ and the funnel region at $5 \leq x \leq 6$.

As the heatmap is symmetric, the nature of the structures within the landscape can be further understood by analysing the lower (or upper) triangle in Figure 6.2b. Darker shades represent small changes in objective function values between the candidate solutions x^i and x^j, while lighter shades indicate large jumps in f. Solid blocks of colour, such as between $[0,1]$, signal a constant change in objective function when the solutions are drawn from a region. The dark lines and curves show steps in the space where $f(x^i) \approx f(x^j)$, e.g. moving



(a) 1-D mixed-structure function

(b) Enumeration of r in the 1-D mixed-structure function

Figure 6.2: 1-D mixed-structure function enumerated at increments of 10^{-3}.

from a point on one side of a basin or funnel to a point on the other side of the minimum with equal objective function value. Approximate locations of optima, such as $x = 3.25$, may be located by observing dark lines where the area on either side of the line changes to lighter shades.

The visualisation in Figure 6.2b also shows how simple structures, such as convex and concave basins of attraction, can combine to give a complex objective function from the viewpoint of an algorithm that only has solutions (and their respective f-values) sampled from the landscape. Consider steps within $[3, 6]$; the change in objective function values vary significantly in a complex, unpredictable manner. This is the only type of information that black-box optimization algorithms can use when solving a given problem.

Direct visualisation of length scales using heatmaps is limited for multidimensional problems. In these situations it may be possible to view the length scale heatmap of particular dimensions, or of the search space resulting from dimensionality reduction [93].

6.1.2 Length Scale Set

The length scale values sampled from a landscape form a multiset, and this can be viewed using simple graphing techniques such as plotting the sorted set, scatterplots and box-and-whisker plots. Figure 6.3 shows the sorted length scale values for the 1-D quadratic (Example 5.3 where $a = 2$) and mixed-structure function (Example 6.1). The length scales used here are the same set used to produce the heatmaps in Figures 6.1 and 6.2b. By positioning the r values along a common domain ($[0, 1]$ in Figure 6.3), comparisons between length scale sets can be made.

Figure 6.3 shows that the mixed-structure function has a wider variety of length scale values than the quadratic. In particular, the mixed-structure function has length scales over 10 orders of magnitude smaller and 3 orders larger than the quadratic's length scales. Note that due to the logarithmic scaling of the y-axis, length scales of 0 are absent from the graph (the mixed-structure function's curve begins after the 1-D quadratic's because the mixed-structure has more length scales of 0). The quadratic's curve indicates that the length scale set is predominately made up of length scales in $[0, 10]$, with few values outside this range. Indeed, both curves show that the majority of length scales fall between a relatively very small range, and that the extreme length scale values within each set make up only a small proportion of the set.

There exist many other approaches to measure and quantify the similarity between sets.

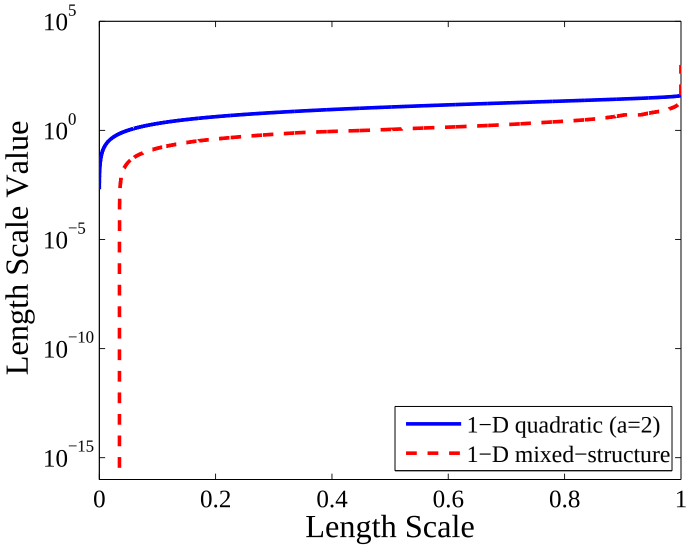


Figure 6.3: Sorted length scale multisets for the 1-D quadratic (where $a = 2$) and 1-D mixed-structure functions.

Arguably the most simplest and well-known measure is the *Jaccard index*, which given two sets A and B, is defined as the proportion of common/intersecting elements to the union of A and B [94]. Mathematically:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{6.1.1}$$

Thus, $0 \leq J \leq 1$, and $J(A, B) = 1$ if $A = B$, while $J(A, B) = 0$ if A and B have no elements in common. The Jaccard index is 0.0023 for the samples of length scales from the 1-D quadratic (where $a = 2$) and mixed-structure function shown in Figure 6.3.

Related set similarity measures that are also based on manipulations of the set unions and intersections include the *Tanimoto similarity* and *Sørensen similarity*. A plethora of other similarity measures can also be found in the clustering literature [196]. A solution to a clustering problem is simply an assignment of each data point into a particular cluster. Hence, each cluster can be thought of as a set of data points, with the constraint that the sets of clusters are disjoint. Therefore, a clustering solution is essentially a set of disjoint sets. A common clustering task is to measure the similarity between two or more cluster solutions. This task typically arises when one wishes to quantify how close a clustering solution is to the optimal solution (if known), or when one wishes to compare the solutions given by different clustering algorithms. While this literature is concerned with computing the similarity between two or more *sets* of sets, nonetheless, techniques from this domain are relevant. Existing measures generally involve counting the number of similar/dissimilar pairs of elements, measuring the overlap of elements between the sets, or measuring the information shared between the sets (see [196] for a comprehensive review).

Counting and overlap-based similarity measures rely on the discrepancy between the *explicit* values within the sets, and so they will often fail to indicate similarity between scaled/translated but otherwise identical sets (e.g. $B = 10A$). Length scale is sensitive to scalings of f and/or $\mathcal{S}$, and so the resulting length scale sets for scaled, but structurally identical problems can vary by a factor. For example, the length scales for the 1-D quadratic are dependent on a, and so the length scale sets for two different realisations of this function will differ depending on the value of a. Thus counting and overlap based measures, including the Jaccard index, are not appropriate for measuring the similarity of length scale sets, and hence by proxy, the structural similarity of problems. To illustrate this point, the Jaccard index between length scale multisets from 1-D quadratics with $a = 1$ and $a = 2$ (enumerated in the same manner as above) is 0.1941, despite the functions differing by simply a scalar factor (and hence containing the same structural information).

On the other hand, information-based approaches utilise the probabilistic properties of the occurrences of elements in the sets, rather than the explicit values of the elements. Because of this, information-based similarity measures between length scale sets are more suitable for quantifying the similarity between problems. A popular technique in this area is the Variation of Information (VI), which is defined as [111]:

$$VI(A, B) = H(A) + H(B) - 2I(A, B) \tag{6.1.2}$$

where $H(A)$ is the entropy of A:

$$H(A) = -\sum_{a_i \in A} p(a_i) \log_2 p(a_i) \tag{6.1.3}$$

and $I(A, B)$ denotes the mutual information between A and B:

$$I(A, B) = -\sum_{a_i \in A} \sum_{b_j \in B} p(a_i, b_j) \log_2 \frac{p(a_i, b_j)}{p(a_i)p(b_j)} \tag{6.1.4}$$

where $p(a_i, b_j)$ is the probability that a_i and b_i are both in $A \cap B$. VI was developed specifically for measuring clustering solutions for a given dataset of size n, and therefore assumes $|A| = |B| = n$. It is a metric bounded by $log_2 n$, and so it can be normalised in order to provide a more intuitive measure that is within $[0, 1]$.

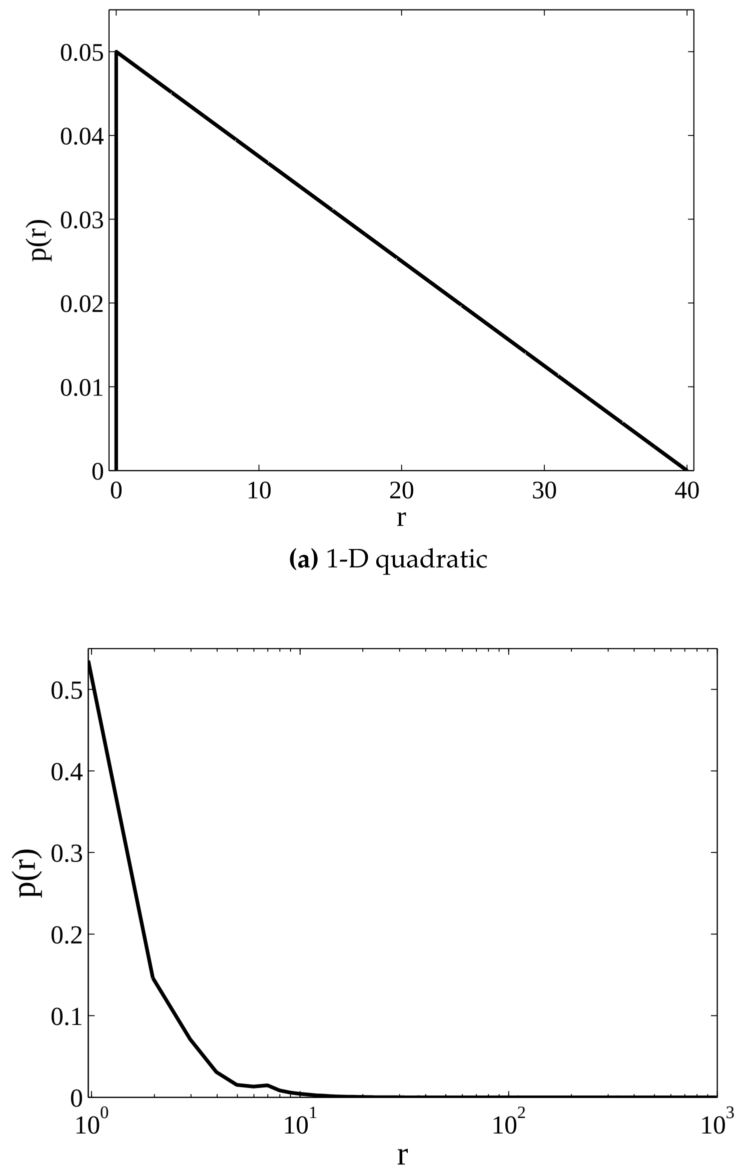
VI measures the amount of information that is lost *and* gained when the clustering solution A is used instead of B. For the purpose of measuring length scale set similarity, VI can be thought of as the amount of information that is lost and gained when the length scale set changes from A to B. As previously discussed, the length scale sets of 1-D quadratics with $a = 1$ and $a = 2$ contain the same *information*, and intuitively, the VI between them is 0 (indicating that no information is lost or gained by using one set over the other). In contrast, the length scale sets of the 1-D quadratic (with $a = 2$) and the mixed-structure function contain some common information (i.e. the mixed-structure function contains quadratics), however the mixed-structure set also contains additional information pertaining to structures not found in the 1-D quadratic. The (normalised) VI calculated between these sets is approximately 0.5524, which indicates that there is a mixture of shared and unique information in the sets.

While VI and other information-based set similarity techniques may be an attractive proxy for quantifying problem similarity, they are of course based on the finite set of length scales sampled, and so may vary across different samples. It is also difficult to apply these techniques to sets of differing sizes; when a discrepancy in size occurred in the above, the smaller of the sets were "padded" with additionally sampled length scales. Furthermore, because the normalisation is based on the size of the set, the values of the normalised VI should only be used to compare sets of the same size. For example, if the VI was calculated between two problems using 1000 length scale values, then it should only be compared to VIs calculated using 1000 length scale values.

In summary, the length scale set can be analysed and compared using sorted plots, scatterplots, box and whisker plots and explicit similarity measures. While counting and overlap-based methods are commonly used to quantify set similarity, they are not invariant to scaled (but structurally identical) sets. Information theoretic similarity measures, such as the Variation of Information, are based on the probabilistic properties of the sets, and consequently, are typically invariant to set scalings.

6.1.3 Length Scale Distribution

The length scale distribution, $p(r)$, provides a statistical model from which to visualise, analyse and interpret length scale values. The distribution is a highly useful summary of the length scale values; it can easily be visualised and facilitates the application of existing statistical analysis techniques. For example, the sorted length scale sets of the 1-D quadratic (where $a = 2$) and mixed-structure function shown in Figure 6.3 can be better visualised by their respective length scale distributions (using kernel density estimators with "solve-the-



(a) 1-D quadratic

(b) 1-D mixed-structure

Figure 6.4: Example length scale distributions.

equation plug-in" bandwidth estimation), shown in Figure 6.4.

The distributions in Figure 6.4 are clear summaries of the length scale information. As previously derived in Equation 5.3, the 1-D quadratic is a folded triangular distribution, and Figure 6.4a nicely illustrates where this name originates from. Figure 6.4a shows that the most probable *r* is 0, and as *r* increases, its probability decreases in a smooth, linear manner. The range of *r* values in Figure 6.4a indicates that the steps encountered in the sample do not contain drastic (relative) fluctuations in objective function value. The 1-D mixed-structure function also contains a frequent amount of small length scales, however large (relative) objective function fluctuations are also possible (note the logarithmic scaling of the x-axis in Figure 6.4b).

Measure	**1-D Quadratic**	**Mixed-Structure**
Mean	13.3337	1.8793
Median	11.716	1.1039
Mode	0	0
Standard Deviation	9.4283	2.4525
Range	39.998	1000
1st Quartile	5.36	0.5270
3rd Quartile	20	2.1814
Skewness	0.5657	9.0731
Kurtosis	2.4	1843.2

Table 6.1: Sample Estimates of Common Statistical Measures

The general hypothesis that follows is that problems with structure of similar complexities should yield similar length scale distributions. Subsequently, summaries of $p(r)$ are potentially very useful for characterising and comparing problems. Perhaps the most well-known summaries of distributions are classical statistical measures of central tendency (mean, median and mode), shape (skewness and kurtosis) and variability (range, quartiles and standard deviation). Statistical measures of $p(r)$ are summaries of the r values, and so it follows that they may not be unique, but will vary depending on the structure present in the sampled r values. Hence while two structurally different problems will yield different $p(r)$'s, there is no guarantee that their corresponding summary statistics are unique. Table 6.1 contains common statistical measures calculated for the length scale sets for the 1-D quadratic (where $a = 2$) and the mixed-structure function. Note that despite being different problems, the mode is 0 for both problems. The mean and median also measure the central tendency of the data, and Table 6.1 shows that the 1-D quadratic tends to be centred around length scales that are approximately an order of magnitude larger than the mixed-structure function. The skewness and kurtosis of the 1-D quadratic are much smaller than the mixed-structure function. Large skewness values indicate the degree of asymmetry in the data, while large kurtosis values indicate the heaviness of the distributions tail(s).

As for the length scale set, concepts from information theory can also be used to characterise $p(r)$. Shannon entropy is widely used as a measure of the uncertainty of a random variable [40]. Specifically, entropy measures the expected amount of information needed to describe the random variable. Let $\text{supp}(r)$ be the set of r values for which $p(r) > 0$ ($\text{supp}(r)$

is commonly referred to as the support set of r). The entropy of a probability density function is known as *differential entropy* and is defined as [40]:

$$h(r) = -\int_{\mathrm{supp}(r)} p(r)\log_2(p(r))\,\mathrm{d}r \tag{6.1.5}$$

Appendix B contains the definition for the discrete case. One important difference between (discrete) entropy ($H(r)$) and (continuous) differential entropy ($h(r)$) is that the $H(r) \geq 0$, while $h(r)$ can be negative.

Consider the circumstances in which $h(r)$ is minimal; the outcome of the random variable for the Dirac delta function can only be one possible value: a. Because only one possible value can occur, intuitively, there is absolutely no uncertainty and so the differential entropy is minimal. One might expect the differential entropy to be 0, however it is actually $-\infty$ [116]. This is because the density of the Dirac delta distribution can be substituted for the density of a Laplace distribution as $\lambda \to \infty$. By making this substitution, the differential entropy as $\lambda \to \infty$ is:

$$\lim_{\lambda\to\infty} \log_2\left(\frac{2e}{\lambda}\right) = \infty \tag{6.1.6}$$

Now consider the circumstances in which $h(r)$ is maximal. As previously mentioned, differential entropy measures the uncertainty of a random variable. Hence to maximise $h(r)$, the outcome of the random variable r must be maximally uncertain. Maximal uncertainty means that all possible outcomes of the random variable are equally probable, and consequently, r is uniformly distributed in a bounded region. Hence, uniformly distributed length scale values maximise $h(r)$. The differential entropy for $r \sim \mathcal{U}[a, b]$ is $log_2(b-a)$ [40].

In terms of landscape structure, the 1-D constant and linear functions are very simplistic and contain minimal structural information. The $p(r)$s for such landscapes are Dirac delta functions, which have minimal differential entropy. In contrast, very complex (i.e. random) landscape structure is required to produce uniformly varying length scale values. For such complex landscapes, $h(r)$ is maximal. Hence, there is a close relationship between the structural complexity of landscapes and the resulting value of $h(r)$. Indeed, given a particular landscape with unknown structural complexity within these two extremes, the resulting $h(r)$ can be used as an indicator of the structural complexity.

Analogous to the similarity measures between length scale *sets*, tools from information theory can be used to directly compare two length scale *distributions*. One commonly used measure of similarity between two distributions is the *relative entropy* (also known as the *Kullback-Leibler divergence*) [92]. Given two length scale distributions $p(r)$ and $q(r)$, the relative entropy measures how similar $q(r)$ is to $p(r)$, and it is defined by:

$$D_{KL}(p||q) = \int_0^\infty p(r) \log_2 \frac{p(r)}{q(r)} \mathrm{d}r \tag{6.1.7}$$

where the convention $0 \log_2 0 = 0$ is used and p is absolutely continuous with respect to q (see Appendix B for discrete definitions).

Relative entropy is not a symmetric measure and $D_{KL}(p||q) \neq D_{KL}(q||p)$ in general. To obtain a symmetric measure, one can instead use the *symmetric Kullback-Leibler divergence*, also known as the *Jeffrey divergence* or *J-divergence*, which was originally defined by Kullback and Leibler in [92] (Equation 2.5):

$$D_J(p||q) = D_{KL}(p||q) + D_{KL}(q||p) \tag{6.1.8}$$

Given that the relative entropy can be used to measure the similarity between two distributions, the relative entropy between length scale distributions is a proxy for comparing the structural characteristics between landscapes. This insight is a crucial contribution of the thesis; the structural similarity between two problems can be quantified without explicitly measuring and comparing specific structural properties. Consequently, the J-divergence is a primary component of the framework proposed to analyse and characterise optimization problems.

While the explicit dissimilarity estimates resulting from calculating the J-divergence are useful for directly comparing problems, they can also be further analysed to gain insight into the *relationship* between problems. For example, given a set of n problems, the dissimilarities between all $\frac{1}{2}n(n-1)$ pairs of problems can be calculated. There are a plethora of techniques that can be applied to infer relationships from dissimilarity data, and in this thesis two popular techniques are utilised: hierarchical clustering and dimensionality reduction.

As mentioned above in Section 6.1.2, cluster analysis is used to infer groups and relationships in multivariate data. Hierarchical clustering is a well-known clustering technique that can be used to visualise potential clusters or groups of objects from a given set [4].

Dimensionality reduction techniques are frequently used in machine learning to reduce a D-dimensional dataset to K dimensions, typically for $K \ll D$ [93]. To achieve a reduction in dimensionality, a measure of distance or similarity is used to quantify the inter-point relationships, and the techniques strive to embed the points in the reduced dimensional space while preserving the inter-point relationships. Here, the inter-point relationships become more important than the actual points themselves, and so there exist a subset of dimensionality reduction techniques that operate purely on the similarities between points (i.e. they require no knowledge of the original dataset). In doing so, these techniques take a similarity matrix as input, and output a K-dimensional dataset of points that are spatially distributed according to the input similarities. Therefore, dimensionality reduction techniques that operate on similarity matrices can be applied to pairwise J-divergence values, and subsequently be used to produce 2-D and 3-D visualisations of the relationships between problems.

Well-known dimensionality reduction techniques that operate on similarity matrices include Principal Component Analysis, Multi-Dimensional Scaling and Stochastic Neighbour Embedding (SNE). A current state-of-the-art dimensionality reduction technique is t-SNE; a probabilistic, non-linear method that aims to distribute points in a lower-dimensional space such that the original, high-dimensional neighbourhood relationships are preserved [186]. To achieve this, a non-convex cost function modelling the discrepancy between the low and high dimensional relationships is minimised using a variant of stochastic gradient descent. t-SNE is parameterised by a *perplexity* term, which essentially controls the number of effective neighbours near a given point.

6.2 Sampling Length Scales in Practice

As mentioned previously, when exact derivation and/or enumeration of the length scales for a problem is infeasible, a representative sample of length scales can be analysed instead. Let $p(r)$ be the true length scale distribution, and let $\hat{p}(r)$ be the length scale distribution estimated from a finite sample of length scales. Intuitively, as the number of sampled length scales increases towards complete enumeration, $\hat{p}(r)$ converges to $p(r)$. However in practice, the methodology used to sample r and the overall size of the sample will affect the convergence of $\hat{p}(r)$ to $p(r)$.

Two solutions are required to compute a single r value, and so a sample of solution pairs is required to construct a sample of r. There are many different methods and schemes applicable to calculating and collating length scale values from samples of solutions. One method is to calculate the r values between all unique pairwise combinations of a sample of solutions. More specifically, with an initial sample of m solutions (assumed to adequately cover $\mathcal{S}$), all $\binom{m}{2}$ unique combinations of pairs are used to construct a sample of length scales.

Using this technique, $\frac{m(m-1)}{2}$ length scales are sampled from m unique solutions in the landscape. While this approach uses the maximum information available from m solutions, it is limited in that a length scale sample of size $O(n)$ is based on only $O(\sqrt{n})$ unique solutions. It is conjectured here that to obtain a sample of length scales representative of the true distribution, r should be sampled from as wide a variety of solutions in $\mathcal{S}$ as possible. In the extreme case, n length scales can be calculated using n pairs of unique solutions, i.e. $2n$ unique solutions. If computational effort/storage is an important consideration, the number of unique solutions used to generate the length scales can be reduced. For example, n length scales can be generated via a sample of n unique solutions by pairing each solution with exactly two other solutions. One way to achieve this is by randomly permuting the order of the samples and calculating r between subsequent solutions in the (permuted) sample. This method is further outlined in Algorithm 6.1, where a multiset of length scales (using Euclidean distance) are calculated from a given sample and objective function.

Algorithm 6.1 Generation of the length scale multiset

```
Input:
    Sample of solutions, $\mathcal{S}' \leftarrow [\mathbf{x}^1, \ldots, \mathbf{x}^n]$
    Objective function, $f : \mathcal{S}' \rightarrow \mathbb{R}$
$\mathcal{S}'' \leftarrow RandomPermutation(\mathcal{S}')$
for $i \leftarrow 1$ to $n$ do
    $\mathbf{x}^i \leftarrow \mathcal{S}''[i]$
    $\mathbf{x}^j \leftarrow \mathcal{S}''[(i+1) \bmod n]$
    $\mathbf{r}[i] \leftarrow \frac{|f(\mathbf{x}^i)-f(\mathbf{x}^j)|}{\|\mathbf{x}^i-\mathbf{x}^j\|}$
end for
return r
```

The method used to generate the initial samples of solutions is an important aspect of the length scale analysis framework and deserves careful consideration. As investigated in the context of Dispersion and FDC in Chapter 4, uniform random sampling of high dimensional continuous problems can yield a sparse sample where the Euclidean distances between solutions are similar [3, 13]. Hence, a uniform random sample is not ideal for generating length scales in high dimensions; the denominator of r (i.e. the distance between solutions) would be similar across all sampled r, thereby essentially reducing r to the magnitude of change in the objective function. The purpose of the length scale analysis framework is to analyse the objective function at a *variety of scales* (i.e. distances), and so a sample of solutions at varying distances apart in $\mathcal{S}$ is required. Lévy random walks were successfully used in Chapter 4 to produce representative samples continuous optimization problems, appropriate for subsequent landscape analysis. Hence in this thesis, candidate solutions of continuous problems are sampled using a Lévy random walk, where steps are taken in a random, isotropic direction and step sizes are sampled from a Lévy distribution [156]. As previously discussed in Section 4.3, the Lévy distribution pertaining to step size is defined by scale (γ) and location (δ) parameters. δ determines the minimum possible step size, and is therefore set to 0 in all experiments in this thesis. γ essentially controls the magnitudes of step sizes generated. To determine appropriate values of γ, the distributions of distances between solutions generated were examined, and γ was adjusted to ensure that steps spanning the diameter of $\mathcal{S}$ were obtained.

The sample size required to produce an adequate sample will vary based on the structure of the landscape. For the 1-D linear and constant objective functions, any pair of solutions will yield the single length scale value that captures the inherent simplicity of the problem's single structural feature (i.e. slope). However, problems with more complex structures, such as the 1-D mixed-structure function defined in Example 6.1, will require many samples to adequately explore and capture the characteristics of the landscape structures.

Of course, the underlying structure of the problem is unknown in the black-box scenario, and so choosing an appropriate sample size is difficult in practice. Even when structural information is available, the number of solutions required to adequately sample and summarise a structure is unclear. For example, 20001 solutions were used to sample the 1-D quadratic displayed in Figure 5.2, but perhaps more or less solutions could have been sampled to achieve a similar result. In this thesis, sample sizes are made as large as practically possible. In addition, a methodology is proposed below to assess the convergence of a sample.

6.2.1 Assessing Sampling Adequacy with Length Scale Analysis

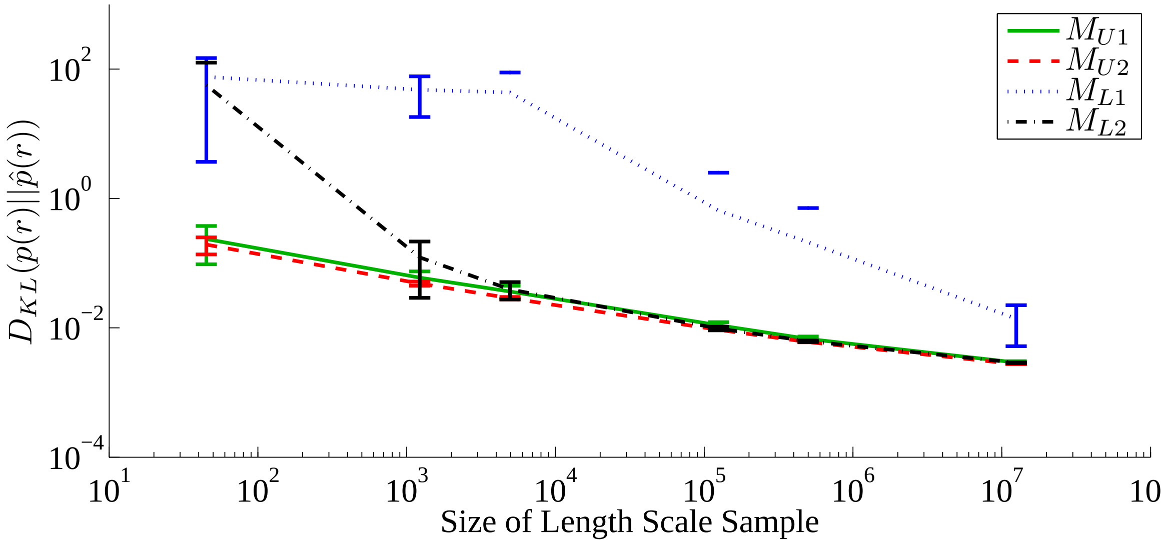
If $p(r)$ is known, the KL-divergence can be used to directly measure convergence, since $D_{KL}(p \mid\mid \hat{p}) = 0$ when $\hat{p}(r) = p(r)$. Often, $p(r)$ is unknown, and so the KL-divergences between different sample sizes (i.e. $D_{KL}(\hat{p}_{n+1} \mid\mid \hat{p}_n)$) can be assessed as an indicator for convergence. That is, once an adequate sample size is achieved, subsequent sampling will not drastically alter the distribution, and so the KL-divergence between the subsequent sample size and the current sample size will be negligible.

The following experiment investigates the conjecture that length scale is affected by the variety of the solutions in the sample. Specifically, the following sampling methodologies are compared:

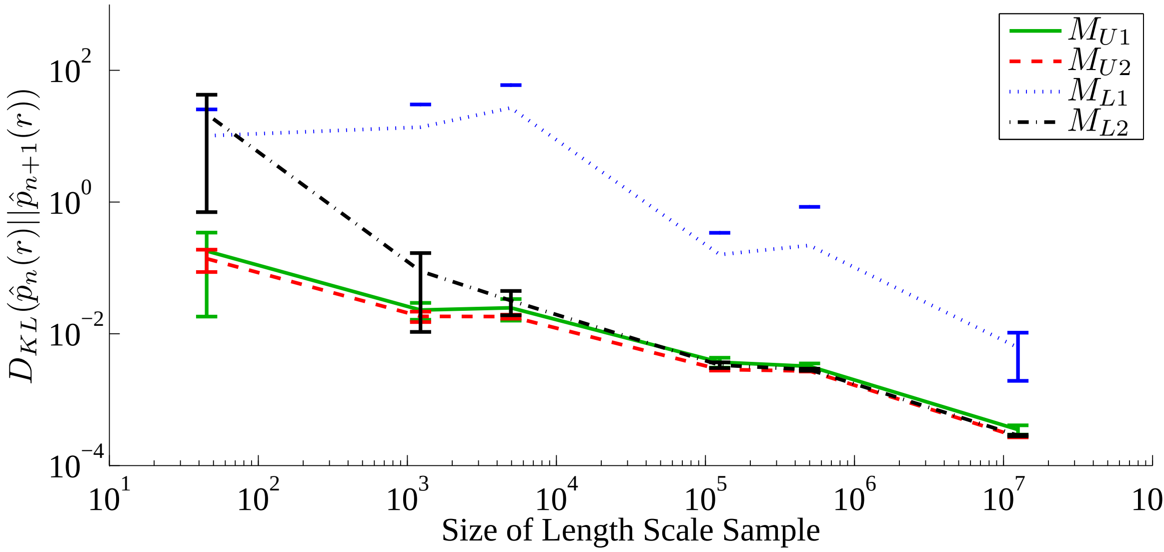
- M_{U1}: generate a uniform random sample of n solutions and calculate $\binom{n}{2}$ length scales from all pairwise solution combinations.
- M_{U2}: generate a uniform random sample of $\binom{n}{2}$ solutions and calculate $\binom{n}{2}$ length scales using Algorithm 6.1.
- M_{L1}: generate a Lévy random walk of n solutions and calculate $\binom{n}{2}$ length scales from all pairwise solution combinations.
- M_{L2}: generate a Lévy random walk of $\binom{n}{2}$ solutions and calculate $\binom{n}{2}$ length scales using Algorithm 6.1.

To evaluate the different methodologies, length scales are calculated for Example 5.3 (1-D quadratic function), where $p(r)$ is known. Using each sampling methodology, $\binom{n}{2}$ length scale values are generated, where $n = [10, 50, 100, 500, 1000, 5000, 10000]$, and 30 different samples are generated for each n. To obtain a wide coverage of $\mathcal{S} = [-1, 1]$, γ is set to 10^{-3} for both Lévy walks. $\hat{p}(r)$ is estimated from the samples via kernel density estimation with a Gaussian kernel, using the "solve-the-equation plug-in" method [154] for bandwidth selection. The KL-divergence is estimated via numerical approximation of Equation 6.1.8. Figure 6.5a shows the mean and standard deviation (as error bars) of $D_{KL}(p \mid\mid \hat{p})$ for each sample size. In the black-box scenario, $p(r)$ is unknown, and so to practically assess the convergence of calculation, the KL-divergence between the distributions for each sample size and its subsequent sample size (e.g. $n = 10$ and $n = 50$) is calculated. The mean and standard deviation of the divergences between sample sizes is shown in Figure 6.5b. Since KL-divergence is non-negative, error bars yielding negative values are omitted from the figure.

Figure 6.5a shows that for small samples of r, both uniform random sampling methods are superior to the Lévy random walks on this 1-D problem. This seems reasonable as large steps in a Lévy walk are not as probable as small steps, and so it can take a number of samples before Lévy walks adequately explore the landscape. Interestingly, a larger diversity in the sample of solutions appears to produce better-represented length scales, as illustrated by the fact that $D_{KL}(p \mid\mid \hat{p})$ for M_{L2} is ≤ 1 much faster than M_{L1}. Furthermore, on this problem M_{L2} is comparable to uniform random sampling after 10^3 samples. Thus the conjecture that a variety of solutions yields well-represented length scales is well-founded; for both uniform random sampling and Lévy random walks, using Algorithm 6.1, as opposed to calculating the length scales between *all* solution-pair combinations, gives a more accurate sample for



(a) $D_{KL}\left(p \mid\mid \hat{p}\right)$



(b) $D_{KL}\left(\hat{p}_{n+1} \mid\mid \hat{p}_n\right)$

Figure 6.5: Estimating sampling adequacy via convergence of $\hat{p}(r)$.

almost all sample sizes.

The divergences of the $\hat{p}(r)$'s from a black-box perspective are shown in Figure 6.5b. Both variants of the uniform random sample have small divergences and are therefore quite stable, even for a low number of samples. M_{L2} achieves a very low divergence (say, ≤ 0.1 bits) after approximately 10^3 samples, whereas M_{L1} doesn't achieve low divergence until 10^6 samples. Small KL-divergences does not necessarily mean that $\hat{p}(r)$ has converged to $p(r)$, however they do indicate how much length scale information (in terms of bits) might be gained by sampling further. If little can be gained, either all the important structure has been sampled (in which case, $\hat{p}(r)$ is a good estimate of $p(r)$), or there exists important structure that is hard to find (e.g. a needle in a haystack). In the former case the sample is adequate, but in the latter case, near-complete enumeration is required, which is a major challenge for any sampling technique.

The trends in Figure 6.5b closely follow those in Figure 6.5a, suggesting that the black-box methodology proposed provides a good summary of convergence, and hence can reliably assist practitioners in determining the adequacy of their samples. This practical technique is used in the experiments within Chapters 7 and 8 to determine and ensure adequate sample sizes.

6.3 Practical Considerations

In practice, using a sample of r values may result in different landscapes yielding the same sets of length scales, and hence length scale summaries. Identical sets of r values can be obtained from two landscapes sharing similar structure, where the structure discriminating them is not captured in the sample. However, any practical landscape analysis technique is limited to the information obtainable via sampling. Compressing $O(n)$ length scale values into a single summary value (e.g. the mean of sampled r values or $h(r)$) may incur information loss. This too is an unavoidable issue for many existing landscape techniques. Hence, while the use of length scale summaries may aid in characterising and analysing problems, they are not necessarily unique for individual problem instances. For example, consider a flat (i.e. neutral) landscape and a needle-in-a-haystack (NIAH) landscape where there is a single, small global basin on an otherwise flat landscape. Clearly, if the global basin in the NIAH landscape is not sampled, then the two landscapes will yield identical length scales. Unfortunately, inadequate sampling is an unavoidable issue for all practical landscape analysis techniques, and length scale is certainly no exception.

A sample of n length scale values requires $O(n)$ storage. For large representations of r, such as the IEEE Standard 754 for double-precision (64-bit) floating points used throughout this thesis, storage can be cumbersome for very large n. For example, $2^{41} \approx 2.199 \times 10^{12}$ length scales (represented using double precision) can be loaded into 16GB of RAM. Like any large dataset, compression schemes, such as applying a lossless compressor, can be used to reduce the burden of *persistent* length scale storage (e.g. on a hard disk).

Computation of a single r value involves a ratio of the distance between objective function values (scalar values) and the distance between two solutions (multivariate values). Hence, assuming dimensionality D, a single length scale is computed in $O(D)$ time, and a set of n length scale values is be computed in $O(nD)$ time. As previously discussed in Section 5.3.1, the evaluation of m points from a kernel density estimator built using n points with the ϵ-exact bandwidth selection algorithm [135] requires $O(mn^2)$ time. Hence because the J-divergence requires iterating over the m evaluation points for the two distributions being compared, it runs in $O(mn^2)$ time. Therefore to calculate the J-divergence between two optimization problems, n length scales are sampled (($O(nD)$) the J-divergence is calculated based on m evaluation points from a kernel density estimator ($O(mn^2)$), all of which totals to $O(mn^3D)$ time.

The success of the length scale analysis is heavily dependent on the size of the length scale sample, n. While the computational time is, in the worst case, cubic with n, approximations and modifications can be made to reduce the complexity. For example, the complexity of the bandwidth selection step can be reduced by using a *sub-sample* of randomly sampled r values from the original set. Using a sub-sample of size $p \ll n$, the total running time of computing a J-divergence is $O(mpn^2D)$. Careful experimental design choices can also reduce the computational complexity. For example, the J-divergences between all problems within a *problem set* can be calculated such that the evaluation points from each kernel density estimator are computed only once, yet used multiple times.

6.4 Summary

Analytical properties of length scale have been discussed and techniques for problem analysis were proposed using statistical and information-theoretic summaries of length scale. Length scale analysis on simple example problems illustrated the framework's ability to capture important problem structures and the complexity of their interactions. A major contribution of this chapter is the application of the entropic Jeffery divergence (J-divergence) for quantifying the similarity between length scale distributions. The J-divergence between length scale distributions effectively measures the similarity of the length scale information between problems, and hence is a proxy for the similarity between problems. The J-divergence between length scale distributions was also used to develop a novel methodology for assessing the adequacy of samples (that can vary in size) from the landscape. The proposed methodology is applicable to sampling both continuous and combinatorial optimization problems.

CHAPTER 7

Length Scale Analysis: Results

An experiment is a question which science poses to Nature, and a measurement is the recording of Nature's answer.

Max Planck

This chapter utilises the length scale analysis techniques proposed in Chapter 6 to analyse continuous and combinatorial optimization problems. To investigate the ability of the length scale information to capture structural features, experiments on artificial continuous problems, the Black-Box Optimization Benchmarking (BBOB) problem set, Circle in a Square (CiaS) packing problems, the Travelling Salesman Problem (TSP) and the Number Partitioning Problem (NPP) are presented. In addition to analysing the length scale values of these problems, the experiments also include a comparison of length scale and several popular landscape analysis methods.

7.1 Analysis of Continuous Artificial Problems

7.1.1 Elliptical Function

The 2-D elliptical function, defined in Table A.1, is essentially a quadratic bowl with elliptical contours, where the eccentricity of the contours is defined by a constant, $a \in \mathbb{R}$. Larger values of a yield functions with steeper, narrower contours, which have been shown to be a problematic landscape structure for certain Estimation of Distribution Algorithms [23, 64, 117]. Therefore, the elliptical function provides a simple and intuitive landscape from which the ability of the length scale analysis to capture varying levels of eccentricity can be assessed.

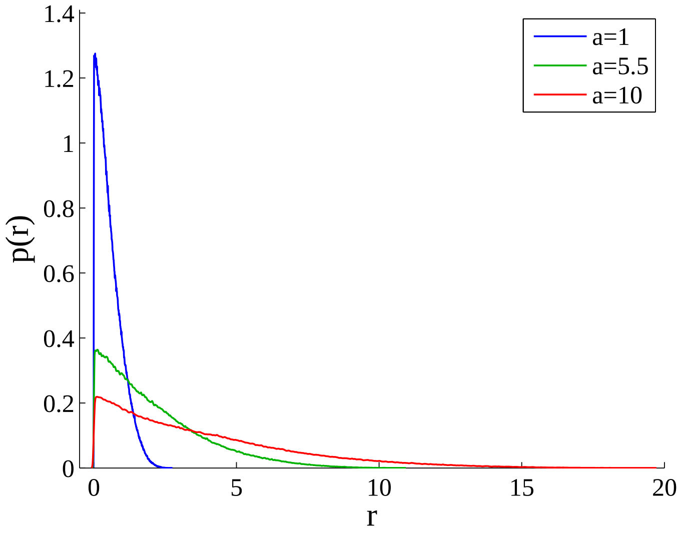


Figure 7.1: Example length scale distributions for ellipse functions with $a = 1, 5.5$ and 10.

The aim of this small experiment is to investigate how well the length scale analysis captures the structural changes between elliptical functions with varying degrees of eccentricity. The elliptical functions used are defined in Table A.1, where $\mathbf{x} \in \mathcal{S} = [-1,1]^2$ and $a \in [1, 1.25, \ldots 10]$. Therefore, a total of 37 elliptical functions are analysed. At each value of a, $2.5 \times 10^5 D = 10^6$ length scales are generated using Algorithm 6.1 with samples from a Lévy random walk parameterised by $\gamma = 10^{-3}$ and $\delta = 0$. Figure 7.1 shows the length scale distributions of three different elliptical functions at equal intervals throughout a, i.e. $a = 1, 5.5$ and 10.

The length scale distributions vary significantly depending on the value of a. Figure 7.1 illustrates that the vast majority of length scales are quite small ($0 \leq r \leq 2.5$) for low values of a. As a increases, larger length scales occur, resulting in a longer and thicker tail. Despite the increased prevalence of larger r, the mode of all of the distributions is 0.

For the length scale data obtained over all values of a, heatmaps summarising the J-divergence, D_J, between all pairs of problems as well as t-SNE (perplexity of 5) visualisations of these D_J values are shown in Figures 7.2 and 7.3. Due to the stochastic nature of t-SNE, 1000 different trials were conducted, with a maximum of 1000 iterations for each trial. The results show the best (i.e. lowest cost) t-SNE result, and the cost of 0.2040 indicates that the discrepancy of distances between points in the original data and reduced data is moderate, and so the visualisation is not able to fully reflect the relationships between the D_J values. The D_J values reflect the similarity between length scale distributions, and is therefore a proxy for the similarity between problems. Hence, pairs of problems with small D_J values will likely be close in proximity in t-SNE reductions, and so Figure 7.3 (and other t-SNE

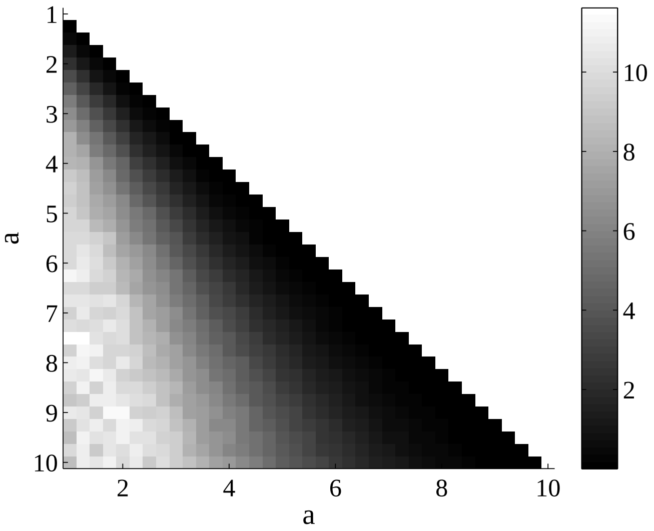


Figure 7.2: Heatmap of D_J values calculated between pairs of 2-D ellipse functions, where $a = 1, 1.25, \ldots, 10$.

visualisations) can be used to grasp an intuition into the relationships between the problems.

The heatmap in Figure 7.2 clearly shows that the greatest difference between problems (depicted by white pixels) are generally between low and high eccentricities (e.g. $a = 1$ vs $a = 10$). Problems with similar eccentricities (i.e. along the leading diagonal) have low D_J values, regardless of where on the eccentricity spectrum they are. For example, D_J between problems $a = 1$ and $a = 1.25$ is approximately the same as D_J between $a = 9.75$ and $a = 10$. This is true across all of a, that is, D_J between a_n and a_{n+1} remains constant and suggests that the structural changes caused by small increases in eccentricity are quite regular. The light colouring of the bottom-left corner of the heatmap in Figure 7.2 indicates that elliptical functions with low a values are highly different to elliptical functions with high a values. However, as a increases, the functions become increasingly more similar. The lower-right corner of the heatmap in Figure 7.2 is much darker than the upper-left, suggesting that as a increases, the difference between an instances at a set eccentricity apart becomes less pronounced. For example, the J-divergence between problems $a = 1$ and $a = 5.5$ is larger than the J-divergence between $a = 5.5$ and $a = 10$.

Figure 7.3 shows the t-SNE visualisation of the problem similarities (and hence, "problem space" according to D_J). Here, the problems are labelled with their respective a value, and the markers are shaded from white to black as a transitions from 1 to 10. Figure 7.3 also captures the general trend that problems with similar eccentricities are similar to each other; the problems are spatially ordered according to a. The problems are initially spatially ordered in a linear manner throughout both the dimensions of the reduced space (i.e. $a = 1$

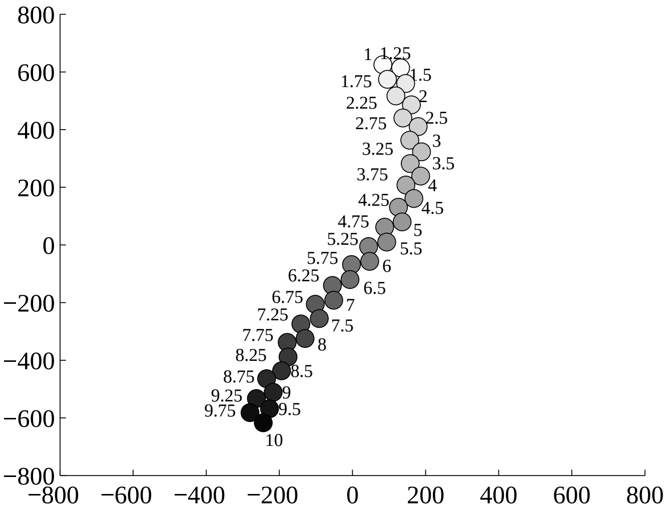


Figure 7.3: t-SNE of D_Js (cost of 0.2040) calculated between pairs of 2-D ellipse functions, where $a = 1, 1.25, \ldots, 10$.

to $a = 3.5$ roughly form a diagonal line). Then, at $a \approx 3.5$, the problems' positions change direction in the space, but are still ordered linearly throughout the space. The positioning of the problems ensures that the largest spatial distance is between low eccentricities and high eccentricities.

By examining the *bottom* of the dendrogram shown in Figure 7.4, it is clear that problems of similar eccentricities (a values) are clustered together. For example, $a = 1$ forms a cluster with $a = 1.25$ at approximately $D_J = 0.0721$, $a = 1.5$ forms a cluster with $a = 1.75$ at approximately $D_J = 0.0618$, and so forth. Overall, the dendrogram is quite balanced; moving upwards from the bottom, instances typically form clusters that double in size. There is however a slight skew in the balance; instances $1 \leq a \leq 3$ are separated from the remainder of the problems. By examining the dendrogram from the *top*, it is clear that the problems form two major clusters; $1 \leq a \leq 3$ and $3.25 \leq a \leq 10$. The large D_J connecting these clusters indicates that they are quite well-separated. Moving downwards, the problems can be further clustered into four clusters by using a D_J threshold of approximately 2 (the clusters can be identified by drawing a line across $D_J = 2$): $1 \leq a \leq 1.75, 2 \leq a \leq 3, 3.25 \leq a \leq 6$ and $6.25 \leq a \leq 10$. The size of the four clusters are non-uniform and increase as a increases. For example, the cluster $1 \leq a \leq 1.75$ has only 4 instances, whereas the cluster $6.25 \leq a \leq 10$ has 16. Thus it is clear from the dendrogram that as a increases, the problems become generally more similar.

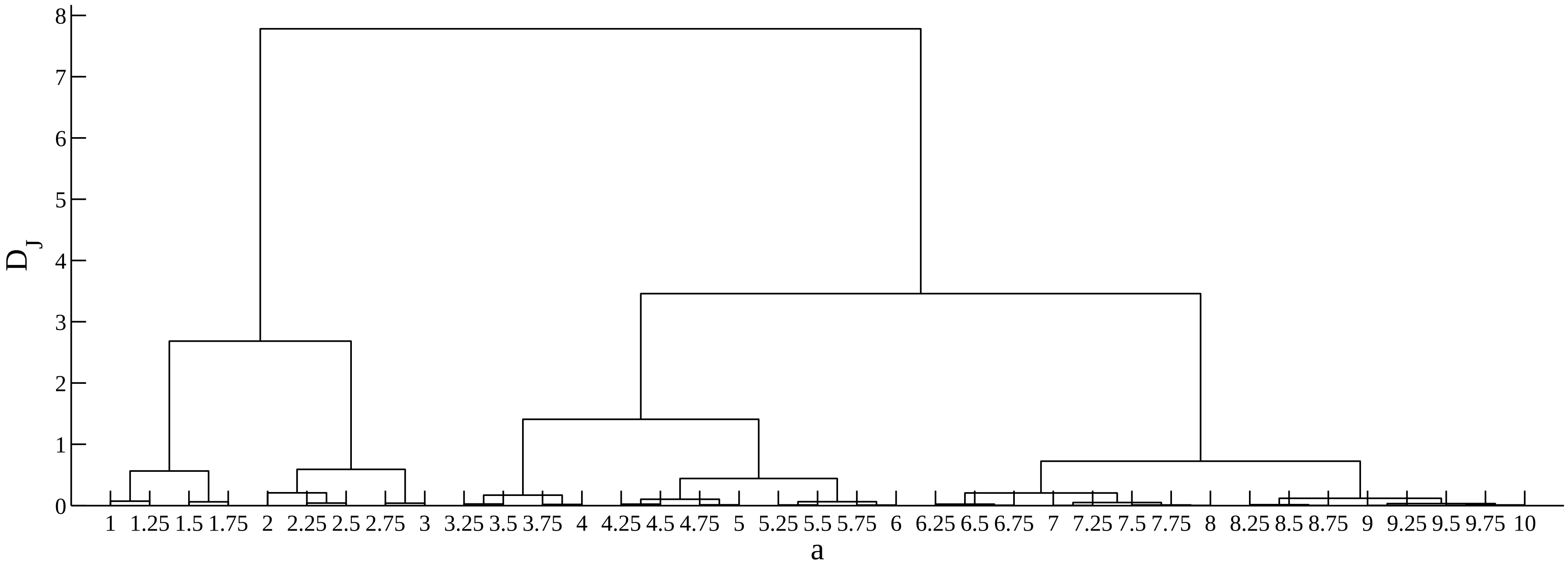


Figure 7.4: Dendrogram of D_Js calculated between pairs of 2-D ellipse functions, where $a = 1, 1.25, \ldots, 10$.

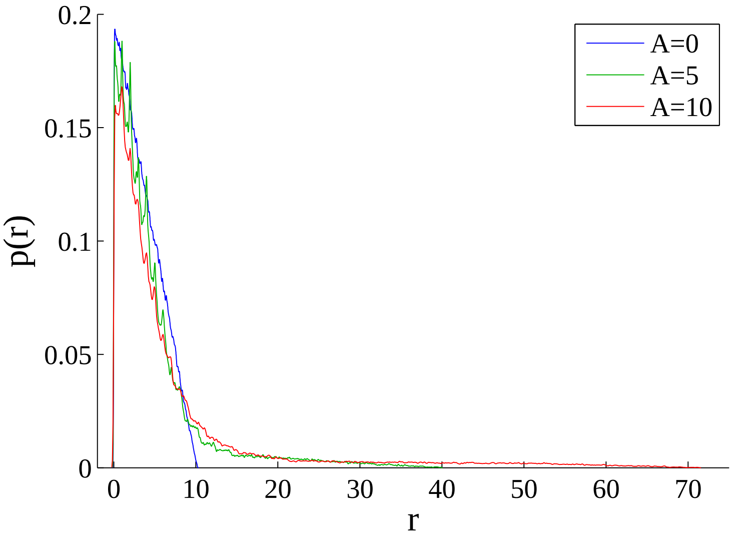


Figure 7.5: Example length scale distributions for 1-D Rastrigin.

7.1.2 Rastrigin

The perturbation term A in the Rastrigin function (Table A.1) dictates the ruggedness of the problem landscape. The goal of this experiment is to investigate how the change in perturbation affects the length scales of the problems. Intuitively, problems with similar perturbation values are likely to be more similar than problems with very different levels of perturbation. The dimensionality is fixed at $D = 1$, and $A = 0, 0.25, \ldots, 10$. A total of 2.5×10^5 solutions are sampled from $\mathcal{S} = [-5.12, 5.12]$ using a Lévy random walk parameterised by $\gamma = 0.005$ and $\delta = 0$. Figure 7.5 shows the length scale distributions three different Rastrigin functions using $A = 1, 5$ and 10.

In contrast to the elliptical functions in Figure 7.1, the length scale distributions of the Rastrigin functions do not vary significantly. Because 1-D Rastrigin with $A = 0$ is simply a 1-D quadratic with $a = 1$, it is expected that $p(r)$ for $A = 0$ to be a triangular distribution. Figure 7.5 confirms this, and it shows that as A increases, the tail length and thickness increases, indicating that larger length scales become more frequent. Large length scales occur when there are relatively large fluctuations in objective function values, which is indeed the case for large A. Hence, the length scale distribution is sensitive to the structural changes in the Rastrigin function.

Heatmaps summarising the D_J values between Rastrigin problems as well as t-SNE (perplexity of 5, best result from 1000 trials with a maximum of 1000 iterations) visualisations of the problem space are shown in Figures 7.6 and 7.7. The cost values from t-SNE indicate that the discrepancy of distances between points in the original data and reduced data is low.

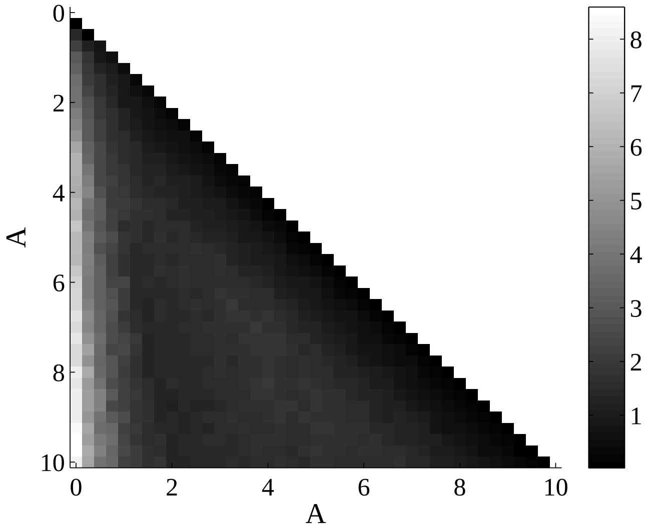


Figure 7.6: Heatmap of D_J values calculated between pairs of 1-D Rastrigin functions with $A = 0, 0.25, \ldots, 10$.

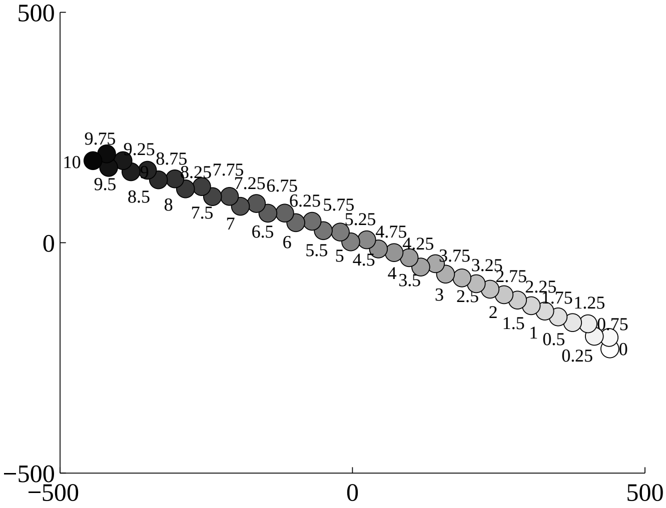


Figure 7.7: t-SNE of D_Js (cost of 0.1303) calculated between pairs of 1-D Rastrigin functions with $A = 0, 0.25, \ldots, 10$.

Overall, the heatmap and t-SNE visualisation shown in Figures 7.6 and 7.7 are very different to those resulting for the ellipse functions (Figures 7.2 and 7.3). The heatmap in particular shows that the largest difference between problems is between $A = 0$ and $A = 10$, and that for $A > 1$, the similarity between problems is rather constant ($D_J \approx 2$). Problems close to the leading diagonal are an exception; the D_J values are very low, indicating that problems with similar A are structurally similar.

Problems in the t-SNE visualisation (Figure 7.7) are labelled with their respective value of A, and the markers are shaded from white to black as A transitions from 0 to 10. Appropriately, the largest amount of space in the visualisation is between $A = 0$ and $A = 10$. Interestingly, the problems are almost perfectly ordered/organised in a sequential, linear manner according to their respective values of A. This is a clear reflection of the gradual change that the increase in perturbation causes on the landscape structure.

Importantly, the above analysis shows that the length scale values, together with D_J and t-SNE, make a framework that is very good at identifying the known/induced relationship between these problems, with *no* prior knowledge, based purely on black-box samples from the landscape.

The dendrogram in Figure 7.8 shows the clusters produced by hierarchical clustering with unweighted average distance linkages, and is also able to give an excellent representation of the increase in problem similarity that occurs between Rastrigin problems as A increases. Looking at Figure 7.8, as A increases, the heights of the connection between neighbouring As (e.g. $A = 5$ and $A = 5.25$) generally decreases, indicating that the D_J values between neighbouring problems is decreasing. An exception to this is $A = 0$ and $A = 0.25$, which are also very small. This behaviour is likely due to the nature of the landscape; for $A = 0$ and $A - 0.25$, the perturbations are small (and perhaps negligible) resulting in low D_J values. Then, as A increases, the perturbations begin to majorly alter the landscape structure, and so problems have larger J-divergences. Finally, a threshold is reached where the perturbations are so frequent and large that the problems once again resemble each other. Using the dendrogram to cluster the problem set into two groups yields $A_1 = \{0, \ldots, 1.25\}$ and $A_2 = \{1.5, \ldots, 10\}$. Excluding the initial split into two clusters, the dendrogram is quite balanced; clusters tend to recursively split in an even manner. In contrast to the elliptical functions in Figure 7.4, the dendrogram of the Rastrigin problems has many more "levels" of clustering, indicating that there are no obvious major clusters. This corroborates the heatmap, which shows that the majority of D_J values between problems are alike.

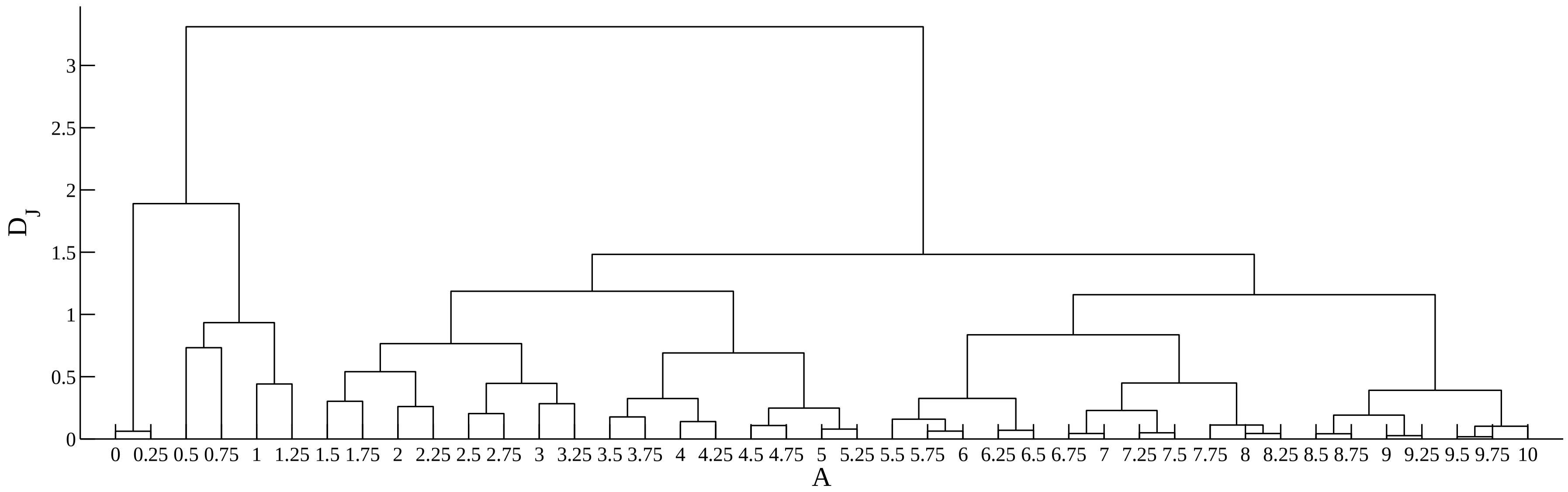


Figure 7.8: Dendrogram of D_Js calculated between pairs of 1-D Rastrigin functions with $A = 0, 0.25, \ldots, 10$.

7.2 Analysis of BBOB Problems

The noiseless BBOB problem set (see Appendix A.1.2) is comprised of 24 artificial (single-objective) continuous optimization problems that scale with dimensionality. The problems are unbounded, unconstrained, noiseless and generally treated as black-box functions for the purposes of benchmarking algorithms (although global optima are known for all problems, they are use only to quantitatively analyse algorithm performance). The developers of the BBOB problem set remark [72]:

> Our intention behind the selection of benchmark functions was to evaluate the performance of algorithms with regard to typical difficulties which we believe occur in continuous domain search. We hope that the function collection reflects, at least to a certain extend and with a few exceptions, a more difficult portion of the problem distribution that will be seen in practice (easy functions are evidently of lesser interest).

There is little doubt that the BBOB problems contain a variety of landscape structures; F1 is a quadratic bowl, F2 contains elliptical contours, F3 is highly multimodal and hence rugged, F5 is a linear slope, F7 mainly consists of plateaus, while F8, F12 and F13 contain valleys and ridges. In addition, many of the landscapes consist of *multiple* structures (e.g. F21), which combine and interact to create complex hybrid-structures. However, the BBOB problems are highly contrived and purposefully constructed to contain "typical" landscape features thought to be difficult for heuristic search. Therefore, the structural features within the problem set are quite biased to the landscape structures that the developers believe are typically difficult. Consequently, it is unlikely that the BBOB problems represent a large proportion of the distribution of problems observed in practice [105]. Furthermore, given the limited understanding of what landscape structures and features contribute to the difficulty of continuous optimization problems, it is also highly unlikely that the BBOB problems represent a difficult portion of the problem distribution observed in practice.

Despite the contrived, and hence biased, nature of the BBOB problems, they are clearly more complex than the highly artificial 2-D elliptical functions and 1-D Rastrigin functions examined in Section 7.1. There is also well-documented knowledge [112, 113, 120] and intuition [72] regarding BBOB landscape structures and features, which is highly useful in assessing the validity of results from landscape analysis techniques. Therefore, the BBOB problems provide an excellent set from which to investigate the efficacy and robustness of the length scale analysis framework, particularly in comparison to existing landscape analysis techniques.

In this section the ability of the length scale analysis framework to characterise BBOB problems is investigated and compared with existing landscape analysis techniques. More specifically, the experiments aim to investigate how well the correlation length, FDC, information content, partial information content, information stability, dispersion and the entropy of the length scale distribution ($h(r)$) can differentiate between different problems within the problem set. These features were chosen because they yield scalar values, are easy to interpret and (with the exception of $h(r)$ of course) are widely used in the landscape analysis literature. The robustness of the length scale analysis, and whether or not there is a relationship between r and the "difficulty" of problems (as measured by the best performing algorithms out of all BBOB competitions prior to 2015) is also investigated. Problems with largely varying length scales are likely to contain a richer, more complex landscape. Consequently, the behaviour and performance of algorithms may be reflected by the length scale analysis. The methodology used in these experiments is general and can be easily applied to other black-box problems.

Each feature is calculated from a sample of solutions resulting from a Lévy random walk in $\mathcal{S} = [-5, 5]^D$, where D is the dimensionality of the problem. 2-D, 5-D, 10-D and 20-D problems are analysed and Euclidean distance is used as the distance metric between solutions. The range of solutions yielded from Lévy random walks with various parameter settings was examined, and a setting of $\gamma = 10^{-3}$ was found to produce widely ranging samples across D. Sample sizes of $1000D^2$, $5000D^2$ and $10000D^2$ were tested on instances from all the BBOB problems, across 2-D, 5-D, 10-D and 20-D, and there was only a negligible ($\leq$ 1 bit) average difference in sampling more than $1000D^2$ solutions, as seen in Figure 7.9 (for 20-D problems). Thus in this experiment, all features are calculated from a sample of solutions obtained using a Lévy random walk of $1000D^2$ solutions in $\mathcal{S} = [-5, 5]^D$.

The robustness of the features is investigated by examining them over varying instances of the problems, and varying samples of those instances. 30 problem instances are produced by supplying seeds 1 to 30 to the BBOB problem generator. For each instance, 30 different samples (of size $1000D^2$) of $\mathcal{S}$ are generated, meaning for a given problem in dimension D (e.g. 2-D Sphere), there are 30 × 30 samples of the problem. Hence each feature is calculated 900 times for a *single* problem.

FDC is calculated using the global optimum, $\mathbf{x}^*$, as well as the best solution in the sample (each estimator is denoted as $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ respectively). The latter gives insight into

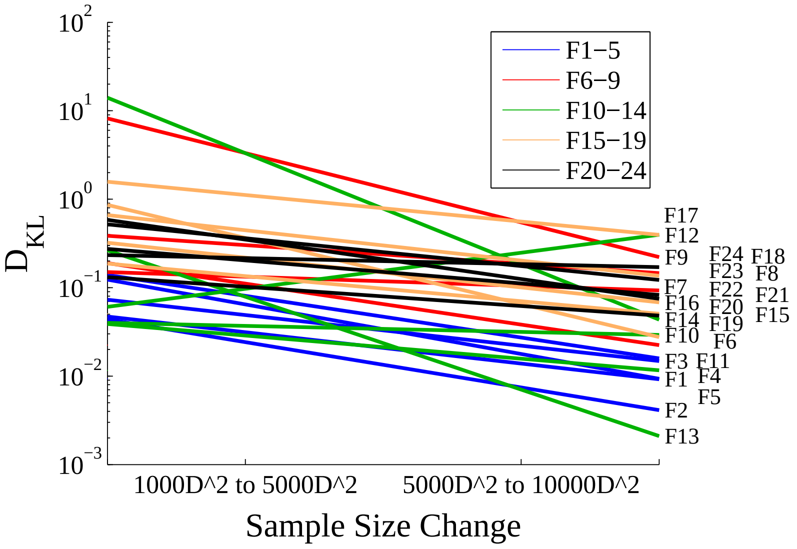


Figure 7.9: D_{KL} between $p(r)$s from subsequent sample sizes $1000D^2$, $5000D^2$ and $10000D^2$ on the BBOB problems in 20-D.

how well FDC performs when the problem is treated as a black-box. Information content and partial information content are estimated with $\epsilon = 0$, meaning transitions in objective function are “neutral” if and only if the change in the objective function value is 0 (to machine precision). Dispersion is calculated using the fittest 5% of solutions in the sample, and it is normalised using bound-normalisation, described in Section 4.4.1. The correlation length, l, is calculated using Equation 3.2.4. Length scale distributions are estimated via kernel density estimation as described in Section 6.2, and the entropy, $h(r)$, is estimated by numerical evaluation of Equation 6.1.5.

Linear and non-linear transformations are applied to many of the BBOB problems, and hence length scales may vary between problem instances. Problems are also randomly translated, and while length scale is invariant to translation, the translation performed on these problems is analogous to shifting and re-labelling the bounds. Consequently, structure originally within the bounds may be removed, and structure originally outside the bounds may be introduced. This can affect the resulting length scales, and so length scale is not expected to be completely invariant between different problem instances. While the structural features that fundamentally define a problem may vary slightly across randomised instances, robust features will capture these structures consistently across the instances.

7.2.1 Length Scale Distribution Results

The length scale distributions varied widely across the BBOB problems. Due to the large variations in the length scales, it is difficult to visualise all distributions on a single graph.

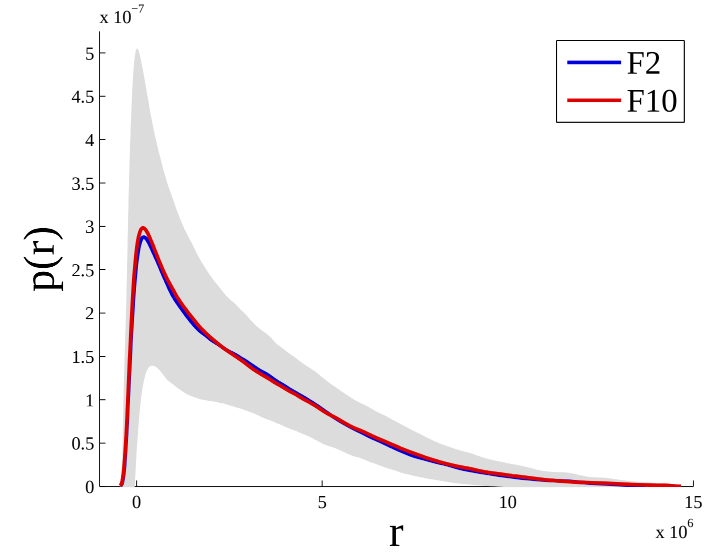


Figure 7.10: Ellipsoidal Function in 2-D.

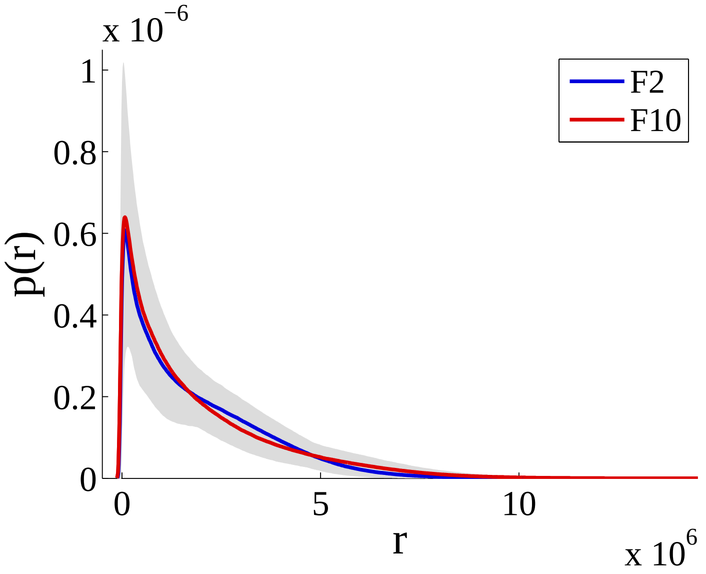


Figure 7.11: Ellipsoidal Function in 5-D.

Instead, groups of similar problems have been identified and visualised together.

Figures 7.10 to 7.13 show the length scale distributions for the 2-D, 5-D, 10-D and 20-D F2 and F10, both characterised as Ellipsoidal problems in the set (F10 is a rotated version of F2). The distributions are almost identical and are quite consistent across D. This empirically confirms that length scales are invariant to rotations of the search space. Interestingly, the variation between sampled length scale distributions, shown as the grey shading, is quite high for 2-D problems, and as D increases, the variability generally decreases. In general, there is more variation for the larger probabilities (e.g. the tail in Figure 7.10 has less variability than the mode), and so the high variation exhibited in the 2-D $p(r)$s may be because the shape of the distributions are more triangular in comparison to the 5-D, 10-D and 20-D

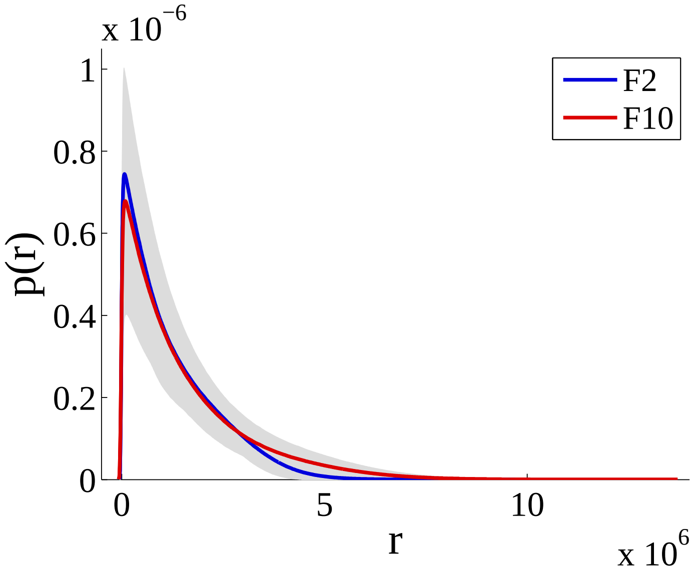


Figure 7.12: Ellipsoidal Function in 10-D.

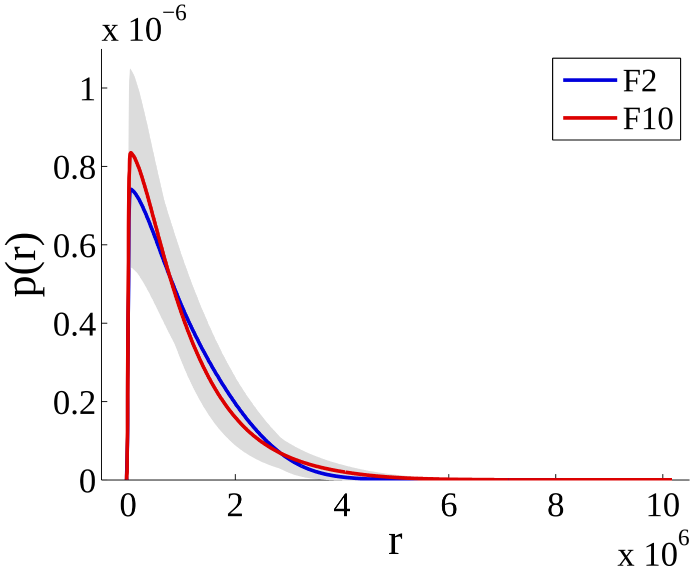


Figure 7.13: Ellipsoidal Function in 20-D.

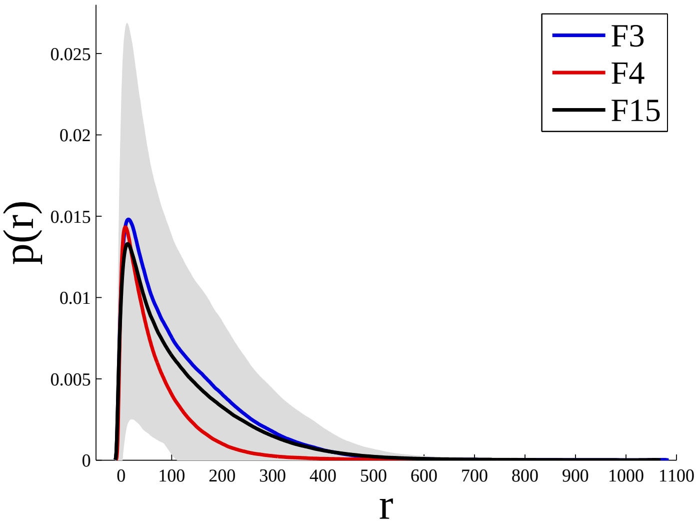


Figure 7.14: Rastrigin Functions in 2-D.

$p(r)$s.

As D increases, the range of length scales remains constant, however the shape of the distributions varies steadily. Specifically, the $p(r)$s in 2-D are quite "triangular", and as D increases, the tail becomes thinner and the mode is higher, indicating more "small" r values are encountered. This is likely due to the objective function, which is essentially $f(\mathbf{x}) = \sum_i^D 10^{6\frac{i-1}{D-1}} x_i^2$. Here, the objective function value of a solution is influenced by two factors; 1) the actual value of each solution component, x_i and 2) the value of the term $10^{6\frac{i-1}{D-1}}$ at each of the components. The term $10^{6\frac{i-1}{D-1}}$ varies between 1 and 10^6 in exponential increments dictated by D. Thus as D increases, there is a wider variety of terms spaced exponentially within $\left[1, 10^6\right]$. As a result, there are an increasing number of "small" terms as D increases, thus reducing the magnitude of f, and consequently, the size of the length scales. Therefore, the change in shape of $p(r)$ is a direct reflection on the nature of the objective function. The average J-divergences between F2 and F10 are approximately 6.3516 (2-D), 3.3611 (5-D), 2.6600 (10-D) and 0.8131 (20-D), indicating that the two problems are relatively more similar as D increases.

In contrast to Figures 7.10 to 7.13, $p(r)$ for F3 (Rastrigin), F4 (Büche-Rastrigin) and F15 (rotated Rastrigin) are shown in Figures 7.14 to 7.17 and vary across *both* the problems and D. An exception is F3 and F15; F15 is a non-separable and less regular variant of F3, and as a result, they have almost identical length scales. Once again this illustrates length scale's invariance to rotated search spaces. F4 contains similar, but asymmetric structure to F3, and the length scales are sensitive to this, causing a slightly different (but still similar) distribution. In 2-D, the distributions are all very similar, however as D increases, the tail of F4 becomes

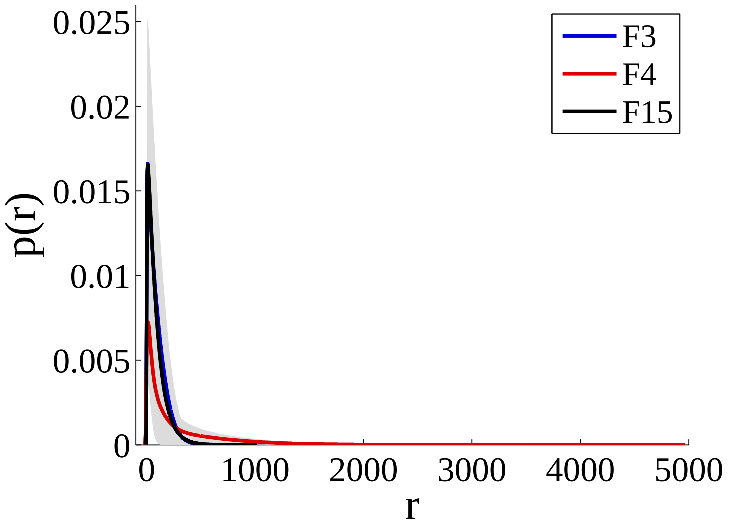


Figure 7.15: Rastrigin Functions in 5-D.

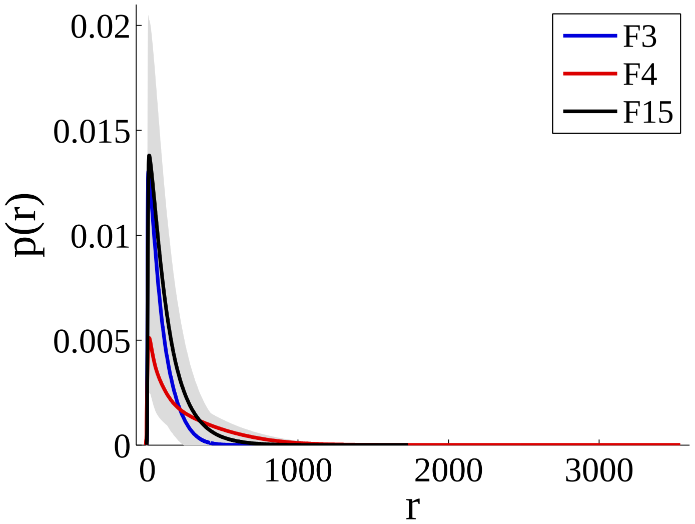


Figure 7.16: Rastrigin Functions in 10-D.

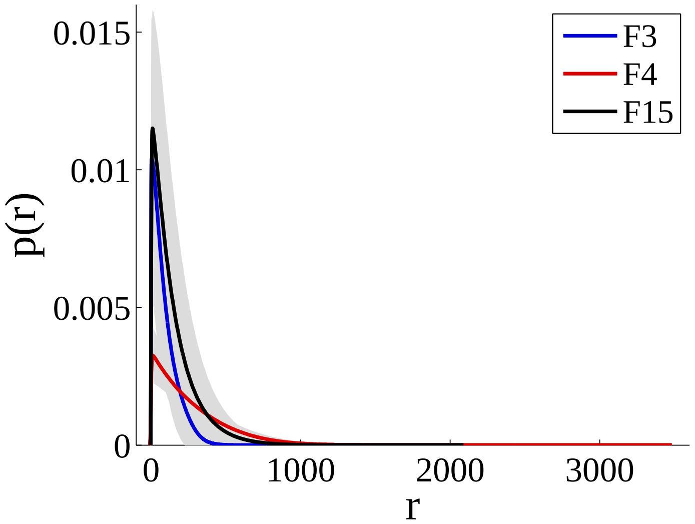


Figure 7.17: Rastrigin Functions in 20-D.

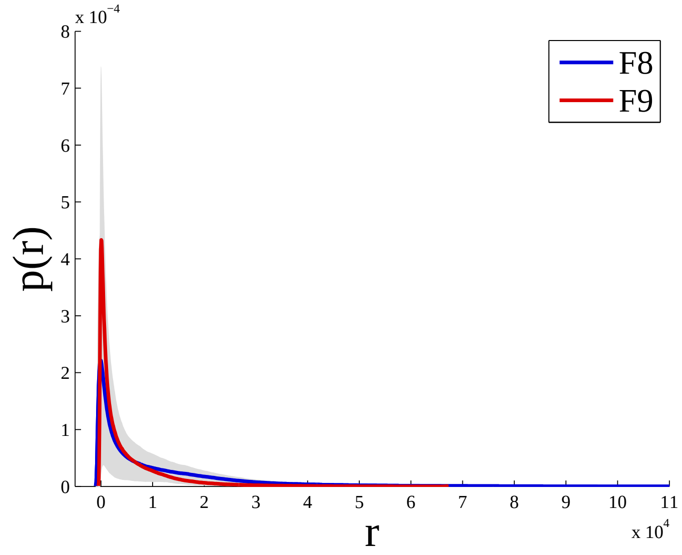


Figure 7.18: Rosenbrock Functions in 2-D.

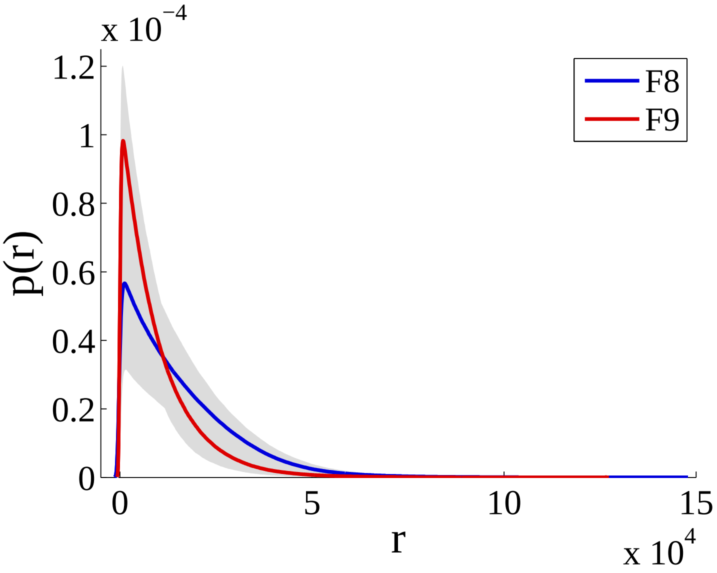


Figure 7.19: Rosenbrock Functions in 5-D.

thicker and the mode is lower. This indicates that there are an increasing amount of larger r values in F4 as D increases. The average J-divergences between F3 and F15 are approximately 15.5612 (2-D), 19.0032 (5-D), 9.6518 (10-D) and 2.6141 (20-D), the J-divergences between F3 and F4 are approximately 13.4518 (2-D), 23.7855 (5-D), 21.7836 (10-D) and 20.3402 (20-D), while the J-divergences between F4 and F15 are approximately 15.0893 (2-D), 24.4685 (5-D), 24.2549 (10-D) and 21.1355 (20-D).

The length scale distributions for F8 (Rosenbrock) and F9 (rotated Rosenbrock) are shown in Figures 7.18 to 7.21. The larger peak in the F9 distribution indicates that it has more low-valued length scales than F8. Both F8 and F9 are instances of Rosenbrock with a rotation about f. Consequently, the problems differ by a sample rotation of the search space, and so

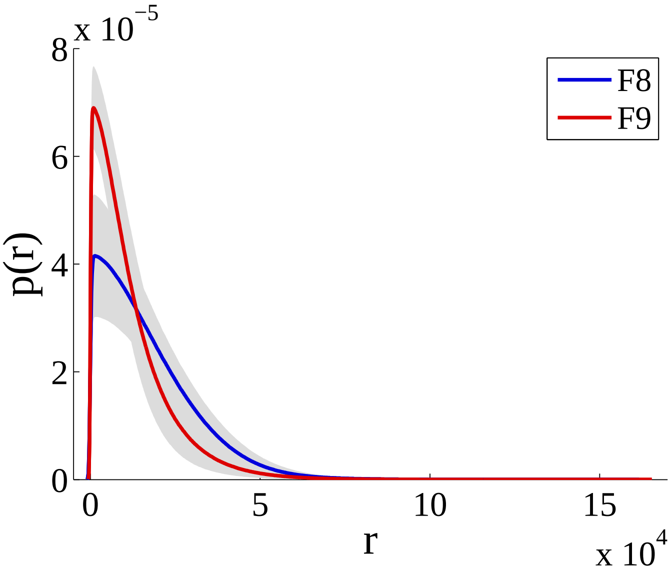


Figure 7.20: Rosenbrock Functions in 10-D.

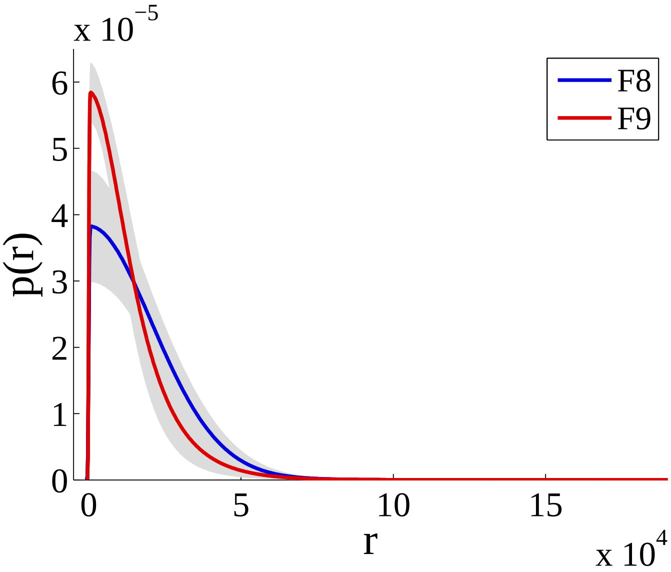


Figure 7.21: Rosenbrock Functions in 20-D.

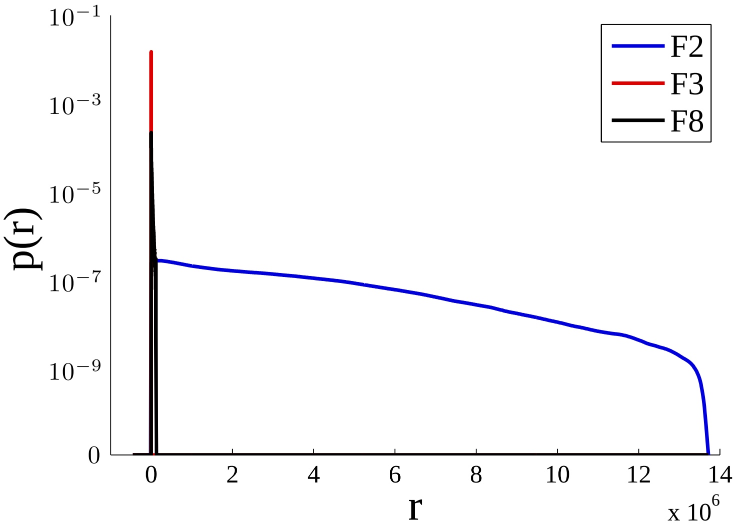


Figure 7.22: Ellipsoidal, Rastrigin and Rosenbrock Functions in 2-D.

the landscapes are structurally identical. Figures 7.18 to 7.21 clearly shows that the resulting length scales and their distributions are similar, thus empirically confirming that length scale is invariant to rotations of $\mathcal{S}$. The average J-divergences between F8 and F9 are approximately 12.8588 (2-D), 2.0693 (5-D), 0.7195 (10-D) and 0.4340 (20-D) indicating that the two problems are indeed almost identical. In comparison, the largest J-divergence out of all problems is 269.2115 between F4 and F19, which is significantly larger than the J-divergences between F2/F10, F3/F4/F15 and F8/F9.

The scaling on the r axis across Figures 7.10 to 7.21 also shows that length scales can be very different between two problems. For example, $r \in \left[0, 10^6\right]$ for the Ellipsoid problems, while $r \in [0, 5000]$ for the Rastrigin problems. To further illustrate the difference in scaling, the length scale distributions for F2, F3 and F8 in 2-D are shown in Figures 7.22 and 7.23.

It is clear that problems with similar structure have similar length scale distributions and low D_J between them, while problems with different structure have different length scale distributions and large D_J between them[1]. This demonstrates $p(r)$ and D_J as powerful tools for characterising and differentiating optimization problems. In addition, the distributions are surrounded by very thin shading, indicating that the standard deviation across samples is low.

[1] The largest J-divergence out of all BBOB problems is 269.2115 between F4 and F19, which is significantly larger than the J-divergences reported above.

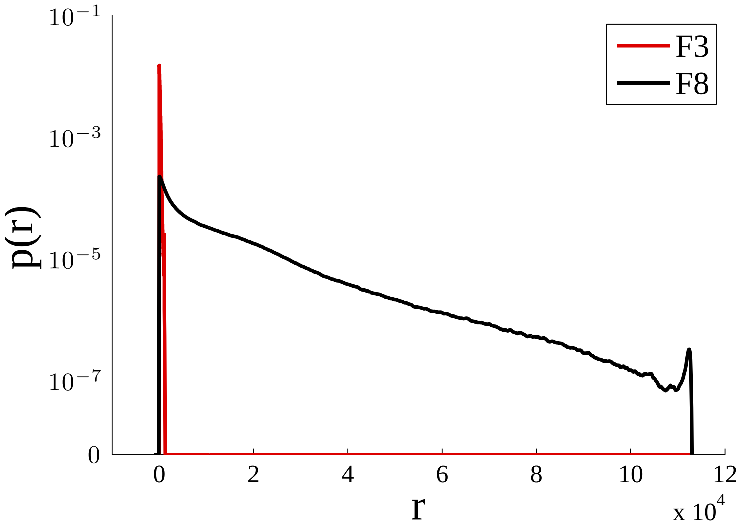


Figure 7.23: Rastrigin and Rosenbrock Functions in 2-D.

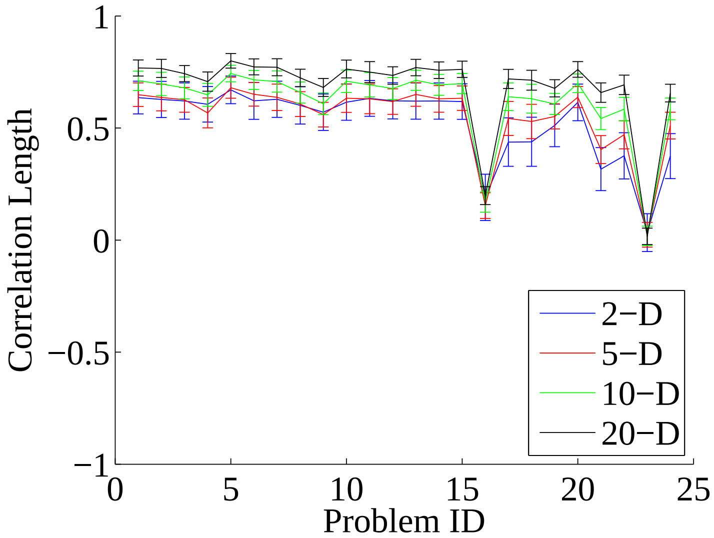


Figure 7.24: Correlation Length of the BBOB problem set in $D = 2, 5, 10$ and 20.

7.2.2 Results Comparing Length Scale to Existing Features

In this section, FDC, information content, partial information content, information stability and dispersion are evaluated for their ability to characterise and distinguish problems in the BBOB problem set. The features are also compared to the entropy of the length scale distribution, $h(r)$ (other summaries of r, such as the mean and variance, could similarly be calculated). Figures 7.24 to 7.31 displays the mean and one standard deviation (as error bars) for each feature across the 24 BBOB problems and D.

Correlation length, l, is intended to indicate the ruggedness of a landscape, and it specifically captures the maximum distance between solutions such that the correlation between objective function values is significant. The correlation lengths shown in Figure 7.24 are

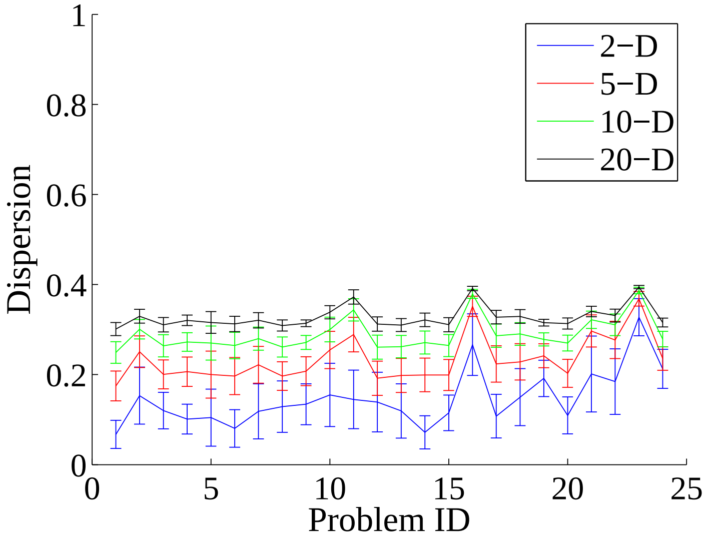


Figure 7.25: Dispersion of the BBOB problem set in $D = 2, 5, 10$ and 20.

positive and generally high across the BBOB problem set, and apart from problems F16, F21, F22 and F23, the correlation lengths are of very similar values. In addition, the correlation lengths vary by a small constant factor across D; as D increases, l increases by a constant (although the size of this constant decreases as D increases). The variations are likely caused by structural changes to the problems as D increases, and the degree to which the structures are being adequately sampled. The sample size used in these experiments increases with D, and so it is likely that as D increases, the structures required to distinguish problems at particular dimensions are not being adequately sampled. While most values of l are quite similar across D, F17, F18, F21 and F22 vary slightly. The standard deviation in correlation length is small, mostly around 0.05 (2-D F18 varies the most, with a standard deviation of 0.11), and decreases slightly as D increases for all problems.

High dispersion values indicate that the "good" solutions in the sample are well-separated and distributed throughout $\mathcal{S}$, thus implying a rugged landscape. Because bound-normalisation is used in conjunction with a Lévy random walk, the convergence of Dispersion values (previously discussed and illustrated in Chapter 4) to $\frac{1}{\sqrt{6}}$ should not occur. Since both l and dispersion aim to measure ruggedness, it is not surprising that the problems in Figure 7.25 with higher dispersion values mostly correspond to the problems in Figure 7.24 with low l. Both Figures 7.24 and 7.25 show F16 and F23 to be more rugged than the other problems. Evidently, the dispersions of F16 and F23 are also quite invariant to dimensionality. F1 to F15 are generally quite smooth compared to F16 to F23. In contrast to l, the dispersion varies more noticeably with D. Similar to correlation length, the variations are likely caused by structural changes and sampling variations as D increases. Like l, as D in-

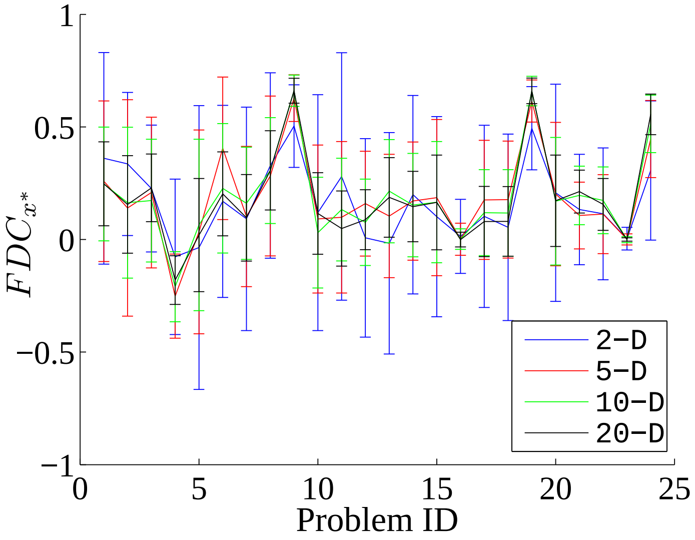


Figure 7.26: $FDC_{\mathbf{x}^*}$ of the BBOB problem set in $D = 2, 5, 10$ and 20.

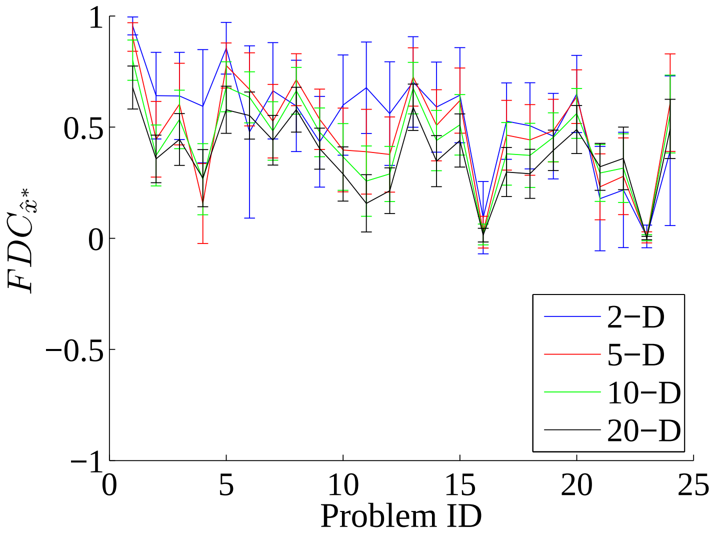


Figure 7.27: $FDC_{\hat{\mathbf{x}}^*}$ of the BBOB problem set in $D = 2, 5, 10$ and 20.

creases, the differences in dispersion between the problems becomes less pronounced (e.g. the range of dispersions for 2-D problems is 0.26, compared to 0.09 for 20-D), most likely caused by the limitations of sampling an exponentially-increasing search space. Overall, l and dispersion provide very limited ability to characterise and differentiate between the BBOB problems.

High FDC values indicate a strong correlation between the f-values of solutions and their distance from the global optimum. Figures 7.26 and 7.27 show considerably different values across the problem set. For some problems, the FDC variants actually indicate conflicting landscape characteristics, e.g. $FDC_{\mathbf{x}^*}$ indicates F4 is slightly deceptive (which it is), while this is not the case with $FDC_{\hat{\mathbf{x}}^*}$. This is an important result, as it demonstrates that $FDC_{\hat{\mathbf{x}}^*}$

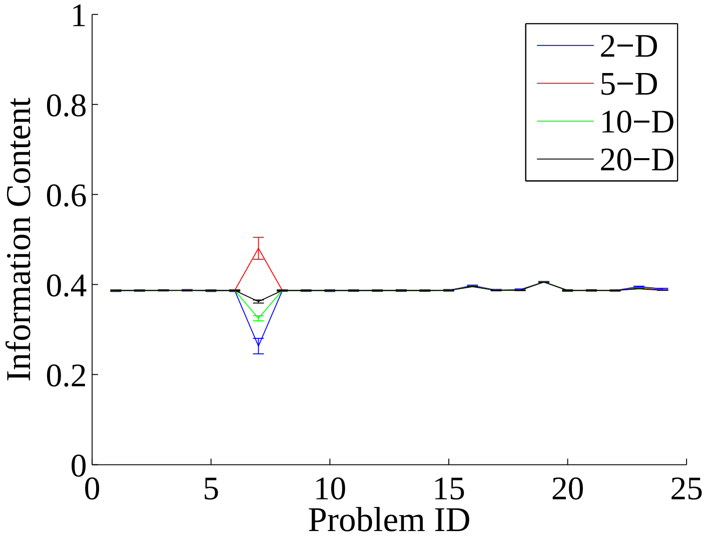


Figure 7.28: Information content of the BBOB problem set in $D = 2, 5, 10$ and 20.

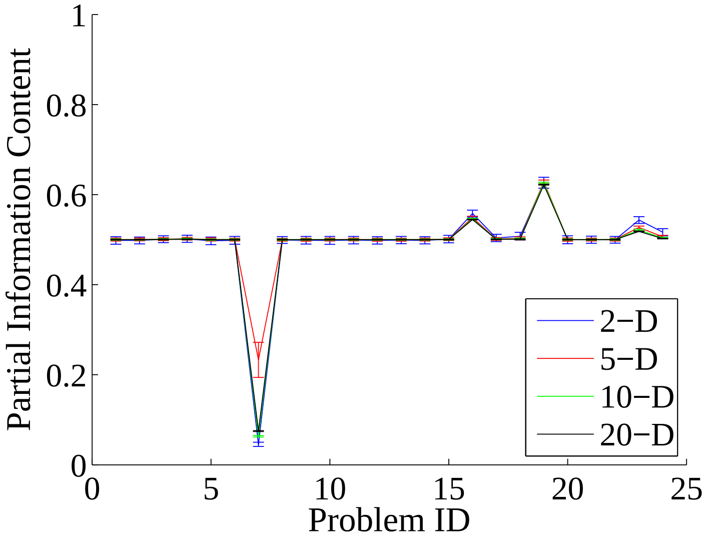


Figure 7.29: Partial information content of the BBOB problem set in $D = 2, 5, 10$ and 20.

is not always a reliable approximation of $FDC_{\mathbf{x}^*}$ and so conclusions based on the theory of FDC may be incorrect if drawn from $FDC_{\hat{\mathbf{x}}^*}$ results. F6 and F24 are also deceptive problems for some algorithms, however neither FDC variant were able to detect this. $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ are largely invariant across D. Across the problem set, $FDC_{\hat{\mathbf{x}}^*}$ is typically larger than $FDC_{\mathbf{x}^*}$, and $FDC_{\mathbf{x}^*}$ has a lot more variation between samples. $FDC_{\hat{\mathbf{x}}^*}$ exhibits similar trends to correlation length; F16, F21, F22 and F24 are far less correlated than the other problems. The similarity between $FDC_{\hat{\mathbf{x}}^*}$ and l is not surprising as they are both based on correlations among the objective function values of points within the sample.

Information content and partial information content are shown in Figures. 7.28 and 7.29 respectively, and it is clear that they are largely unable to differentiate the BBOB problems.

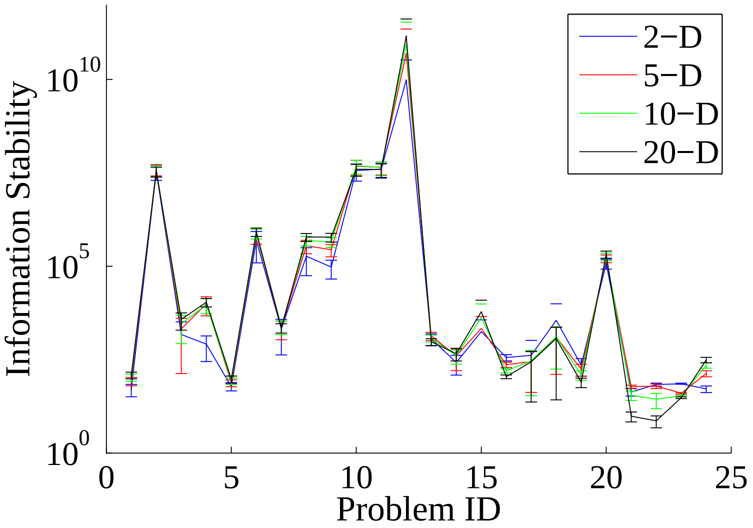


Figure 7.30: Information stability of the BBOB problem set in $D = 2, 5, 10$ and 20.

Information content measures the variety of fluctuations in f along the sample. A value of $log_6 2$ ($\approx$ 0.3869) indicates a highly rugged landscape with no neutral regions, and the results (erroneously) suggest that the BBOB problems are all highly rugged. Indeed, F1 and F5 are very smooth, and yet their information content values suggest otherwise. Partial information content indicates the degree of modality by measuring the variety of non-neutral regions in the sample, and the results in Figure 7.29 are very similar to information content. With the exception of F7, the partial information content is invariant to both the problems and D, and has very small variance between samples. F7 contains numerous neutral regions, which both information content and partial information content appear to have detected. Figure 7.29 and further analysis of F7's information content (not shown here) suggest that the 5-D problem contain significantly more "mode-like" structures than the 2-D, 10-D and 20-D problems. Subsequent results for $7 - D$, $9 - D$, $11 - D$ and $19 - D$ F7 problems showed similar behaviour to 5-D, suggesting that in general, F7 problems with odd-D are more multimodal than even-D. Figure 7.29 also indicates F17, F19, F23 and F24 are more multimodal than the other problems.

The information stability is the largest transition in objective function values encountered along the walk, i.e. $\max\left(\left|f(\mathbf{x}_i) - f(\mathbf{x}_j)\right|\right)$. Hence, it is conceptually very similar to $\max(r)$. Figure 7.30 and 7.31 show the information content and $h(r)$ respectively; both contain very similar trends, however there are some minor differences (e.g. $h(r)$ is more varied across D for F21 to F23). Both information stability and $h(r)$ are generally well-correlated with the conditioning of the problem; high conditioned problems, like F12, have high information stability and $h(r)$ values.

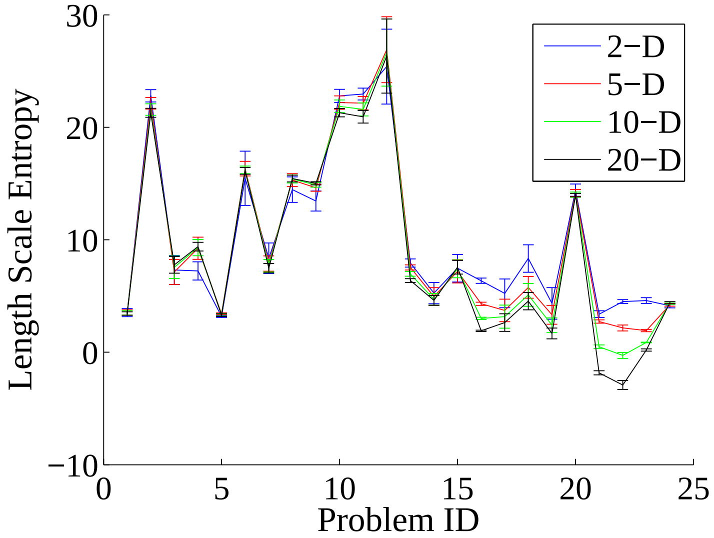


Figure 7.31: Length scale entropy of the BBOB problem set in $D = 2, 5, 10$ and 20.

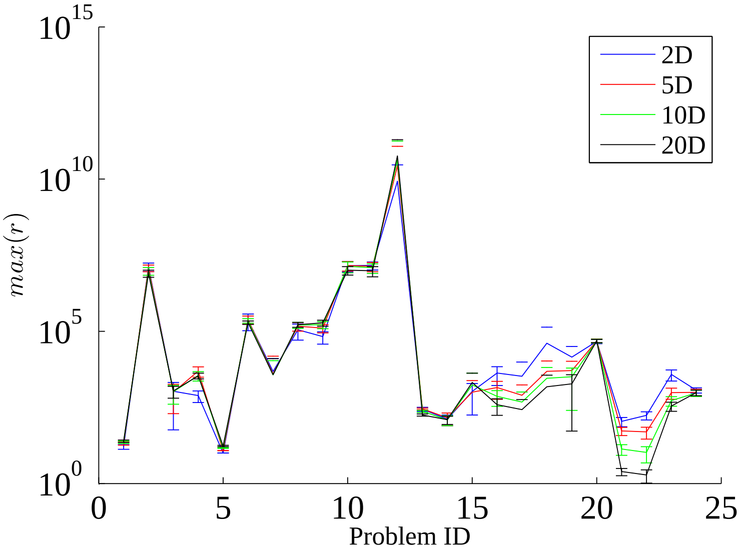


Figure 7.32: Maximum r of the BBOB problem set in $D = 2, 5, 10$ and 20.

The length scale entropy will vary depending on the distribution's shape and location. The magnitudes of the length scale values varies significantly across the BBOB problems, however the values generally yield unimodal, long-tailed distributions. Consequently, the similarity between $h(r)$ and information stability is likely due to the similarity in the shapes of the length scale distributions across the BBOB problems. As shown in Figures 7.32 to 7.35, statistical measures such as the maximum, median, mean and variance of r also exhibit similar trends to $h(r)$ and information stability. While the results here show similar trends in the length scale statistics, length scale distributions with different shapes and locations can produce different trends.

Encouragingly, information stability and the length scale statistics exhibit a strong ability

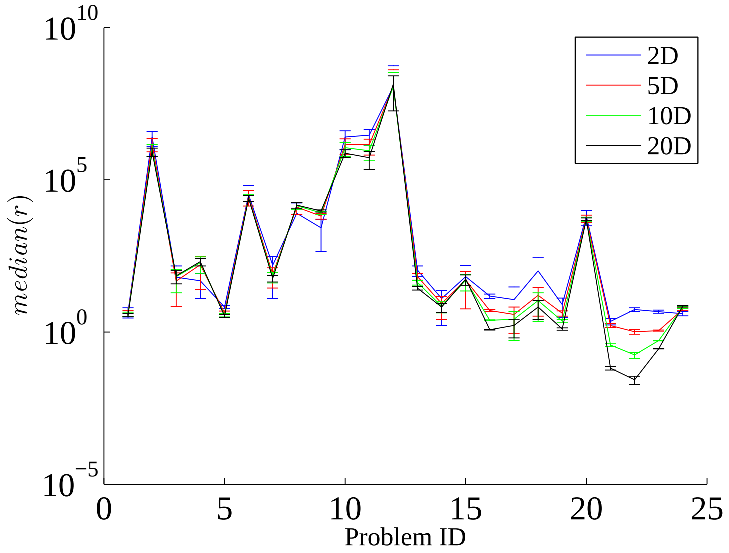


Figure 7.33: Median of r of the BBOB problem set in $D = 2, 5, 10$ and 20.

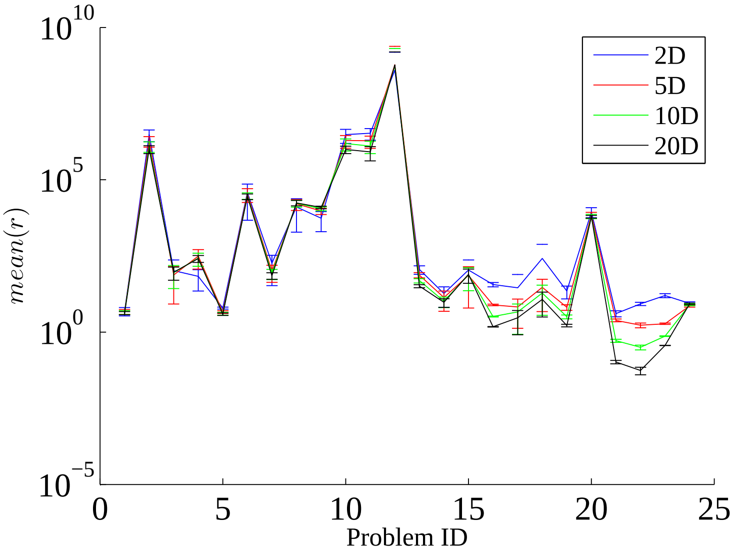


Figure 7.34: Mean of r of the BBOB problem set in $D = 2, 5, 10$ and 20.

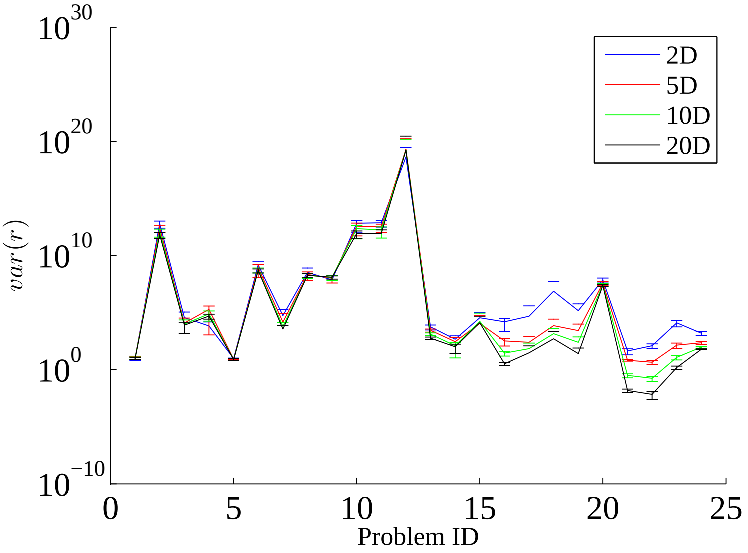


Figure 7.35: Variance of r of the BBOB problem set in $D = 2, 5, 10$ and 20.

to differentiate the BBOB problems, are invariant to D and show very little variance between samples. Thus in terms of characterising the BBOB problems, information stability and the length scale statistics are clearly superior to the other landscape features analysed.

To summarise, the features compared in these experiments show some ability to capture landscape structure, however, most features are unable to be used to detect the known differences across the BBOB problem set. In terms of characterising and distinguishing problems, the existing features seem limited. There were two notable exceptions, information stability and the length scale statistics, that produced a wide range of consistent, reliable values with clear relationships to the problems.

7.2.3 Results Comparing Length Scale with an Ensemble of Features

The length scale analysis in Section 7.2.1 was clearly able to characterise and distinguish the BBOB problems, while many of the existing features in Section 7.2.2 struggled. Collectively, the existing features may offer a greater ability to characterise and distinguish problems [77, 164], and so this experiment evaluates the ability of an *ensemble* of these existing features to characterise problems, and compares this to the length scale analysis. Given that the features were largely invariant to D across the BBOB problems analysed above, the following analysis focuses on 20-D BBOB problems.

For the feature-ensemble approach, each problem is represented by a 7-D feature-vector consisting of the correlation length, dispersion, $FDC_{\mathbf{x}^*}$, $FDC_{\hat{\mathbf{x}}^*}$, information content, partial information content and information stability, averaged across the seeds/walks. The features are normalised by their appropriate bounds, and because information stability is unbounded, it is normalised by the range of information stability values obtained across the problems. Each of the 20-D BBOB problems are thus represented by a 7-D feature vector that can be further analysed via clustering and visualisation techniques.

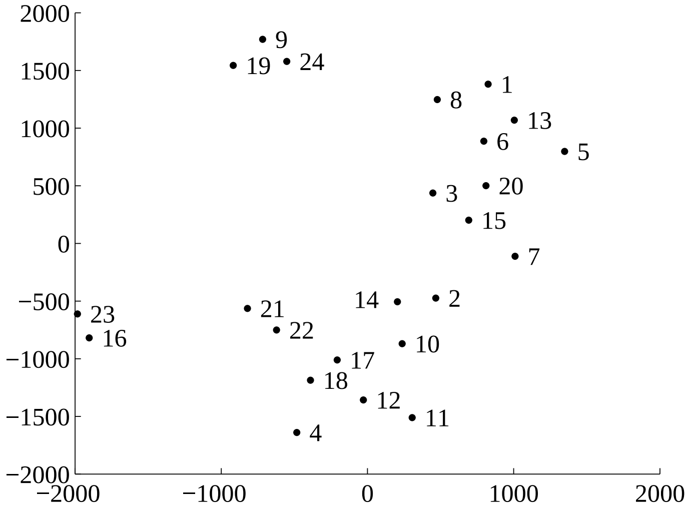
As discussed in Section 6.1.3, the J-divergence between two length scale distributions is a measure of problem similarity. Hence the J-divergences between problems can be used to *implicitly* define the problem space. That is, the J-divergences can be used to infer certain properties of the problem space, without knowledge of the explicit locations of problems within the space. In contrast to the feature-ensemble approach, this approach does not explicitly define, and hence constrain, the problem space.

t-SNE [186] is again used to visualise the problem spaces resulting from the length scale and the feature-ensemble approaches. In order to apply t-SNE, the average J-divergences between the 24 BBOB problems (across the different walks/seeds) were used to calculate a 24×24 dissimilarity matrix. To apply t-SNE with the feature-ensemble approach, a 24×24 distance matrix was generated by calculating the Euclidean distance between problems' feature-vectors. Based on the recommendations in [186] and exploratory experimentation, the perplexity was set to 5 for all visualisations in these experiments. Similarly to the previous experiments on artificial problems, 1000 different trials were conducted, with a maximum of 1000 iterations for each trial. Figure 7.36 shows the best t-SNE visualisation (in terms of the final cost) from the trials for each approach. The cost of a t-SNE solution indicates the discrepancy between the neighbourhood relationships in the two-dimensional visualisation and the original high-dimensional data, and were quite consistent across the 1000 trials. Specifically, the costs of the feature-ensemble approach ranged between 0.1064 and 0.5522 with a median cost of 0.1314, while the costs of the length scale approach ranged between 0.1979 and 0.3244 with a median cost of 0.2034.

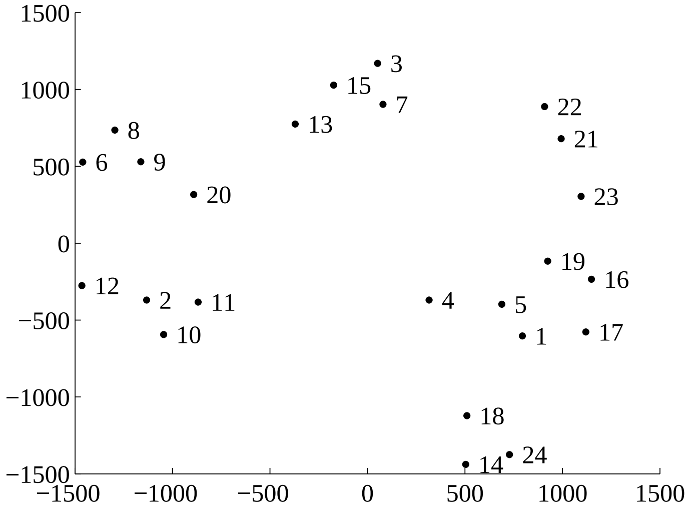
While the feature-ensemble approach shown in Figure 7.36a has lower error, the relationships between structurally similar problems according to the known BBOB function properties are not evident, and overall, it is less discriminating between problems. For example F8 (Rosenbrock) and F9 (Rotated Rosenbrock) are well separated in the space, despite the problems differing only by rotation. In contrast, Figure 7.36b shows that the length scale analysis better reflects the known relationships between BBOB functions; not only are F8 and F9 close, but so are F2 and F10 (ellipsoidal problems) as well as F3 and F15 (Rastrigin problems).

To analyse cluster structure in the data, hierarchical clustering was applied to the J-divergence matrix and the feature-vector distance matrix using unweighted average distance linkages. The resulting dendrograms for the 20-D BBOB problems can be seen in Figure 7.37. The clusters yielded from both hierarchical clustering correspond well with the clusters in the visualisations produced by t-SNE. However, the clusters and relationships suggested by the t-SNE visualisations and dendrograms do not correspond to the problem categories in the original BBOB specification. The BBOB problems and categories were defined by researchers with the aim of providing a wide variety of test problems with different landscape structures. This was done using intuition in two and three dimensions, modifying previously proposed problems and experience with algorithms. Hence, the defined-categories might not truly reflect the underlying structures within the problems.

The feature-ensemble approach shown in Figure 7.37a lacks strong clusters, although there are perhaps three weak clusters (F7, F16/F23 and the remaining problems). While

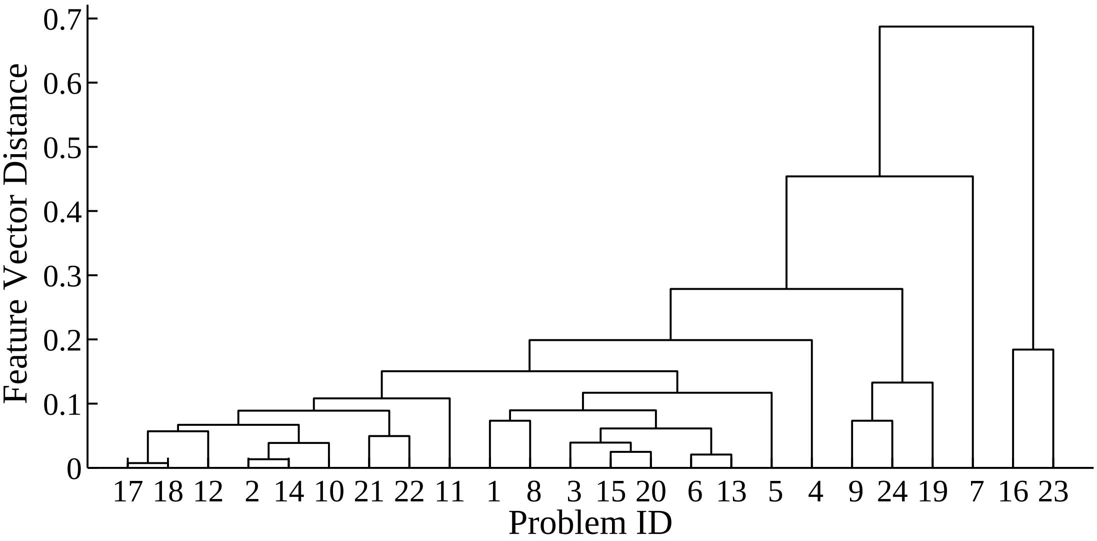


(a) Feature-ensemble approach

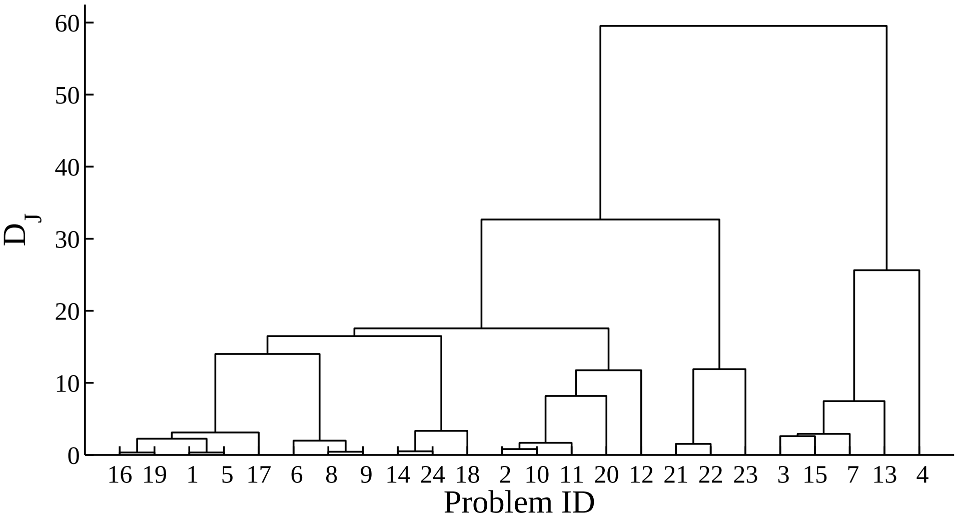


(b) Length scale approach

Figure 7.36: Feature spaces of 20-D BBOB problems reduced via t-SNE.



(a) Feature-ensemble approach



(b) Length scale approach

Figure 7.37: Dendrograms of the 20-D BBOB problems.

many similar problems are close in proximity (e.g. F3/F15, F9/F19, F17/F18 and F21/F22), many structurally-dissimilar problems are also close in proximity (e.g. F1/F8, F3/F20, F12/F18, F24/F9). In contrast, the J-divergence of length scales (Figure 7.37b) clusters the problems into many different classes that correspond well with the underlying structures of the problems. For example, F8 and F9 are together, as are the elliptically-structured F2, F10 and F11, and the Gaussian-constructed F21 and F22. There are some exceptions; F17 and F18 are separated despite both being Schaffer F7 functions (F18 is moderately ill-conditioned). Compared to the feature-ensemble approach, the length scale approach appears to be more indicative of the defined BBOB problem structures.

7.2.4 Relationship to Problem Difficulty

Landscape properties are often related to algorithm performance in an attempt to explain why certain algorithms perform well on one problem, but poorly on another. The BBOB problem set not only provides practitioners with an opportunity to compare their algorithms with other, state-of-the-art techniques, but results can yield insights into the behaviour of algorithms on the problem landscape structures. Consequently, there is a copious amount of published and publicly available algorithm performance and trajectory data for the BBOB problems. This provides an exciting research opportunity, as the landscape properties measured above can be directly compared to algorithm performance.

The purpose of this experiment is to evaluate how well length scale measures and existing measures correlate with the difficulty of BBOB problems. Defining what constitutes a "difficult" problem is non-trivial; as discussed in Section 3.1, a problem that is difficult for one algorithm may be easy for another. In these experiments, the difficulty of a problem is measured by the performance of the *best* performing algorithm. The performance of an algorithm on a given problem is measured by its Expected Running Time (ERT) [71]:

$$ERT(f_{precision}) = t_{success} + \frac{1 - p_{success}}{p_{success}} t_{fail} \tag{7.2.1}$$

where $t_{success}$ and t_{fail} are the average number of function evaluations for successful and failed trials respectively, and $p_{success}$ denotes the number of successful trials. It can be equivalently written as:

$$ERT(f_{precision}) = \frac{\#FEs \;\; while \;\; |f(\mathbf{x}^*) - f(\hat{\mathbf{x}}^*)| \geq f_{precision}}{\#success} \tag{7.2.2}$$

where *#FEs* is the number of function evaluations conducted in all trials and *#success* is the number of successful trials. For example, if a minimisation algorithm successfully finds $\mathbf{x}^*$ for 20 out of 30 trials, and throughout all trials a total of 1500 function evaluations are made while the current (at the time of the objective function evaluation) best solution, $\hat{\mathbf{x}}^*$, has worse f-value than $f(\mathbf{x}^*) + f_{precision}$, then the resulting ERT is $\frac{1500}{20}$.

The BBOB developers advocate ERT as it allows comparisons of performance profiles from differing dimensionality, search spaces, global optimum values and so on [71]. However, ERT is parameterised by $f_{precision}$, which specifies how close the algorithm must get to the global optimum. For the purpose of analysing difficulty, this is a non-trivial choice to make and can contribute to the perceived difficulty of the problem. For example, a large ERT calculated with large precision (e.g. 1) indicates that the algorithm is far away from the global optimum, and hence the problem is difficult. However, using a large ERT may not be very informative; algorithms that can quickly find the area of the global solution will appear to do well (and hence yield a small ERT), despite the fact that they may not be able to converge on the global solution. Conversely, using a small ERT may be too restrictive and harsh on the algorithm. In this scenario, algorithms that are able to get close to the global optimum quickly may still have a large ERT because they take time to converge. Hence, the choice in precision is an explicit choice in the trade-off between deeming a problem hard because the global area is hard to find and/or the global optimum is hard to converge on.

With these considerations in mind, the ERT (with a precision of 10^{-8}) of the *best* performing algorithm is used in the following experiments as a proxy for problem difficulty. The high precision will yield a wider range of ERT values than lower precisions.

Algorithm performance results from previous BBOB competitions are available from `http://coco.gforge.inria.fr/doku.php`. The ERTs for all algorithms participating in the 2009, 2010, 2012 and 2013 competitions are used in these experiments.

To investigate the relationship between problem difficulty and landscape features, the landscape features analysed in Section 7.2.2 are directly compared to the ERT values. Pearson's correlation coefficient, ρ_p, was used (with a significance level of 0.01) to quantify the correlation between the landscape features and problem difficulty. More specifically, for a given feature (e.g. FDC), there are 900 estimates of the feature for each of the 24 problems in each dimension (e.g. 2-D Sphere). Hence there are a total of 900×24 pairs of feature/ERT values for a given dimension. Table 7.1 contains the Pearson correlation coefficient between these feature estimates and ERT values, separated by dimensionality. Each correlation is tested against the hypothesis that there is no correlation, and the resulting p-values are re-

Feature	**2-D**	**5-D**	**10-D**	**20-D**
Correlation Length	-0.2073	-0.1625	-0.0431	-0.0456
Dispersion	0.1756	0.0997	-0.0503	-0.0890
$FDC_{\mathbf{x}^*}$	0.0489	0.2717	0.2439	0.3538
$FDC_{\hat{\mathbf{x}}^*}$	-0.1056	0.0152 (0.03)	0.1123	0.1176
Information Content	0.0681	0.0517	0.0422	0.0776
Partial Information Content	0.0654	0.3235	0.0416	0.0667
Information Stability	-0.0207	-0.0210	-0.0150 (0.03)	-0.0241
Length Scale Entropy	-0.2063	-0.2904	-0.1399	-0.1409

Table 7.1: Pearson's correlation coefficient, ρ_p, between BBOB ERT values and problem metrics.

ported in brackets when greater than 0.01.

All features in Table 7.1 have relatively low correlation coefficients, with information content, partial information content and information stability essentially uncorrelated with ERT. The largest correlation was only 0.3538 for $FDC_{\mathbf{x}^*}$ on 20-D problems. Interestingly, $FDC_{\mathbf{x}^*}$ was rather uncorrelated in low dimensions, and as dimensionality increased, correlation also increased. This trend was also observed for $FDC_{\hat{\mathbf{x}}^*}$. The correlation length feature is slightly negatively correlated with ERT for 2-D problems ($\rho_p = -0.2073$), however as dimensionality increases, the correlation diminishes ($\rho_p = -0.0456$ in 20-D). Perhaps the most consistent feature with correlation across dimensionality is the length scale entropy, which ranges between -0.2904 (5-D) and -0.1399 (10-D). Overall, while there are some correlations between the features and ERT, these correlations are not very strong and vary considerably with dimensionality.

The correlation coefficients in Table 7.1 indicate that there is a lack of linear relationship between the features and ERT, however a non-linear, more complex relationship may still exist. To investigate if this is the case, each feature was averaged across the 900 samples and compared to the problem's difficulty (i.e. ERT) using scatter plots, shown in Figures 7.38 to 7.45.

With the exception of information content and partial information content, the features in Figure 7.38 to 7.45 have a complex relationship with ERT. As none of the figures have a clear trend (linearly or non-linearly), there is no obvious, direct relationship between the features and problem difficulty. However, some of the figures show slight trends and clear clusters of problems, which indicates a complex relationship between the features and prob-

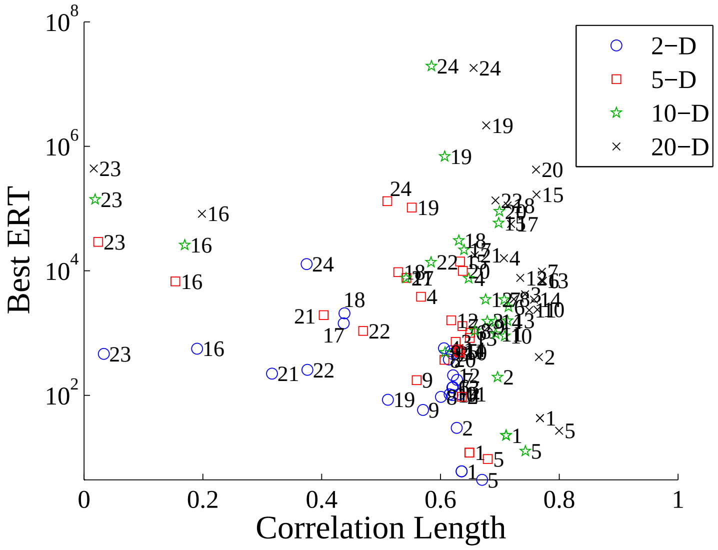


Figure 7.38: Relationship between correlation length and the best ERT.

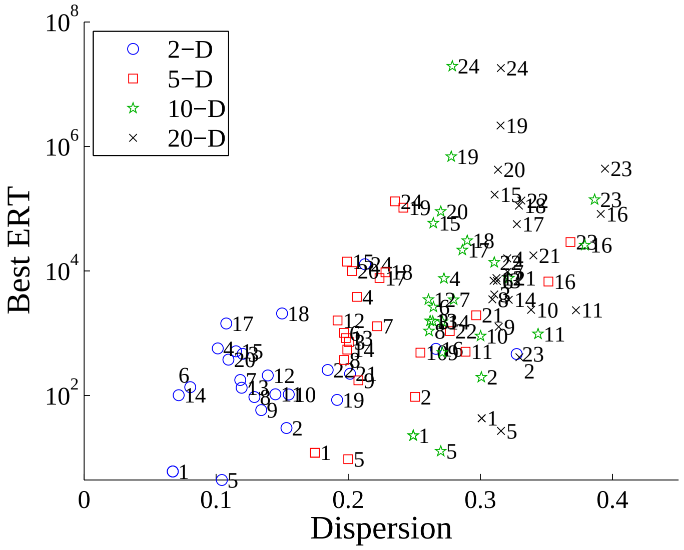


Figure 7.39: Relationship between dispersion and the best ERT.

lem difficulty. Problems are generally spread throughout the best ERT, with lower dimensional problems having lower ERT values, and high dimensional problems having high ERT values. While this reflects the notion that higher dimensional problems are generally more difficult to solve, there are some exceptions. Two exceptions are the $F1$ (Sphere) and $F5$ (Linear Slope) problems, which have low ERT values with almost no regard to D.

The relationship between correlation length and the best ERT, shown in Figure 7.38 is rather complex; there appears to be no correlation between the feature and ERT, however some problems cluster together based on their type. For example, $F1$, $F2$, $F5$, $F9$, $F16$ and $F23$ appear near each other, despite differences in D.

The combination of dispersion and ERT, shown in Figure 7.39, further clusters the BBOB

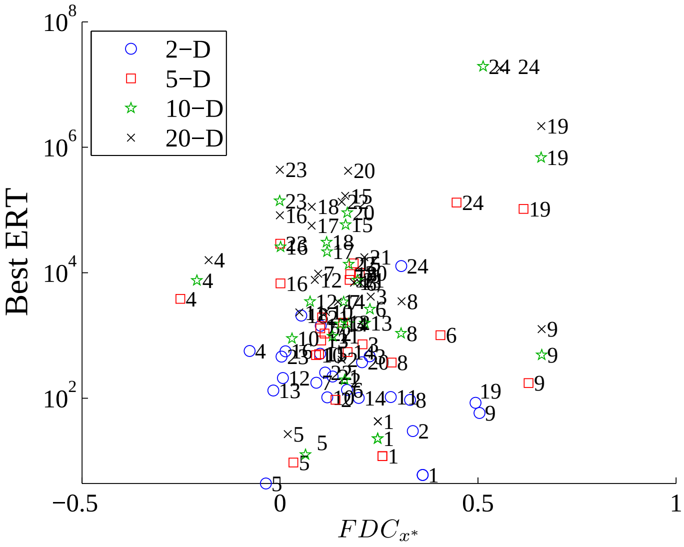


Figure 7.40: Relationship between $FDC_{\mathbf{x}^*}$ and the best ERT.

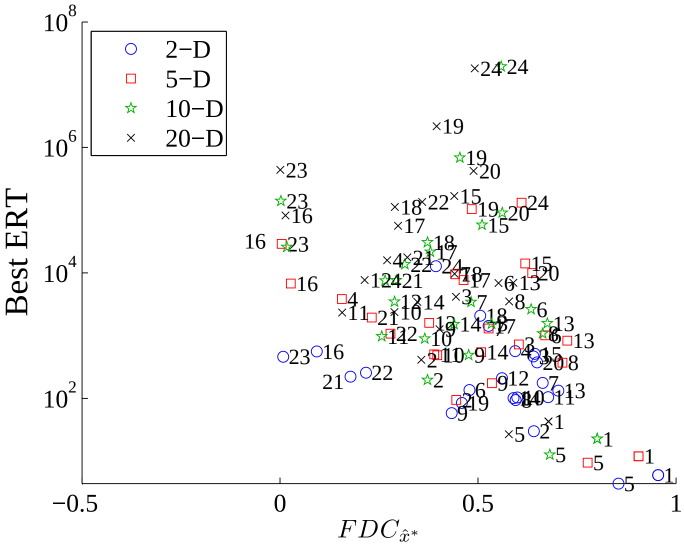


Figure 7.41: Relationship between $FDC_{\hat{\mathbf{x}}^*}$ and the best ERT.

problems into their respective dimensionality's. In general, the value of ERT generally increases as the value of dispersion increases. Looking closely at the problems for each D in isolation, the *same* trends are exhibited. For example, $F1$ and $F5$ are always found towards the lower left hand side of the dimension's cluster; while $F24$ is always found at the top right. The results for correlation length, information stability and length scale also show moderate segregation by D, but it is not as pronounced as the discrimination in Figure 7.39. Overall, dispersion has a slightly positive correlation with the best ERT, and the combination of dispersion and ERT can be used to discriminate problems by their dimensionality.

In contrast to dispersion, $FDC_{\hat{\mathbf{x}}^*}$, shown in Figure 7.41 has a slight negative correlation with the best ERT, and the problems are well-mixed with respect to D. The problems also

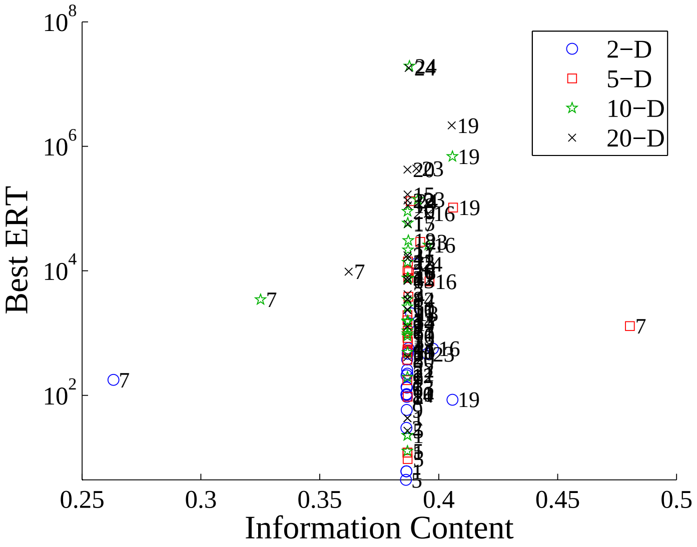


Figure 7.42: Relationship between information content and the best ERT.

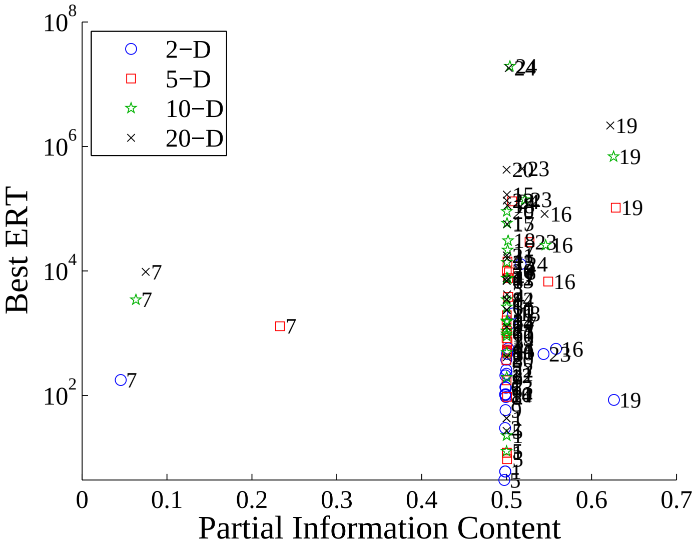


Figure 7.43: Relationship between partial information content and the best ERT.

seem to be positioned similarly with respect to D; $F1$ and $F5$ are at the bottom-right of the single cluster, and $F16$ and $F23$ are positioned at the top-left. The problems in Figure 7.41 are much more spread out than the $FDC_{\mathbf{x}^*}$ vs best ERT problems in Figure 7.40. Here, the problems are distributed throughout $FDC_{\mathbf{x}^*}$ in a much smaller range, however $F4$, $F19$ and $F24$ are exceptions. Figure 7.40 does not show any clear trends, although problems $F1$, $F4$, $F5$, $F19$ and $F24$ are quite well-separated from the single major cluster.

Figure 7.45 shows that, like correlation length, there is no discernible correlation between length scale entropy and the best ERT value. A few of the problems are also quite well-spread through the space, with quite different clusters than correlation length. For example, $F6$, $F8$, $F9$, $F10$, $F11$, $F12$, and $F20$ are all quite separated. As previously shown in Sec-

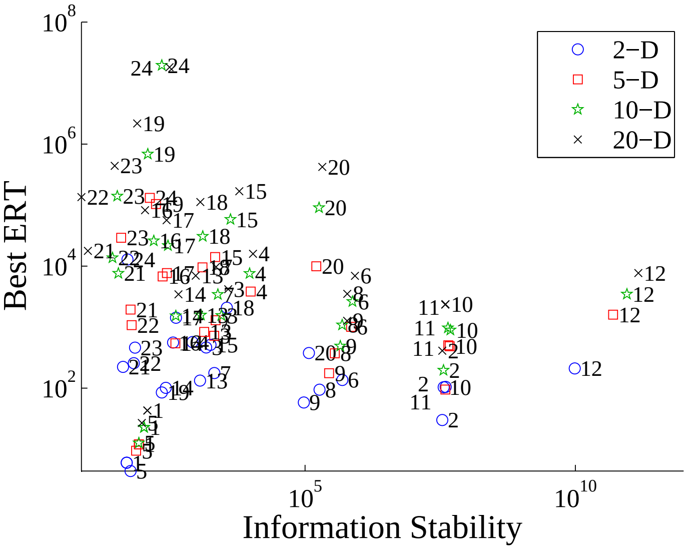


Figure 7.44: Relationship between information stability and the best ERT.

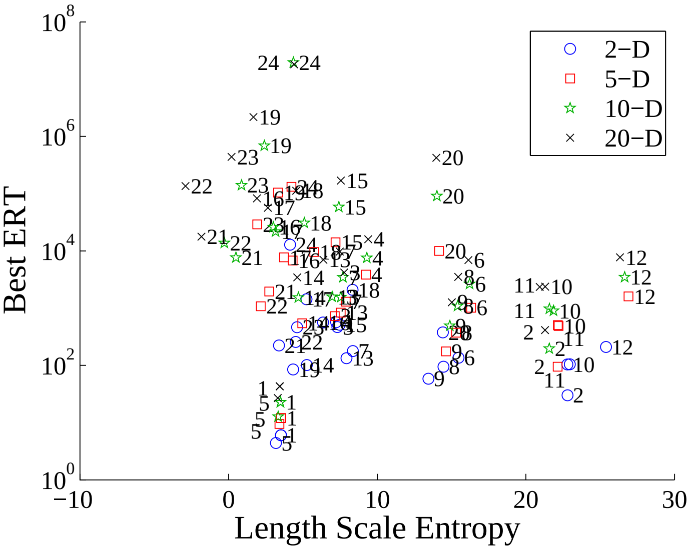


Figure 7.45: Relationship between length scale and the best ERT.

tion 7.2.2, there is a strong similarity between length scale entropy and information stability, and this is reflected in Figures 7.44 and 7.45. Again, $F6$, $F8$, $F9$, $F10$, $F11$, $F12$ and $F20$ are separated, and there is no clear relationship between information stability and the best ERT.

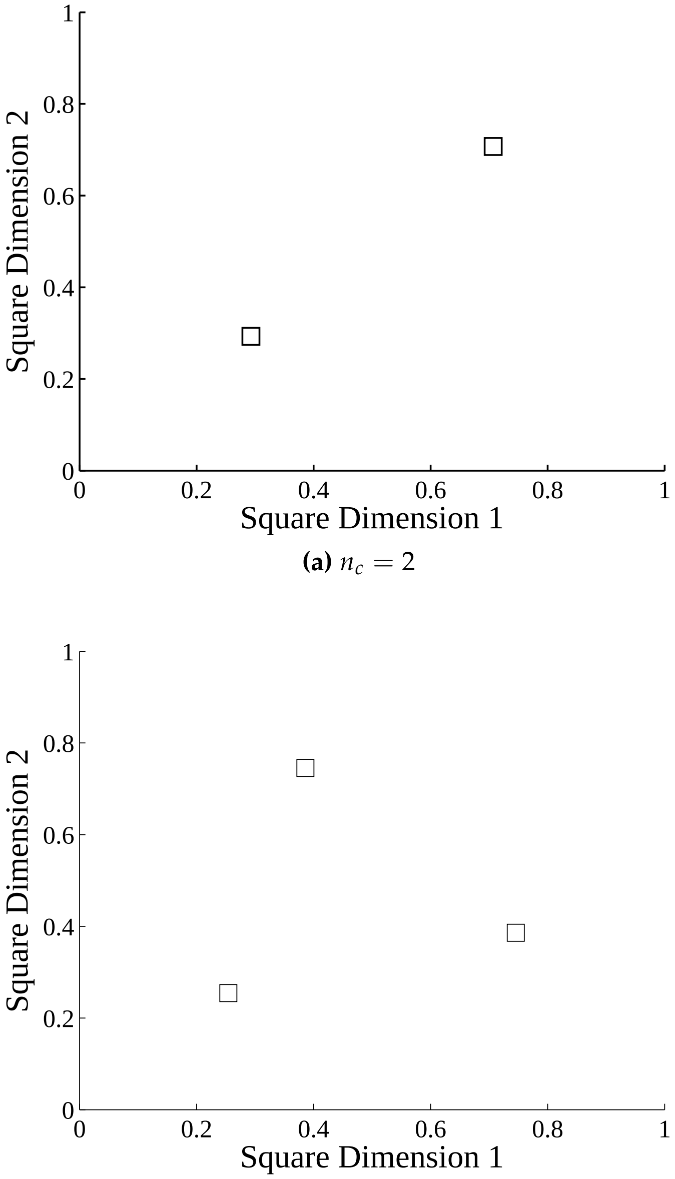
Overall, neither the problem features nor ERT can adequately characterise the problems on their own, however, when used in combination, interesting insights can be drawn. Due to the differences in the problems segregated, the *combination* of correlation length, length scale entropy and ERT will likely produce better problem discrimination. While information content and partial information content were very limited, the remaining techniques are often able to cluster a small subset of the problems. The ability to identify particular subsets of problems indicates that perhaps none of the features can capture *all* landscape structures required to distinguish and discriminate individual problems, but rather, each feature specialises in capturing a specific structural property (inherent to the problems clustered). This is clearly related to feature ensembles in classification, where combinations of features generally have more discriminatory power together than alone [4].

7.3 Analysis of Circle in a Square Problems

The purpose of this experiment is to analyse the landscapes of Circle in a Square (CiaS) packing problems (see Appendix A.1.3) and evaluate the ability of existing landscape features and the length scale analysis to (robustly) characterise these problems. In particular, the experiments investigate how well correlation length, dispersion, FDC, information content, partial information content, information stability and the entropy of the length scale distribution ($h(r)$) characterise the packing problem for an increasing number of circles.

CiaS packing problems represent a challenging class of optimization problems. In general, they cannot be solved using analytical approaches or via gradient-based mathematical optimization. These problems are also believed to generally contain an extremely large number of local optima. For the related problem of packing equal circles into a larger circular region, Grosso et al. [67] use a computational approach to estimating the number of local optima by repeatedly running a local search algorithm over a large number of trials. Although a conservative estimate, this indicates that the number of local optima grows unevenly but steadily, with at least 4000 local optima for packing 25 circles and more than 16000 local optima for packing 40 circles.

CiaS problems are parameterised by the number of circles, n_c, and little is known about how n_c affects the structure of the problem. By considering the optimum solution, the (op-



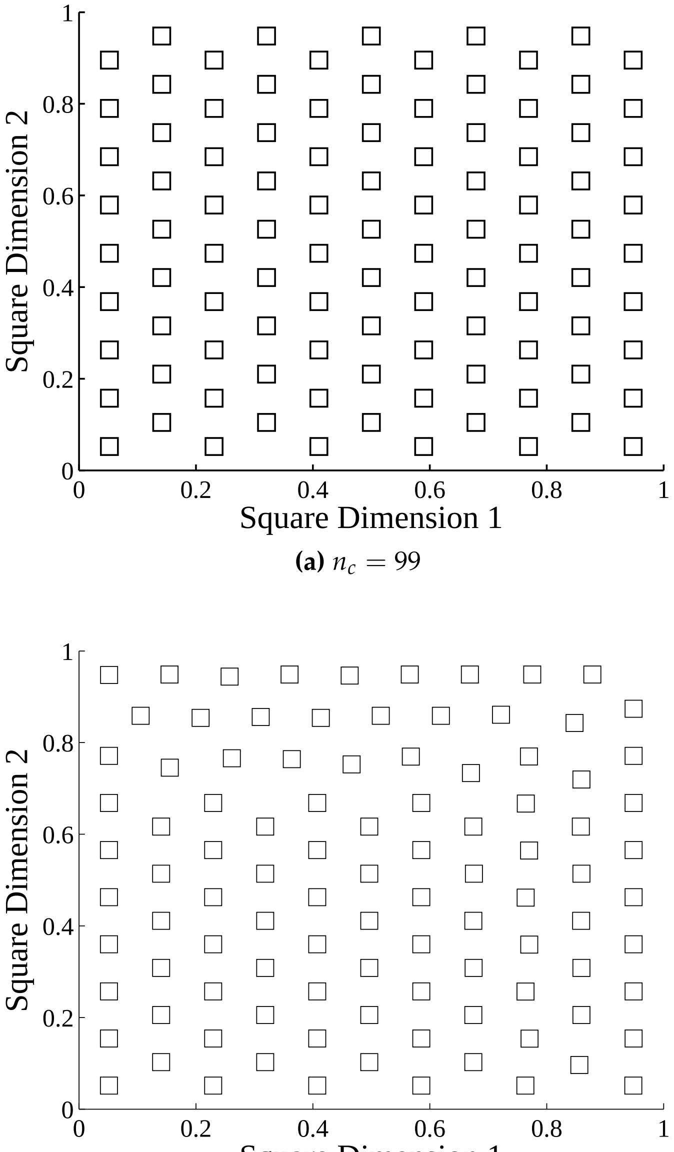
(a) $n_c = 2$

(b) $n_c = 3$

Figure 7.46: Circle centres of the optimal Circle in a Square packings for similar n_C that show very different arrangements of circle centres.

timal) arrangement of 2 circle centres in a square intuitively seems very different compared to 3 circle centres (see Figure 7.46), and yet the arrangement of 99 circle centres seems very similar to 100 circle centres (see Figure 7.47). While the illustrations in Figures 7.46 and 7.47 only show the circle centres of the optimal packings, it does give a sense of the way in which the CiaS problem scales with n_c.

Given a solution of n_c circles, the ordering of the circles in the solution vector may be permuted without affecting the objective function value of the solution. Hence, for any given solution, there are $n_c!$ equivalent solution vectors. Generating $n_c!$ equivalent solutions for each solution in the sample is obviously computationally infeasible for large n_c, and so



(a) $n_c = 99$

(b) $n_c = 100$

Figure 7.47: Circle centres of the optimal Circle in a Square packings for similar n_C that show very similar arrangements of circle centres.

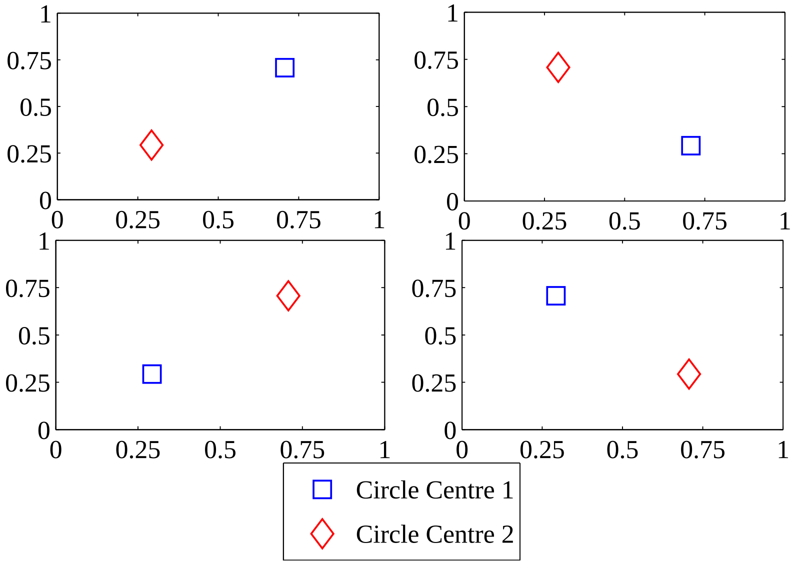


Figure 7.48: Four globally optimal packings for $n_c = 2$. The centres of each circle are shown, where circles 1 and 2 and represented by □ and ◇ respectively.

the issue of permutation is ignored for these experiments. This is a reasonable design choice as algorithms are unlikely to generate permuted solutions, and so the landscapes analysed here are the landscapes an algorithm would typically search. Furthermore, in the black-box scenario, information such as this is not known a priori.

The correlation length, dispersion, information content, partial information content, information stability and entropy of the length scale distribution are all calculated based on the settings from the BBOB experiments (described in Section 7.2). FDC is again calculated using the best known solution [168], as well as the best solution in the sample, as reference points. Because some solutions may be rotated and/or reflected in the 2-D packing space without affecting their objective function value, many of the packings have multiple global optima. For example, Figure 7.48 illustrates the 2-D packings of the 4 equivalent global optima for $n_c = 2$. For packings with multiple global optima, $FDC_{\mathbf{x}^*}$ is estimated based on the distance between solutions and their closest global optimum.

Each feature is calculated from a sample of solutions resulting from a Lévy random walk of $1000 \times 2n_c$ steps in $\mathcal{S} = [0,1]^{2n_c}$, where n_c is the number of two dimensional circles being packed. To provide a large sample of problems, 2 to 100 circles are analysed, and Euclidean distance is used as the distance metric between solutions. The robustness of each feature is assessed by examining the variation between different samples. For each problem, 30 different samples (of size $1000 \times 2n_c$) of $\mathcal{S}$ are generated, resulting in 30 estimates of a given feature. Results are reported using the mean of the 30 estimates for each feature. Error bars on the figures indicate one standard deviation over the 30 estimates.

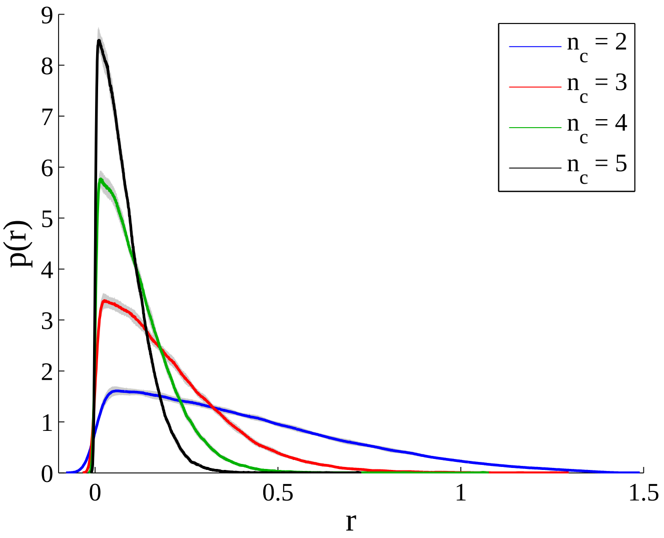


Figure 7.49: Length scale distributions for $n_c = 2, 3, 4$ and 5.

7.3.1 Length Scale Distribution Results

Because each problem is sampled 30 times, 30 length scale distributions are estimated for each problem. To visualise the $p(r)$s, a single representative distribution is constructed and shown (based on the average probabilities for each r), with grey shading indicating 1 standard deviation of the probabilities at the given r.

The length scale distributions for $n_c = 2, 3, 4$ and 5, shown in Figure 7.49, are all smooth, left-skewed unimodal distributions with little variation between the 30 samples. Clearly, as n_c increases, the range of length scales decreases, and smaller length scales occur with greater probability. Figure 7.50 shows the length scale distributions for $n_c = 97, 98, 99$ and 100. In contrast to the small values of n_c, the large values of n_c have very similar distributions. The scaling on the r axis across Figures 7.49 and 7.50 illustrates that length scales can vary significantly between problems. In particular, the scale of the r values in Figure 7.50 is very small, indicating that the change in objective function value observed relative to the size of the change in solution is very small. This is likely to be an artifact of the nature of the objective function. The objective function is based on the *maximum* of the minimum distance between any two circle centres. Assuming the circles are distributed with roughly even coverage, increasing the number of circles will decrease the minimum distance between any two circles. Hence the overall decrease in magnitude of the r values is likely due to the decrease in magnitude of f.

Figures 7.49 and 7.50 also show that as n_c increases, the mode decreases and the tail thins. To quantify the change in shape of $p(r)$, the ratio of mode and 99th percentile are shown

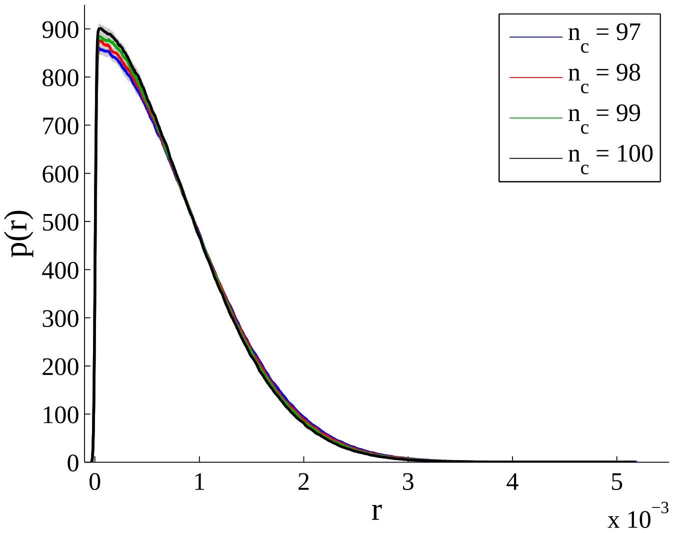


Figure 7.50: Length scale distributions for $n_c = 97, 98, 99$ and 100.

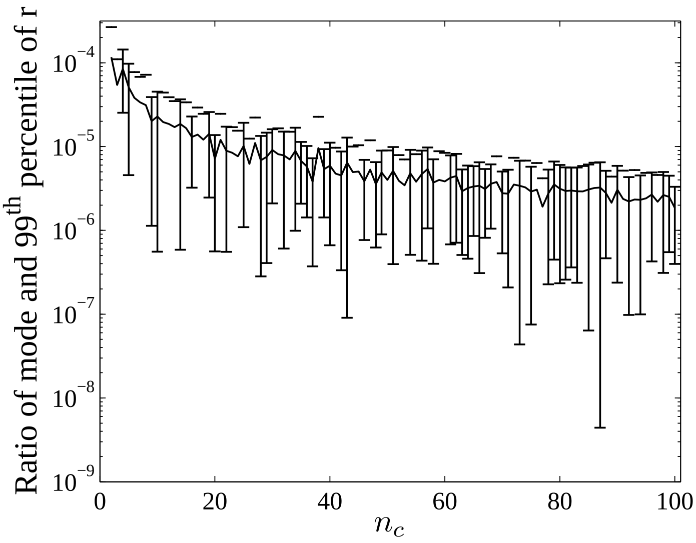


Figure 7.51: Ratio of the mode and 99th percentile of r, suggesting a non-uniform decrease in r values as n_c increases.

in Figure C.8. It is clear that the 99th percentile is decreasing at a faster rate. This result suggests that the decrease in magnitude of r is non-uniform across the r values. Specifically, as n_c increases, there is an increase in the number of (relatively) small length scales.

Overall, the analysis of the length scale distributions suggests that the structure of the problem varies considerably as n_c increases from low values. However, for larger values of n_c, the structure is quite similar between problems, resulting in similar length scale distributions. In general, the magnitude of r is decreasing as n_c increases, however the decrease is not uniform across r values.

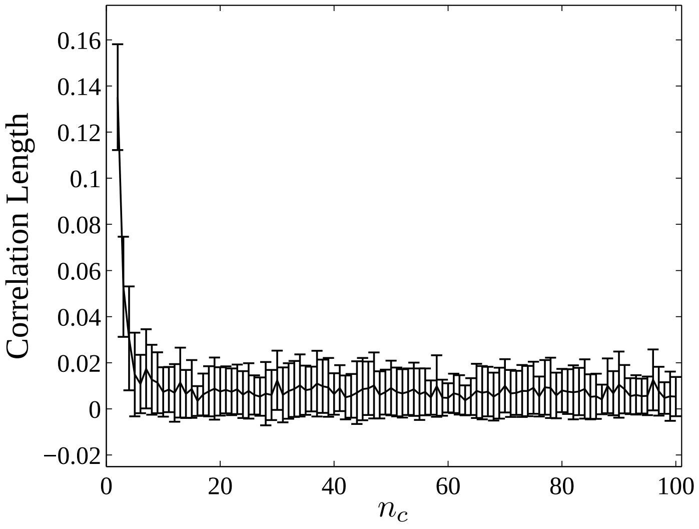


Figure 7.52: Correlation Length of the CiaS problems for $n_c = 2, \ldots, 100$.

7.3.2 Results Comparing Length Scale to Existing Features

Similar to the BBOB experiments in Section 7.2.2, this experiment investigates how well correlation length, dispersion, FDC, information content, partial information content and information stability characterise the circle packing problems. The length scale entropy, $h(r)$, is also included and compared to the existing features. Figures 7.52 to 7.59 displays the mean and one standard deviation (as error bars) for each feature across the CiaS problems.

The correlation lengths shown in Figure 7.52 are generally very low and do not discriminate between the circle packing problems. At $n_c = 2$, the correlation length is approximately 0.1352, and as n_c increases, the correlation length decreases until a value of approximately 0.0074 is reached (at $n_c = 10$). The values remain rather constant, with only slight fluctuations, as n_c increases. Furthermore, The standard deviation between samples, shown as error bars, is quite low. Correlation lengths near 0 indicate extremely rugged landscapes. Hence, the results in Figure 7.52 suggest that the CiaS problems are highly rugged, and that ruggedness increases as n_c increases until a threshold of ruggedness is reached. While correlation length is able to detect a high degree of ruggedness, it is unable to distinguish between problems for $n_c > 10$.

The bound-normalised Dispersion values for the CiaS problems are shown in Figure 7.53, and range between 0.5044 and 0.5627, with small variability between samples (indicated by the small error bars). Encouragingly, the use of a Lévy walk and bound-normalisation has prevented the values from converging to 0. The dispersion metric is defined on the interval $[0, 1]$, with near-0 values indicating a close proximity between fit solutions, and near-1 values

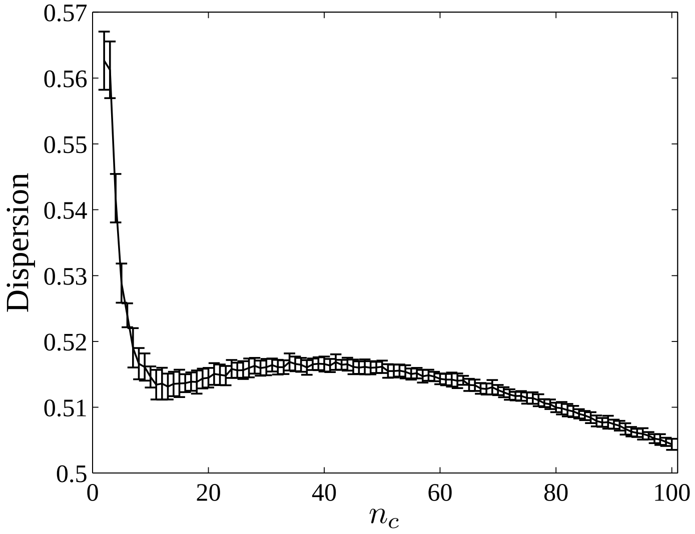


Figure 7.53: Dispersion of the CiaS problems for $n_c = 2, \ldots, 100$.

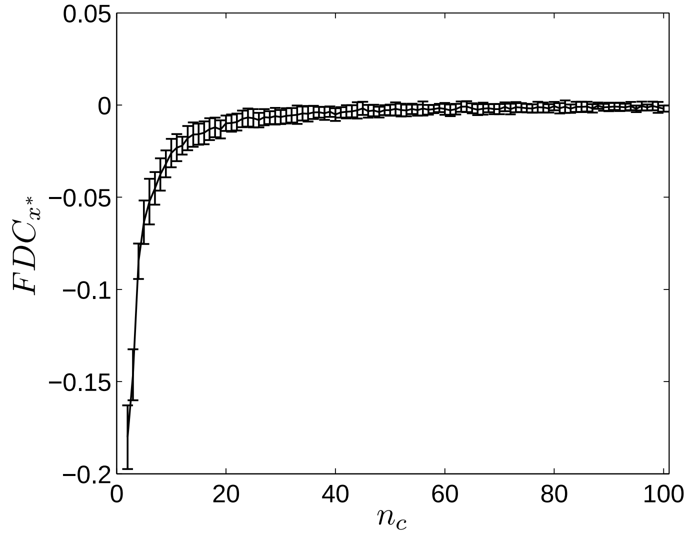


Figure 7.54: $FDC_{\mathbf{x}^*}$ of the CiaS problems for $n_c = 2, \ldots, 100$.

indicating a wide spread of fit solutions throughout $\mathcal{S}$. Hence the values in Figure 7.53 indicate that fit solutions are moderately dispersed throughout the solution for all values of n_c, with fit solutions being *slightly* more dispersed for lower values of n_c, and *slightly* more clustered for large values of n_c. In terms of dispersion's ability to differentiate and classify the CiaS problems, Figure 7.53 shows that it is very limited; the values (particularly for $n_c > 10$) are very similar and non-unique across the problems. Interestingly, correlation length is also unable to differentiate problems for $n_c > 10$, which suggests that the landscape structures dispersion and correlation length inherently rely upon are inadequately characterising the CiaS problems.

In general, both estimators of FDC (shown in Figure 7.54 and 7.55 respectively) have

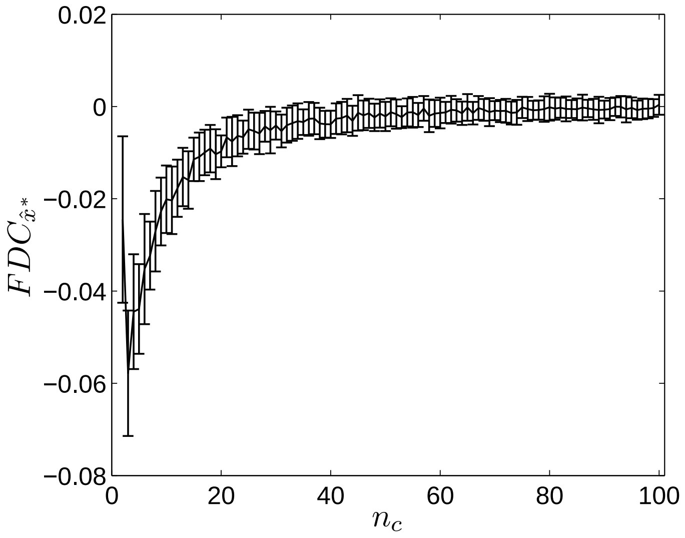


Figure 7.55: $FDC_{\hat{\mathbf{x}}^*}$ of the CiaS problems for $n_c = 2, \ldots, 100$.

small standard deviation (errorbars) over samples, which decreases as the number of circles increases. FDC values are typically small and negative for small n_c, and as n_c increases, values increases towards 0. However, an exception of this trend occurs at the transition from $n_c = 2$ to $n_c = 3$, where the $FDC_{\hat{\mathbf{x}}^*}$ transitions from -0.0245 to -0.0578. Values then steadily increase towards 0 as n_c increases.

Figure 7.54 and 7.55 generally indicate that for low numbers of circles (i.e. $n_c < 20$), the objective function values of solutions is slightly negatively correlated with their distance to the global optimum, however, as the number of circles increases, the objective function values of the solutions has essentially no correlation with their distance to the global optimum. A negative value of FDC in the context of a minimization problem indicates that in general, the f-values of the sampled solutions gets better as the distance from their closest global optimum increases. Such a circumstance can be caused by many factors (and their interactions), including the presence of many local optima and multiple global optima, which CiaS problems are known to have (as discussed in Section 7.3 above). The FDC values alone give no further insight into such factors, nor do they adequately differentiate between problems of varying n_c (particularly for $n_c > 40$).

The information content and partial information content features are highly correlated for the CiaS problems (the sample correlation coefficient is 0.9988). Consequently, having calculated information content, no additional information is obtained from the partial information content (and vice versa). The values of information content and partial information content are roughly constant over all of the CiaS problems, with small fluctuations as indicated by the scale on the axes in Figures 7.56 and 7.57. Comparisons with the information

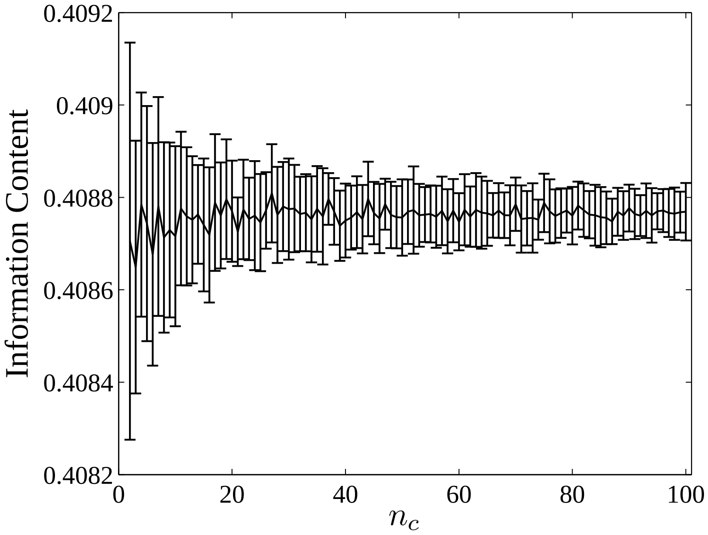


Figure 7.56: Information content of the CiaS problems for $n_c = 2, \ldots, 100$.

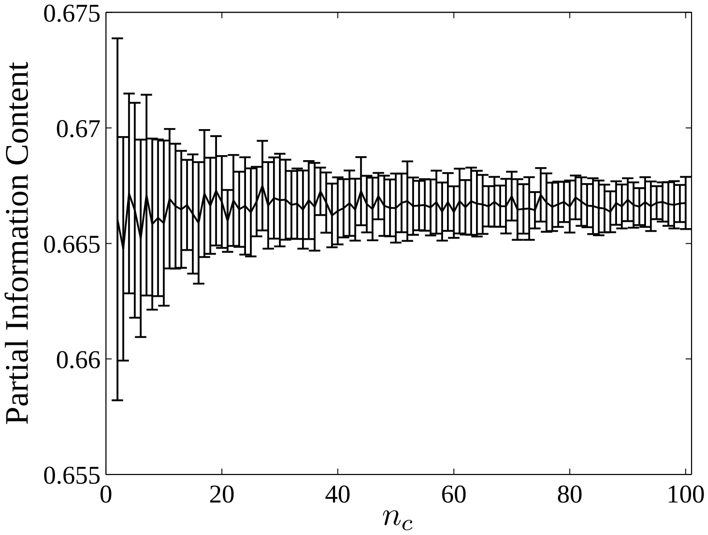


Figure 7.57: Partial information content of the CiaS problems for $n_c = 2, \ldots, 100$.

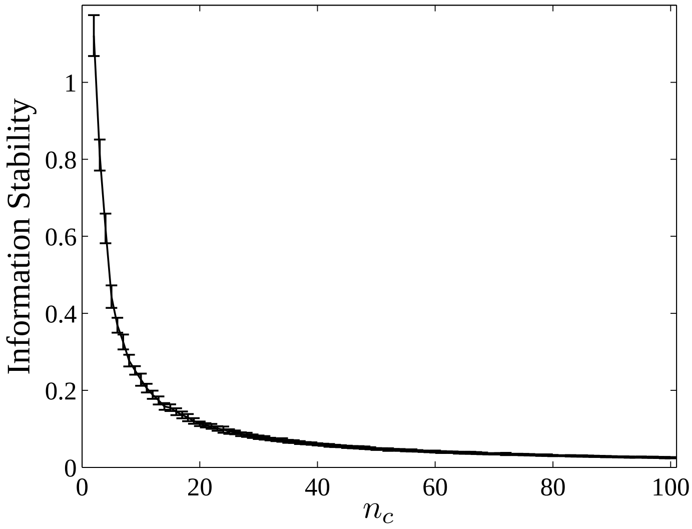


Figure 7.58: Information stability of the CiaS problems for $n_c = 2, \ldots, 100$.

content and partial information content of highly rugged landscapes in [190] suggest that the values (and fluctuations) obtained are reasonable. Similar to other features, the standard deviation decreases as n_c increases. The information content and partial information content features indicate that the problems do not significantly change in ruggedness. Most importantly, the features are clearly unable to differentiate and characterise CiaS problems.

In contrast to information content and partial information content, the information stability feature, shown in Figure 7.58, exhibits a strong, smooth trend as n_c increases and has very small standard deviation between samples. In particular, as n_c increases, the information stability is a monotonically decreasing function, approaching 0. This is determined by the nature of objective function values in CiaS problems and the evaluation of solutions. As discussed in the context of the length scale distribution analysis, the objective function value assigned to a solution is the minimum distance between any two circle centres (multiplied by -1, as this analysis assumes minimisation). As n_c increases, the radius of the circles decreases, and so for a random solution, the minimum distance between any two circle centres is expected to also decrease. Consequently, the magnitude of the minimum objective function value of a random solution is also expected to decrease as n_c increases. Information stability is simply the largest change in objective function value between two steps in the walk. Thus, while information stability is a robust and unique characteristic the problem for changing n_c, the decrease in information stability that is observed in Figure 7.58 is likely to be due to the decrease in magnitude of the objective function, rather than the landscape structure. Furthermore, analysis of individual information stability values does not give much insight into landscape structure. For example, at $n_c = 2$, the average information

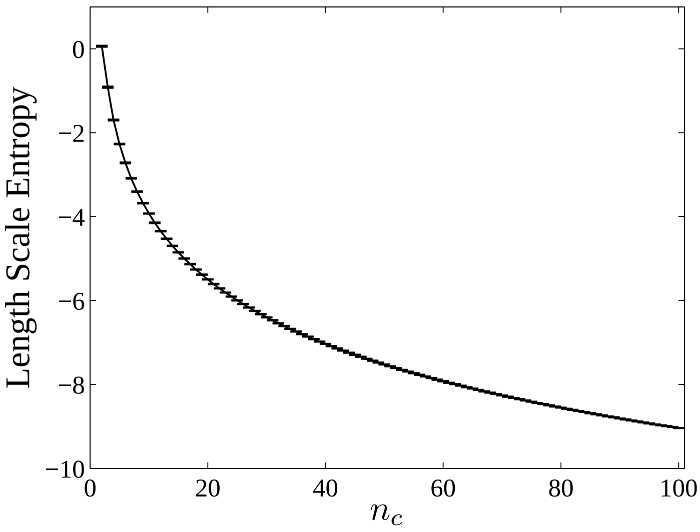


Figure 7.59: Length scale entropy of the CiaS problems for $n_c = 2, \ldots, 100$.

stability over the 30 samples is approximately 1.1213. This merely indicates that the largest change in objective function values (between a step in the walk) is 1.1213; no information regarding other changes in objective function values, the distribution of objective function values or the interaction of solutions and objective function values is captured.

The entropy of the length scale distribution, $h(r)$, is shown in Figure 7.59. It clearly characterises the problem. Specifically, as n_c increases, $h(r)$ decreases in a smooth, highly predictable manner. The $h(r)$ values in Figure 7.59 have very low standard deviation across the repeated samples, which suggests that for these problems, it is a highly robust landscape feature.

Since $h(r)$ is decreasing as n_c increases, the information required to describe r is decreasing, meaning that there is an increasing frequency of similar length scales. This in turn indicates that as n_c increases, the diversity of the changes in objective function between two solutions is decreasing. Figure 7.49 nicely illustrates this; $p(r)$ for $n_c = 2$ has a much heavier tail than $p(r)$ for $n_c = 5$, indicating $n_c = 5$ has less diversity of length scales, and hence, less diversity in objective function changes.

A second experiment is included in Appendix C, which is based on the same experimental setup, except a uniform random walk is used instead of the Lévy random walk. The results using the uniform random walk are quite similar to the results above, with the small exception of the FDC coefficients, which appear to be much more volatile when the uniform random walk is employed.

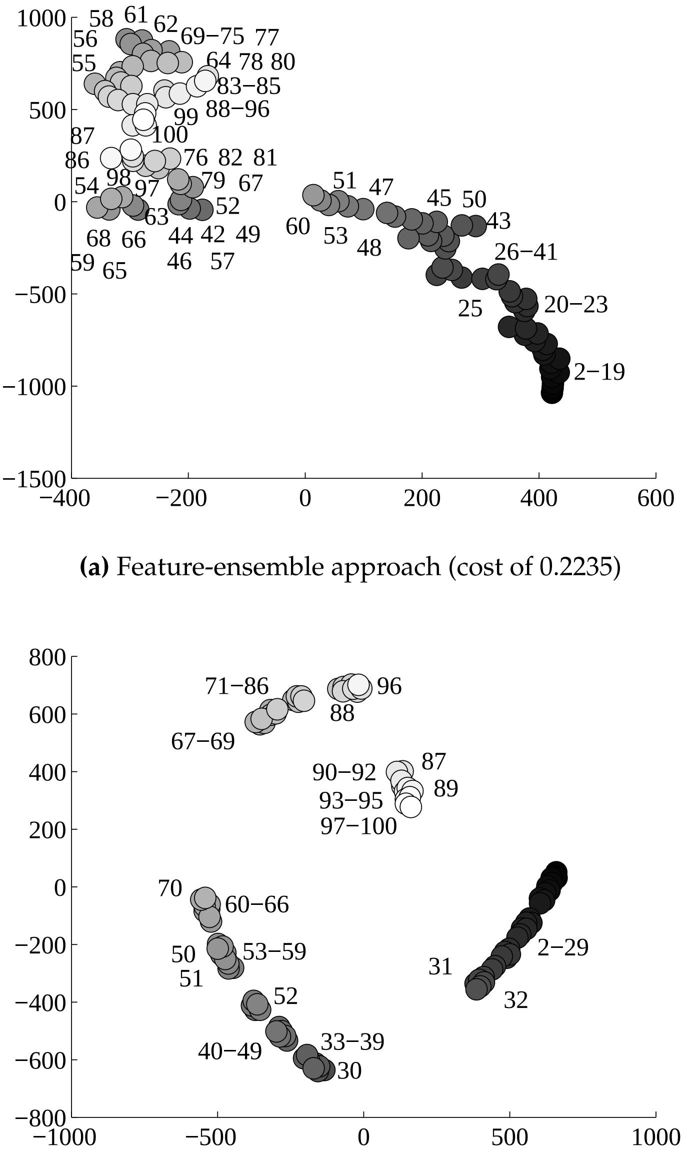
7.3.3 Results Comparing Length Scale with an Ensemble of Features

Similar to the analysis conducted on the BBOB problems in Section 7.2.3, here the seven existing features (correlation length, dispersion, $FDC_{\mathbf{x}^*}$, $FDC_{\hat{\mathbf{x}}^*}$, information content, partial information content and information stability) estimated on the CiaS are combined into a feature ensemble and visualised in a two dimensional "problem space". The features are normalised by their appropriate bounds (information stability is normalised by its range). The 7-D feature space is reduced using t-SNE (with the Euclidean distances between features vectors as input).

The J-divergences estimated between the length scale distributions of the CiaS problems are also used to infer the problem space. Specifically, the average J-divergences between the CiaS problems across the 30 different samples were used to calculate a dissimilarity matrix, and t-SNE was applied. Based on the recommendations in [186] and exploratory experimentation, the perplexity was set to 5 for all t-SNE reductions in these experiments. Due to the stochastic nature of t-SNE, 1000 different trials were conducted, with a maximum of 1000 iterations for each trial. Figure 7.60 shows the best t-SNE visualisation (in terms of the final cost) for each approach. The costs of t-SNE solutions were quite consistent across the 1000 trials; the feature-ensemble approach ranged between 0.2235 and 3.1630 with a median cost of 0.2606, while the length scale approach ranged between 0.1186 and 4.5436 with a median cost of 0.1820.

The visualisation of the feature-ensemble t-SNE solution (Figure 7.60a) shows that the CiaS problems form two major clusters, roughly based on n_c. Specifically, the problems are split roughly around $n_c = 50$ (with a few exceptions). The visualisation shows that as n_c increases from 2, the locations of the problems progressively follow a linear pattern, indicating that the similarity between problems of small n_c and large n_c are dissimilar. As the number of circles passes 40, the pattern deteriorates and a second cluster is formed. The second cluster consists mainly of problems where $n_c \geq 40$. The second cluster does not follow a linear pattern, rather, problems are arranged roughly in the shape of a circle, indicating that the problems are similar.

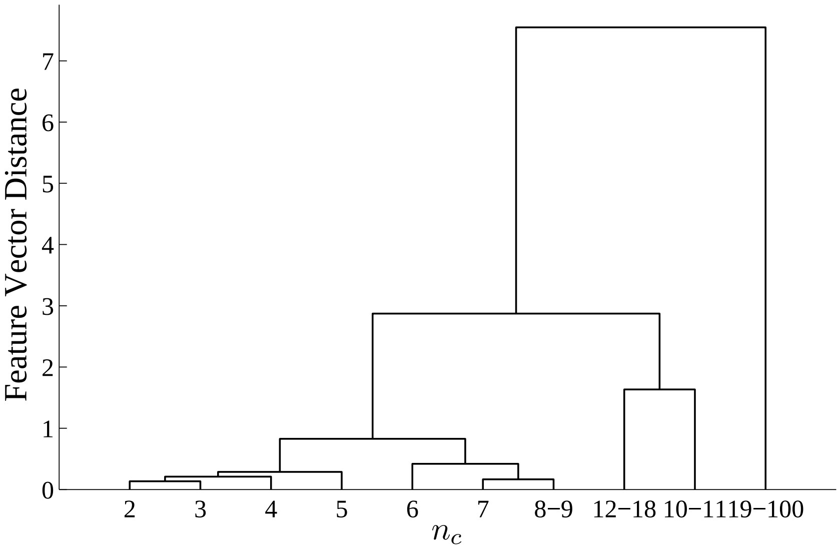
The t-SNE visualisation of the J-divergences between problems (Figure 7.60b) is quite different to the feature-ensemble approach. In particular, the J-divergences form 4 clear clusters, approximately defined by $2 \leq n_c \leq 32$, $33 \leq n_c \leq 66$, $67 \leq n_c \leq 86$ and $87 \leq n_c \leq 100$. There are a few exceptions; $n_c = 30$, 70 and 96 are in different clusters compared to instances with one less/more circle to pack (e.g. $n_c = 30$ is positioned in a different



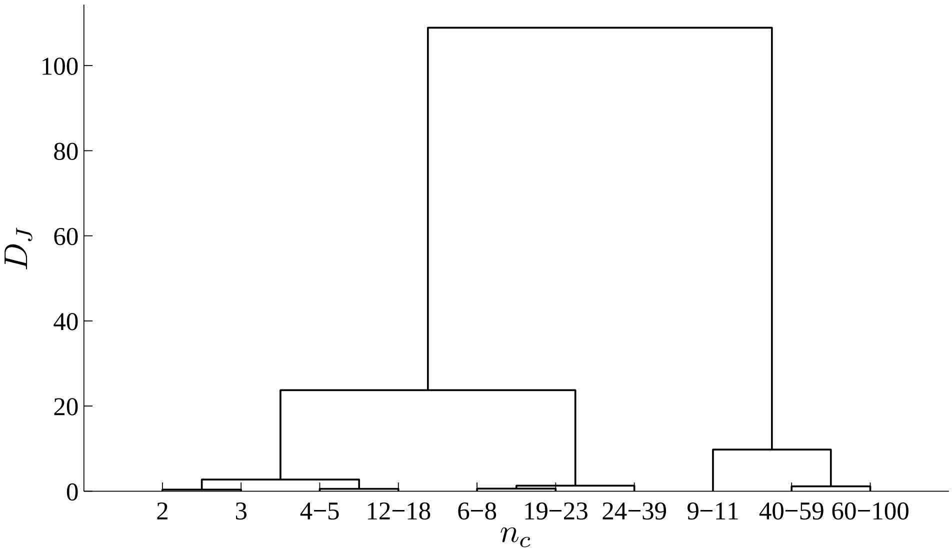
(a) Feature-ensemble approach (cost of 0.2235)

(b) Length scale approach (cost of 0.1186)

Figure 7.60: Feature spaces of CiaS problems ($n_c = 2, \ldots, 100$) reduced via t-SNE.



(a) Feature-ensemble approach



(b) Length scale approach

Figure 7.61: Dendrograms of the CiaS problems ($n_c = 2, \ldots, 100$) problems.

cluster to $n_c = 29$ and $n_c = 31$). However, because $n_c = 30$, 70 and 96 remain close to instances of *similar* packing sizes (e.g. $n_c = 30$ is positioned in cluster $33 \leq n_c \leq 66$), the separation is likely due to t-SNE. Indeed, the dendrogram in Figure 7.61b shows instances $n_c = 30$, 70 and 96 are similar to instances differing by one circle. The problems in clusters $2 \leq n_c \leq 32$ and $33 \leq n_c \leq 66$ are positioned in a linear pattern (as n_c increases), while the problems in clusters $67 \leq n_c \leq 86$ and $87 \leq n_c \leq 100$ are more circular. As discussed above, linear patterns indicate a dissimilarity between problems, while circular patterns indicates similarity. Hence, the t-SNE visualisations of both the feature-vector distances and J-divergences shows that as n_c increases, problems of similar n_c values become increasingly more similar.

The dendrograms for the feature-ensemble and length scale approaches are shown in Figure 7.61. To aid in visualisation and interpretation, the 99 CiaS problems are categorised into 10 groups based on their similarities (as measured by D_J and the Euclidean distance between feature vectors). Both dendrograms show that there is a strong similarity between CiaS problems with large numbers of circles; problems where $n_C > 19$ and $n_C \geq 60$ form the largest clusters in Figures 7.61a and 7.61b respectively. The dendrograms also show that the largest dissimilarity between problems is between low (e.g. 2) and high (e.g. 100) values of n_C. The feature-ensemble approach (Figure 7.61a) assigns clusters to the individual problems $n_c = 2$ to $n_c = 7$, whereas the J-divergence approach (Figure 7.61b) only assigns $n_C = 2$ and $n_C = 3$ their own clusters. Hence while both dendrograms show similar trends, the feature-ensemble dendrogram shows a greater dissimilarity between problems with low numbers of circles.

To summarise, both the feature-ensemble and length scale approaches to analysing the similarity between the CiaS problems indicates that problems with a small number of circles to pack are more dissimilar than problems with a large number of circles to pack. In particular, it appears that as n_c increases, the similarity between problems also increases.

7.4 Analysis of TSPLib

Thus far, the length scale analysis has been used to characterise continuous optimization problems. However, length scale can also be readily applied to combinatorial problems, and so the experiments and results in this section and Section 7.5 and 7.6 focus on analysing well-known combinatorial problem sets. In particular, this section examines TSPLib, which is a widely-used Travelling Salesman Problem benchmarking set.

The TSPLib collection is an interesting problem library to analyse as there are a wide variety of instances, with some of the instances sharing similar sources, distance metrics, and/or number of cities (n). The aim of this experiment is to apply the length scale analysis to a subset of instances from TSPLib to evaluate whether the length scales capture relationships between the instances. Some of the instances within TSPLib come from a similar source, and so it is expected that there may be structural similarity between such problems.

The TSPLib instances in Table A.3 range between 17 and 100 cities. A 100-city symmetric TSP has $\frac{99!}{2}$ ($\approx 4.67 \times 10^{155}$) candidate solutions, while a 100-city asymmetric TSP has 99! ($\approx 9.33 \times 10^{155}$) candidate solutions, and so clearly, complete enumeration of the solutions in *all* of the TSPLib instances is infeasible. Therefore in this experiment, a sample of r values is obtained using Algorithm 6.1 with $2.5 \times 10^5 n$ solutions generated by uniform random sampling of feasible tours. A tour can be represented by a sequence of integers from 1 to n; integers uniquely identify each city, and the order of the sequence corresponds to the order in which the cities are visited. Uniform random (feasible) tours are generated by uniform randomly permuting the order of integers in the sequence $(1, \dots, n)$. The objective function value of a solution is the total distance of the tour solution. Using the TSP and ATSP solution distances defined in Equations A.2.7 and A.2.8 (see Appendix A.2.1) respectively, the length scale between two TSP solutions can be calculated via:

$$r(\mathbf{x}^a, \mathbf{x}^b) = \frac{\left|\sum_{i \neq j} D_{i,j} x^a_{i,j} - D_{i,j} x^b_{i,j}\right|}{\mathrm{dist}_{\mathrm{TSP}}(\mathbf{x}^a, \mathbf{x}^b)} \tag{7.4.1}$$

and the length scale for two candidate ATSP solutions is:

$$r(\mathbf{x}^a, \mathbf{x}^b) = \frac{\left|\sum_{i \neq j} D_{i,j} x^a_{i,j} - D_{i,j} x^b_{i,j}\right|}{\mathrm{dist}_{\mathrm{ATSP}}(\mathbf{x}^a, \mathbf{x}^b)} \tag{7.4.2}$$

7.4.1 Length Scale Analysis Results

The resulting length scale distributions in Figures 7.62 to 7.65 show that instances from a common source have strong similarities in their length scale distributions. While instances from a common source clearly have similar length scale distributions, the difference in axes between the sources illustrates that there is a clear difference *between* sources. For example, Figures 7.62 and 7.64 show that the $p(r)$s of instances from a particular source can be very different to the $p(r)$s of instances from *other* sources. The length scale distributions are shown together in Figures 7.66 and 7.67. While some instances have very similar length scale distributions (e.g. the "kro"-type instances), the length scale distributions can also vary by orders of magnitude (e.g. "br17"). It is clear that problems from similar sources have more similar length scale distributions than problems from different sources. Overall, the length scale distributions are a feature of these problem landscapes that can distinguish sources.

Instance p43 has a particularly interesting length scales, as shown by the length scale multiset and length scale distribution in Figure 7.68. Almost all length scales within this problem are close to one of four values; 1, 5000, 10000 and 25000. As a result, $p(r)$ has four clear modes. While the majority of length scale distributions examined in this thesis are

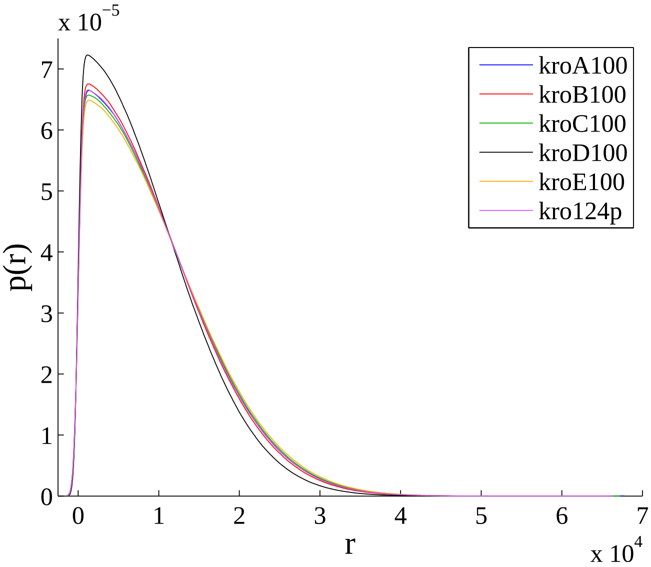


Figure 7.62: Length scale distributions of "kro"-type instances from TSPLib.

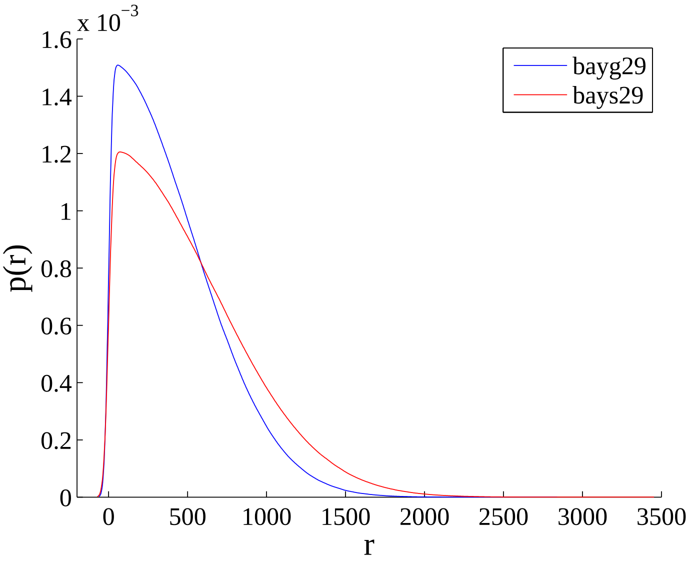


Figure 7.63: Length scale distributions of "bay"-type instances from TSPLib.

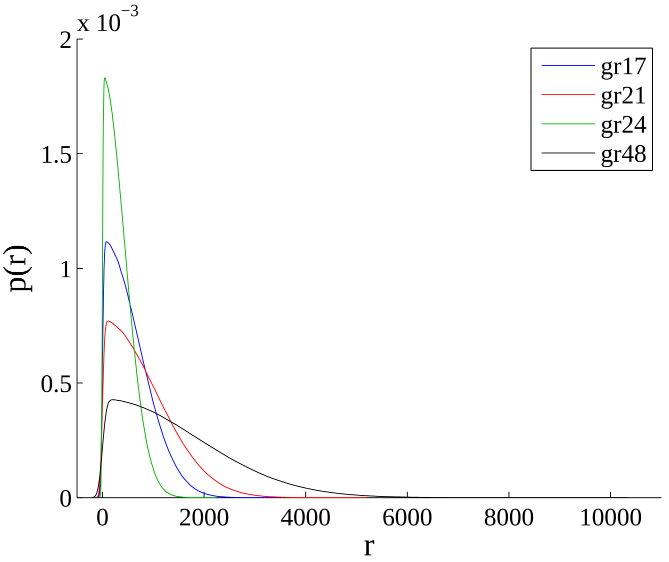


Figure 7.64: Length scale distributions of "gr"-type instances from TSPLib.

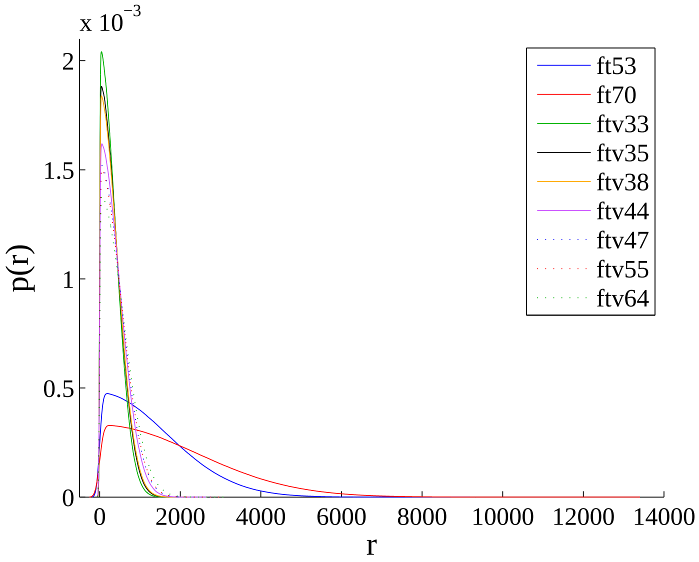


Figure 7.65: Length scale distributions of “ft”-type instances from TSPLib.

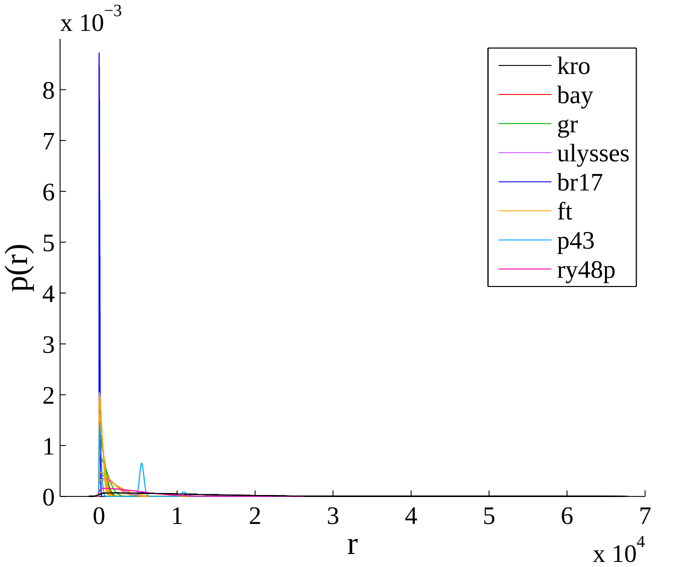


Figure 7.66: Length scale distributions of the TSPLib instances.

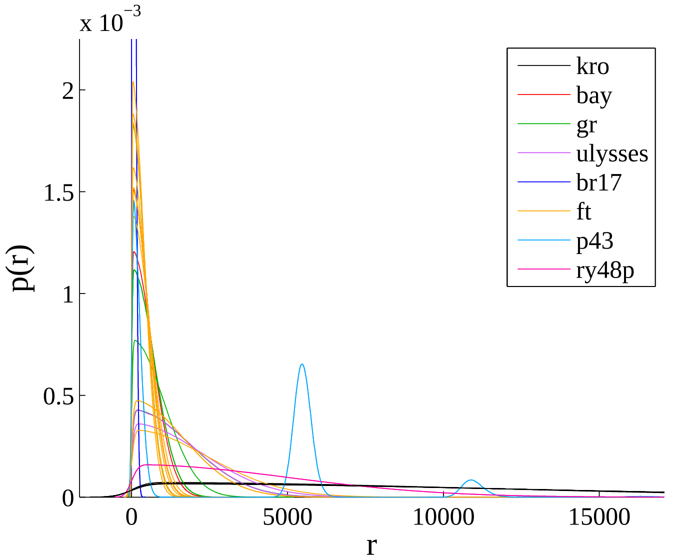
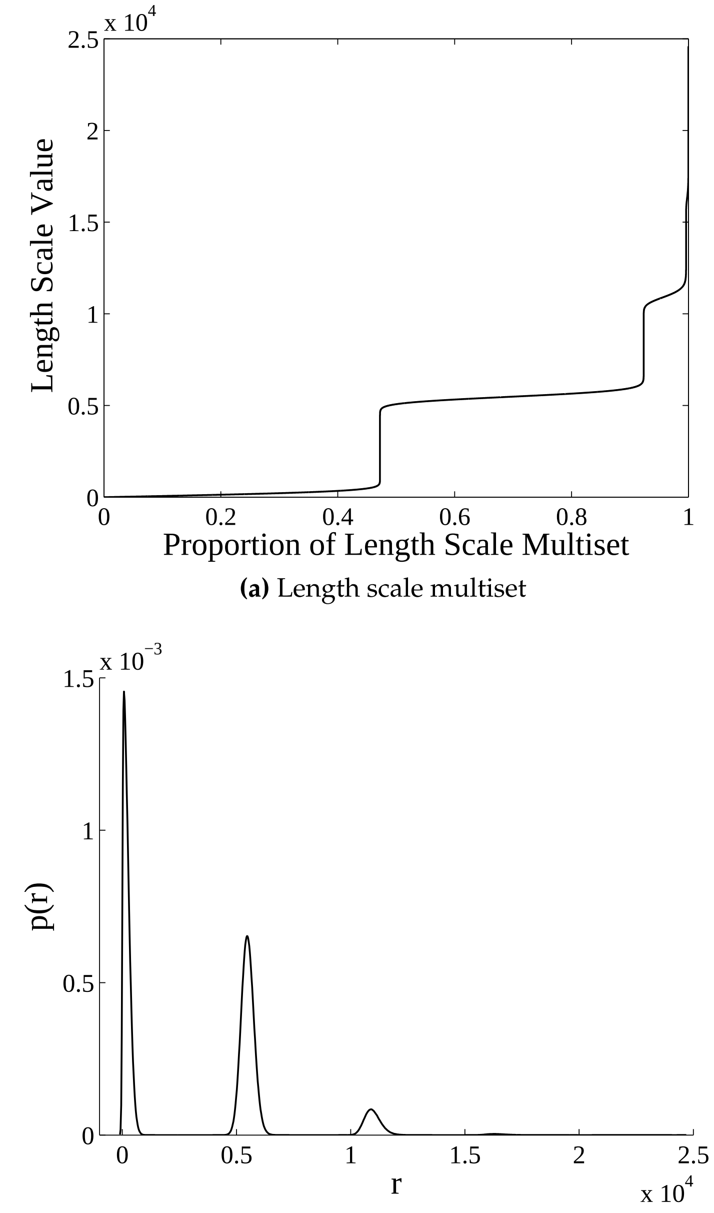


Figure 7.67: Zoomed-in length scale distributions taken from Figure 7.66.



(a) Length scale multiset

(b) Length scale distribution

Figure 7.68: Length scales for instance p43.

unimodal, Figure 7.68b illustrates that multimodal distributions are possible in practice.

Heatmaps summarising the D_J between problems as well as t-SNE (perplexity of 5, best result from 1000 trials with a maximum of 1000 iterations) visualisations of the problem space are shown in Figures 7.69 and 7.70. The final cost of the optimization for t-SNE is 0.2041, which indicates that the visualisation is an adequate depiction of the relationships between instances.

The heatmap in Figure 7.69 clearly shows that instances from similar sources (e.g. "kro") have small J-divergences and are therefore similar to each other. While instances from different groups are generally dissimilar, the largest difference between instances is between p43 and ft53. For many of the instances, p43 is the most different instance in comparison to

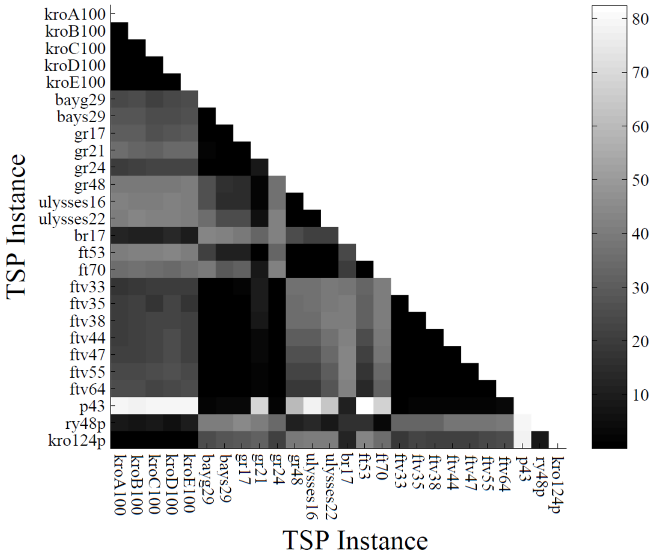


Figure 7.69: Heatmap of D_J values calculated between pairs of TSPLib instances.

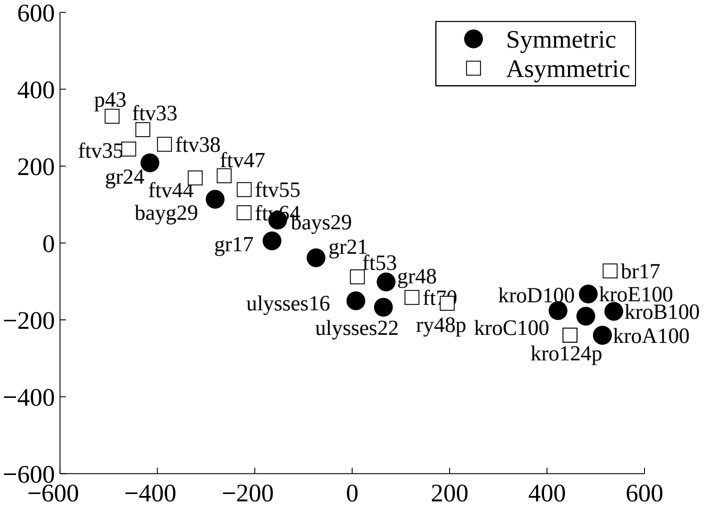


Figure 7.70: t-SNE of D_Js (cost of 0.2041) calculated between pairs of TSPLib instances.

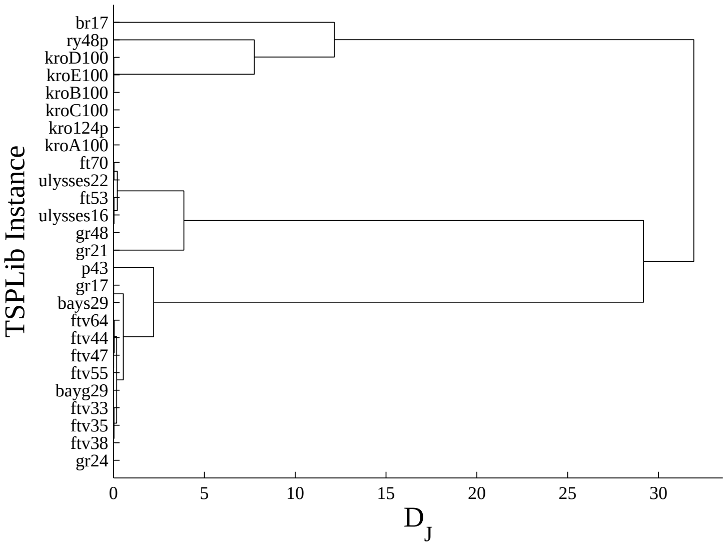


Figure 7.71: Dendrogram of D_Js calculated between pairs of TSPLib instances.

the other instances. Interestingly, there is no discernible difference between symmetric and asymmetric "kro"-type instances.

Figure 7.70 shows the t-SNE visualisation of the problem similarities (and hence, "problem space"). Here, the instances are labelled with their name marked according to their TSP type class (symmetric problems are black circles, asymmetric are represented by white squares). Figure 7.70 captures the general trend that instances from similar sources are similar to each other. Here, the "kro"-type instances form their own cluster (without regard to (a)symmetry). The remaining instances form a second cluster, and within this cluster are loosely organised according to their problem source and type. For example, all "ftv"-type problems are located together, as are the "ulysses"-type problems. There is one exception; gr24 is not located with the remaining "gr"-type problems. Overall, Figure 7.70 illustrates that D_J nicely discriminates between the TSPLib instance sources.

The dendrogram in Figure 7.71 also demonstrates the similarity of problems from common sources. Instances are generally clustered according to their source. For example, all "kro"-type problems form a cluster, as do the "ftv"-type problems. Interesting, the "gr"-type problems are split between different clusters. Using the dendrogram to cluster the instances into two clusters yields {br17, ry48p, kroA100, kroB100, kroC100, kroD100, kroE100, kro124p} and the remaining problems, which closely matches the problem space visualisation in Figure 7.70.

7.4.2 Existing Results

TSPLib instances have been widely used to benchmark and compare the performance of algorithms. Consequently, a large amount of the TSPLib literature has a strong emphasis on analysing algorithm behaviour and performance. Developing an understanding of algorithm behaviour is often closely associated with landscape analysis, and as a result, a relatively large amount of analysis has been conducted on TSPLib instances. However, due to the strong focus on algorithm behaviour, the analysis is often aimed at understanding algorithm behaviour, rather than the problem itself.

Draskoczy [46] calculated FDC coefficients (using the global optimum as reference) for 5 TSPLib instances (not studied in the experiments above) with 5 different distances, based on commonly used move operators. Merz and Freisleben [115] also analysed the FDC calculated between random solutions and the global optimum (as the reference point) for 9 TSPLib problems (their experiments used two distances induced by the move operators 3-opt and Lin-Kernighan). In addition, distinct local optima were determined using random restarts of a local search and the following features were derived and analysed: 1) the minimum distance between the global optimum and a local optimum, 2) the average distance between all local optima and the global optimum, 3) the average distance between all local optima and 4) the total number of distinct local optima. The aforementioned analyses are obviously very computationally expensive and rely on knowledge of the global optimum, which is not readily available in many situations (e.g. black-box and real world problems).

The resulting FDC coefficients of the TSPLib instances analysed in [46, 115] are mainly positive. The analysis of the distances between local and global optimum indicates that many of the instances defined using Euclidean-distance have a central massif/big-valley structure, where the global optimum is centrally located between local optima. As a result, many of these instances are reportedly “easy to solve”, and contain relatively smooth transitions in objective function values [115, 137].

Elementary landscape analysis on symmetric TSPLib instances (with a focus on the 2-opt move operator) also indicates a degree of smoothness between random neighbouring solutions [203]. The elementary landscape analysis also showed that TSPLib instances are *partially decomposable*, meaning that for certain move operators, the calculation of f for a new solution generated by adding/removing tour edges does not involve an entire calculation of the tour. Instead, the calculation of the previous tour can be modified to include/exclude the added/removed edges.

In contrast to the highly algorithm-focused analysis discussed above, Smith-Miles and Tan [164] analysed the *kroA1OO*, *kroB1OO*, *kroC1OO*, *kroD1OO*, *kroE1OO*, and *rd1OO* TSPLib instances using 40 different TSP features (a thorough review of the TSP features can be found in Smith-Miles and Lopes [163]). Their results showed that the landscape features were not able to adequately discriminate between the TSPLib instances. The results in Section 7.4.1 are in agreement that the "kro"-type instances are highly similar, and so perhaps the lack of discrimination in [164] is simply because the problems are fundamentally similar in structure.

To summarise, much of the landscape analysis of TSPLib instances has been performed with a strong focus on understanding algorithm behaviour. In addition, many of the features used to analyse the instances are highly TSP-specific, and rely on information, such as the global optimum, that is not readily available in real-world or black-box problems. In contrast, the length scale analysis conducted in this section is able to accurately distinguish and discriminate between instances using *only* a finite sample of solutions and their objective function values.

7.5 Analysis of Asymmetric Travelling Salesman Problems

Asymmetric Travelling Salesman Problems (ATSPs) are known to undergo phase transitions in various structural features of the problem as well as the difficulty of some exact solvers [206]. For further details on how to generate ATSPs throughout various stages of the phase transition, please refer to Appendix A.2.1. The aim of this experiment is to investigate how the length scales of ATSP instances change throughout various stages of the phase transition, and whether or not length scale analysis is sensitive to the phase transition. Zhang [206] previously analysed the landscapes of random ATSP instances with $n = 100, 200, \dots, 1000, 1500$ cities, where distances are generated uniform randomly from integers in the range $[0, \dots, \lfloor 10^b \rfloor]$, where $b > 0$. The results show that the phase transition is invariant to n and typically occurs within $\beta = [0.5, 3.5]$, where $\beta = \frac{b}{\log_{10} n}$. In addition, Zhang [206] shows that the critical point of the phase transition is 2 as the expected number of distinct distances in D approaches ∞. Because the number of cities does not affect the phase transition behaviour, this experiment focuses on 10-city problems (i.e. $n = 10$) and uses $b = \beta = 0.5, 0.6, \dots 6.5$ to generate the instances at various stages of the phase transition. At each value of b, 10 random ATSP instances are generated using 10 distinct distance matrices, where entries in each matrix (excluding the leading diagonal) are sampled uniform

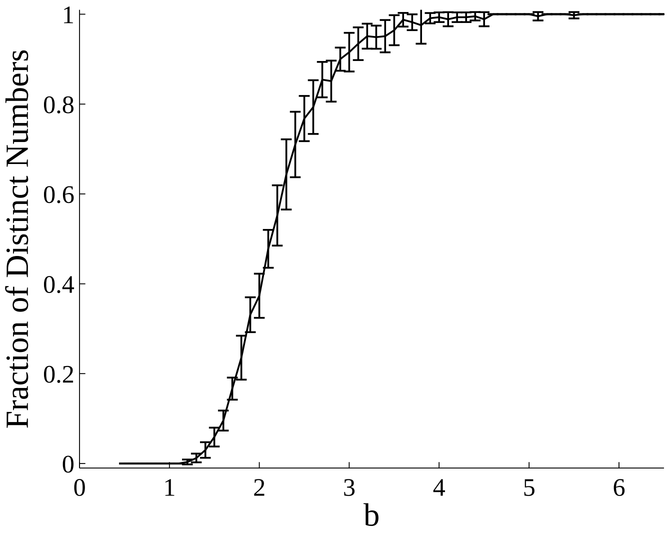


Figure 7.72: Proportion of unique distances averaged across 10 random instances of each b.

randomly from integers in the set $[0, \ldots, \lfloor 10^b \rfloor]$.

One particular problem-feature of TSPs is the fraction of unique distances in the distance matrix. Zhang [206] showed that this feature (among others) exhibits a phase transition as b increases. Figure 7.72 shows the fraction of unique distances, calculated for each ATSP instance and averaged across the 10 instances at each value of b. The results show that the problems generated exhibit the phase transition behaviour described in [206].

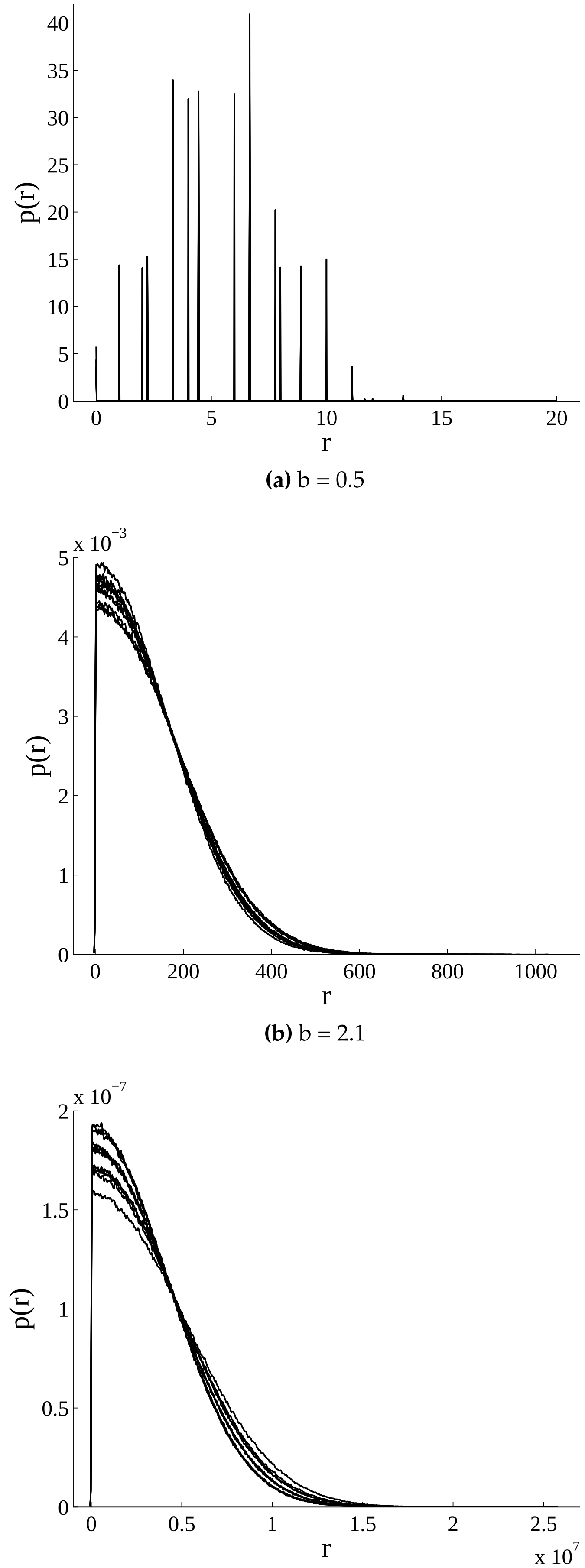
A 10-city ATSP instance has a total of 9! (362880) candidate solutions, which is feasible to completely enumerate. A sample of r values is obtained using Algorithm 6.1 with all 9! solutions as the sample, $\mathcal{S}'$. The length scale for two candidate ATSP solutions is defined in Equation 7.4.2.

7.5.1 Length Scale Analysis Results

Figure 7.73 shows the length scale distributions for $b = 0.5, 2.1$ and 6.5, which correspond to three different parts of the phase transition (specifically, an average fraction of distinct numbers of 0, 0.4788 and 1 respectively). The length scales are very consistent between ATSP instances at a given value of b, and as b increases, the magnitude of the length scales also increases.

Clearly, as b increases, the magnitude of the length scales increases. From the definition of length scale, such an increase can be caused by an increase in the magnitude of the change in objective function and/or a decrease in magnitude of the distance between solutions.

For these experiments, all candidate solutions are utilised and so the search space (i.e.



(a) b = 0.5

(b) b = 2.1

(c) b = 6.5

Figure 7.73: The change in shape of $p(r)$ as b increases.

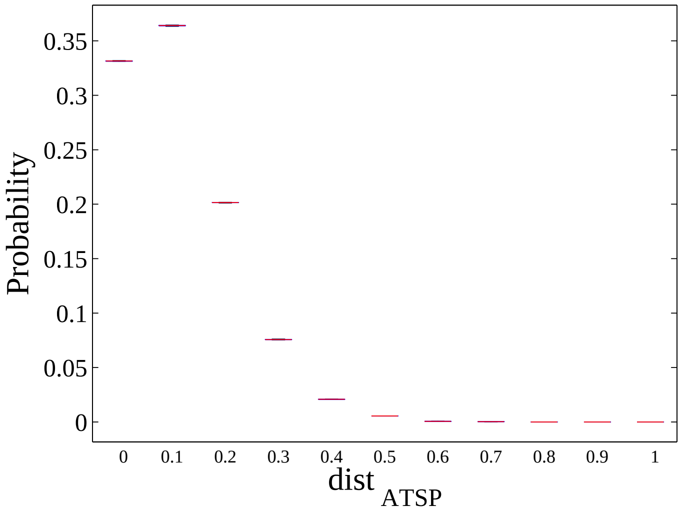


Figure 7.74: ATSP distance distributions for the ATSP instances.

set of candidate solutions) is identical across b. Consequently, the resulting ATSP distances between all pairwise solution combinations are also identical across b. However, not all the conceivable distances between solutions (i.e. all pairwise distances) are used in these experiments, rather, a random subset (due to the use of Algorithm 6.1) is used. Therefore there may be some variation in the ATSP distances, however this is expected to be low. To investigate how much variation there is in the subsets' distances, Figure 7.74 shows a box and whisker plot of the ATSP distance probabilities across all of the random subsets used in the experiments. Because 10 instances are generated at each value of b, there are a total of 10×37 random subsets. Note that because $n = 10$, the number of shared edges is between 0 and 10. Since ATSP distance is calculated between unique candidate solutions, the distance distribution is a probability mass function with probabilities defined at distances of $0, \frac{1}{10}, \ldots, 1$.

Figure 7.74 shows that, as expected, the ATSP distance probabilities have very small variance across the random subsets. Therefore, the increase in magnitude of length scale values exhibited in Figure 7.73 can only be due to an increase in the difference of objective function values. This is indeed the case for the ATSP problems; the objective function is based on the total distance of the tour, and as b increases, the range from which the distances are generated increases. Consequently, the total tour distance (i.e. objective function value) increases, thereby increasing the difference *between* objective function values, and thus, r.

To further examine the behaviour of the ATSP instances, a heatmap summarising the average D_J between problems, a t-SNE (perplexity of 5, best result from 1000 trials with a maximum of 1000 iterations) visualisation of the problem space and dendrogram are shown

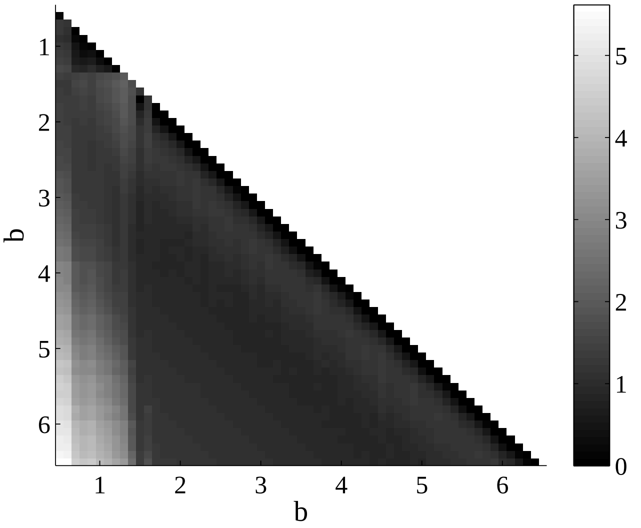


Figure 7.75: Heatmap of D_J values calculated between pairs of ATSP instances with $b = 0.5, 0.6, \ldots, 6.5$.

in Figures 7.75 to 7.77. While most J-divergences are between 0 and 900, some extreme values occur; $D_J(b = 0.5, b = 6.5) = 413592.34$. The outlying J-divergences make the visualisations difficult to interpret, and so $\log_{10} D_J$ is used to generate both the heatmap and dendrogram. The cost value of 0.2473 from t-SNE indicates that the discrepancy of distances between points in the original data and reduced data is moderate, and so the visualisation is not able to fully reflect the relationships between problems.

The heatmap in Figure 7.75 clearly shows that the greatest difference between problems (depicted by white pixels) is between low values of b and high values of b (e.g. $b = 0.5$ and $b = 6.5$). Problems with similar b (i.e. along the leading diagonal) have low D_J values, except for a section where $1.3 \leq b \leq 1.6$ which corresponds to the beginning of the phase transition. This is a highly exciting result; the phase transition behaviour is incredibly well-described by the D_J values. Consequently, the length scale analysis applied to instances of ATSPs has detected the phase transition, but with no prior knowledge of its existence, **and** based purely on black-box samples from the landscape.

Figure 7.76 shows the t-SNE visualisation of the problem similarities (and hence, "problem space"). Here, the problems are labelled with their respective b values, and the markers are coloured and shaped according to their approximate position in the phase transition. Blue dots denote before the transition, red circles denote the transition itself and problems after the transition are marked by black squares. Remarkably, the problem space visualisation is able to separate the problems into their respective phases. The problems before the transition (blue dots) form a tight cluster and are very well-separated from the other

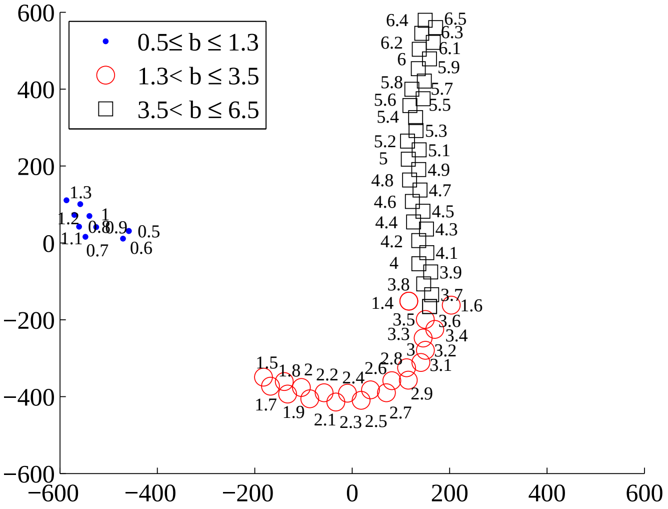


Figure 7.76: t-SNE of D_Js (cost of 0.2473) calculated between pairs of ATSP instances with $b = 0.5, 0.6, \ldots, 6.5$.

problems, indicating that they are structurally similar to each other, but very different to the problems during and after the transition. Problems during the transition (the red circles) are well-organised and curl around the problem space, showing that while there are some similarities in problem structure, the structure is changing as b increases. Interestingly, $b = 1.4$ and $b = 1.6$ (at the start of the transition) are exceptions and are more similar to problems at the end of the transition ($b = 3.5$ and $b = 3.6$). Problems after the transition (black squares) form a linear progression as b increases.

The dendrogram in Figure 7.77 also clearly shows that problems of similar values of b have low D_Js compared to problems with very different values of b. Using the dendrogram to cluster the problems into two major clusters yields problems before the phase transition (i.e. $b \leq 1.3$) and problems after the phase transition begins (i.e. $b \geq 1.3$). While the dendrogram can identify problems before the phase transition, it is unable to distinguish problems *during* the phase transition from problems *after* the phase transition.

To summarise, the length scales are clearly able to capture the information required to describe the phase transition behaviour of the ATSPs. Remarkably, the length scales are calculated *purely* from the information available in the black-box setting, and have no prior notion of the phase transition behaviour. Clearly, length scale is a very powerful summary of landscape information.

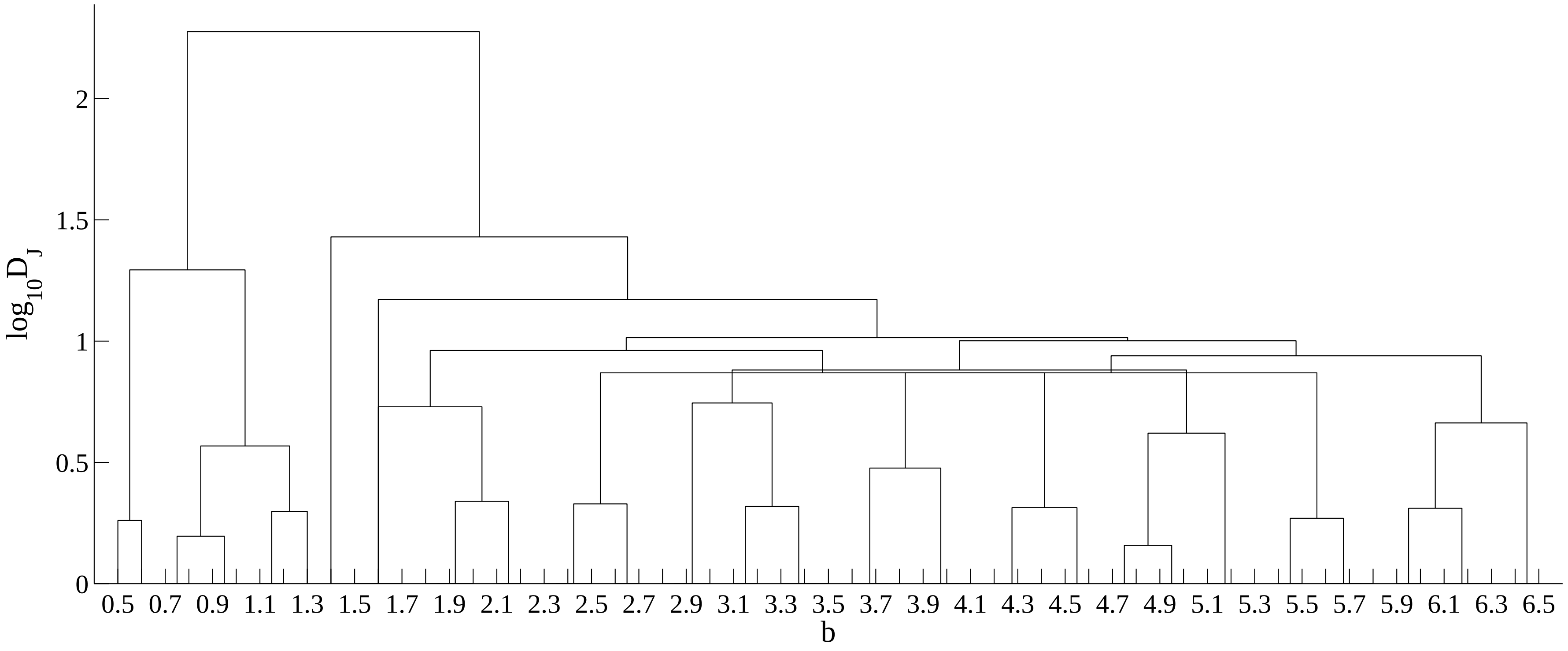


Figure 7.77: Dendrogram of D_Js calculated between pairs of ATSP instances with $b = 0.5, 0.6, \ldots, 6.5$.

7.5.2 Existing Results

TSPs have been well-studied in the problem analysis literature, and as a result, over 40 TSP-specific problem features have been proposed and analysed to date (see [163] for a review). The motivation driving the development of many of the TSP features is to correlate the feature with a notion of problem difficulty. For example, Cheeseman et al. [34] showed that the standard deviation of the distance matrix has a strong relationship with the computational cost of Little's algorithm (an exact solver). TSP features include but are not limited to statistical measures (e.g. variance) of the distance matrix, measures of the number of backbones, characteristics of city clusters and traditional landscape analysis techniques such as autocorrelation and the number of local optima [88, 163, 173]. While there are numerous features proposed, only a handful exhibit the TSP phase transition behaviour, and of these, none are practically feasible to estimate *and* solely reliant on black-box information. For example, Slaney and Walsh [159] define the backbone of an optimization problem as the set of "frozen" decision variables, that is, variables with constant values across all of the optimal solutions. While computation of a TSP backbone technically requires no domain knowledge, it does require finding all optimal solutions, which is NP-hard. To achieve tractability, Kilby et al. [88] define an approximation of the backbone based on various exploitations of the problem, which are only known through a deep level of TSP domain understanding. Hence, *tractable* measures of the TSP backbone rely heavily on intricate TSP domain knowledge. Hernando et al. [73] propose two problem measures for capturing phase behaviour; 1) the proportion of the global basin of attraction with respect to $\mathcal{S}$ and 2) the proportion of unique local optima with respect to $\mathcal{S}$. These measures are also heavily dependent on prior information (e.g. the global optimum) and require enumeration of all local optima, which is a very computationally expensive task for large n. As demonstrated in the above experiment, length scale is very adept at capturing phase behaviour, while it is remarkably easy to apply, practically feasible and operates solely on black-box information.

7.6 Analysis of Number Partitioning Problems

The Number Partitioning Problem (NPP) defined in Appendix A.2.2 is known to undergo phase transitions in terms of the number of global optima, the size of plateaus, and the difficulty of exact solvers [17, 22, 174]. This section investigates how the length scales of instances of NPP change throughout different stages of the phase transition. As outlined in Appendix A.2.2, the phase transition of the NPP is determined by the control parameter, $k = \log_2 \frac{M}{N}$ (the number of bits required to encode the set divided by the number of elements in the set).

Alyahya and Rowe [6] have recently analysed the landscapes of NPP instances consisting of 20 elements (i.e. $n = 20$), where $k = 0.4, 0.5, \ldots, 1.3$. In order to draw comparisons with existing literature, the experiments conducted in this section use a similar experimental setup. Using the derivation for the critical control parameter value (k_c) defined in Equation A.2.10, $k_c = 1 - \frac{\ln\left(\frac{10\pi}{3}\right)}{40\ln(2)} \approx 0.9153$. Therefore, instances at $k = 0.4, 0.425, \ldots, 1.3$ are generated in order to observe the phase transition behaviour. A single instance is generated by populating S with integers uniformly selected (without replacement) from $\{1, \ldots, m\}$, where $m = 2^{nk}$. Due to the random generation of the NPP instances, 10 different instances are generated for each value of k.

Because $n = 20$, there are a total of 2^{19} unique and feasibly enumerable candidate solutions. Candidate solutions are represented by a bit-string of length n, where the element at position i in the bit-string is 0 if assigned to the first partition, and 1 if assigned to the second partition. In these experiments Hamming distance is used as a measure of distance between solutions. Length scales are calculated from the complete set of enumerated solutions using Algorithm 6.1.

7.6.1 Length Scale Analysis Results

The length scales for NPPs are very consistent between instances at a given value of k. As k increases, the magnitude of the length scales also increases. To best illustrate this, the (sorted) length scale sets are shown in Figure 7.78 (note the log-scale on the y-axis). Each line represents the (sorted) length scale set from a NPP instance. Lines are coloured according to k, with blue representing $k = 0.4$ and red representing $k = 1.3$.

Clearly, as k increases, the magnitude of the length scales increases. This behaviour is very similar to the ATSP instances analysed in Section 7.5, where the length scale values increased as b increased. To briefly review, an increase in r can be caused by an increase in the magnitude of the change in objective function and/or a decrease in magnitude of the distance between solutions. For these experiments, the search space (i.e. set of candidate solutions) is identical across k. The solutions are fully enumerated, and so the resulting Hamming distances between all pairwise solution combinations are also identical across k. However, not all of the conceivable distances between solutions (i.e. all pairwise distances)

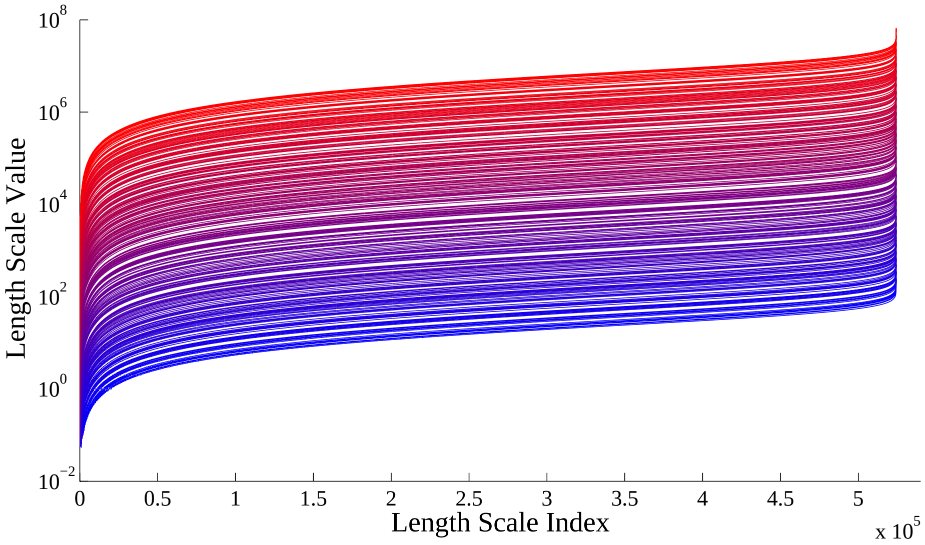


Figure 7.78: Length scale sets for the NPP instances. Lines are coloured according to k, with blue representing $k = 0.4$ and red representing $k = 1.3$.

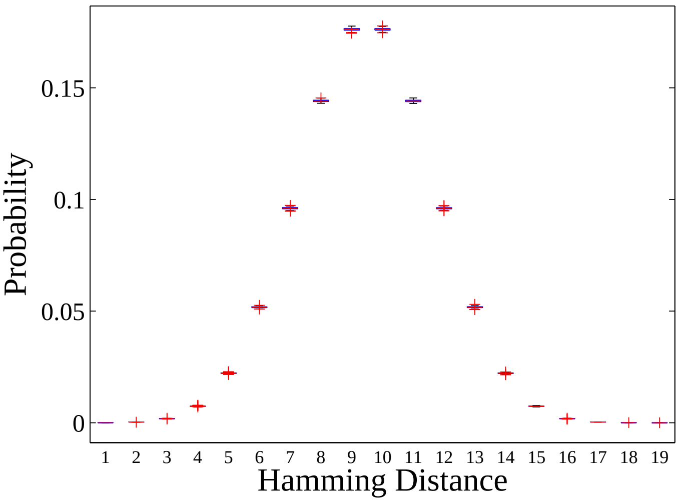


Figure 7.79: Hamming distance distributions for the NPP instances.

are used in these experiments, instead, a random subset (due to the use of Algorithm 6.1) is used. Therefore some variation in the Hamming distances may occur, however this is expected to be low. To investigate how much variation there is in the subsets' distances, Figure 7.79 shows a box and whisker plot of the Hamming distance probabilities across all of the random subsets used in the experiments. Because 10 instances are generated at each value of k, there are a total of 10×37 random subsets. Note that since Hamming distance is calculated between unique candidate solutions, the distance distribution is a probability mass function with probabilities defined at distances of $1, \ldots, 19$.

Figure 7.79 shows that, as expected, the Hamming distance probabilities have very small variance across the random subsets. Therefore, the increase in magnitude of length scale

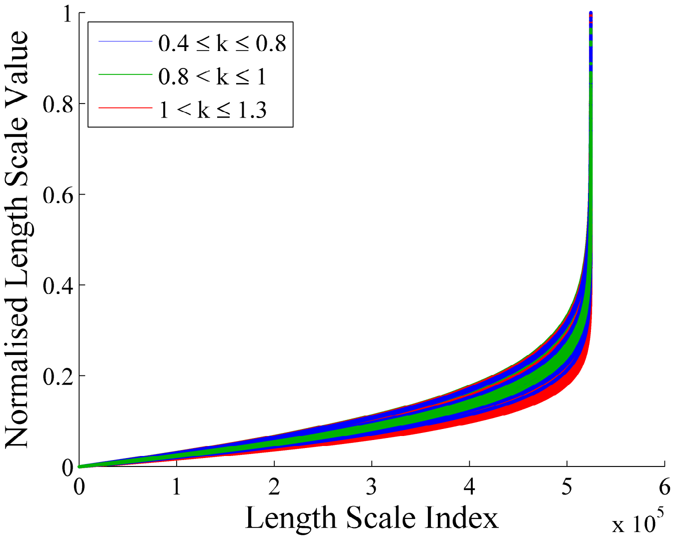


Figure 7.80: Length scale sets for the NPP instances normalised by 2^{nk}. Lines are coloured according to k, with blue representing $0.4 \leq k \leq 0.8$, green representing $0.8 < k \leq 1$ and red representing $1 < k \leq 1.3$.

values exhibited in Figure 7.78 can only be due to an increase in the difference of objective function values. This makes sense with some reflection of the problem definition. The objective function is based on the discrepancy of the integers in S_1 and S_2, which are generated from randomly sampling $\{1, \ldots, m\}$, $m = 2^{nk}$. Thus for each increase in k, m increases by a factor of 2^{nk}. This results in larger integers in S_1 and S_2, which leads to larger objective function values and larger length scales. However, one must be careful not to conclude that the change in f as k increases is solely caused by the scaling of 2^{nk}. While the scaling is certainly a contributing factor, so too is the nature of the landscape structure, which may change as k increases. To illustrate this point, Figure 7.80 shows the length scales in Figure 7.78, normalised by 2^{nk}.

The normalised length scale sets exhibit different values across k. Here it is clear that instances after the phase transition (coloured in red) have a wide variety of length scale values, while instances prior to the phase transition (coloured in blue) have a smaller variety of (larger) length scale values. The length scales of instances close to the phase transition (coloured in green) do not vary as much as the aforementioned sets. This can also be observed from the length scale distributions, shown in Figure 7.81.

Overall, Figures 7.80 and 7.81 illustrate that k highly influences the value and variety of the length scales for an instance. Given that k affects the length scales, length scale information may potentially be used to infer k, and hence, the particular area of the phase transition that a given problem belongs to. To evaluate the discriminatory power of the

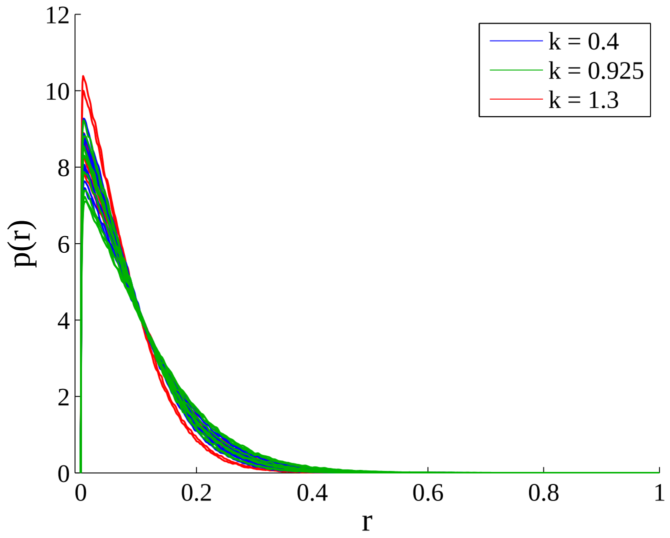


Figure 7.81: Example $p(r)$s for the 10 NPP instances normalised by 2^{nk}. Lines are coloured according to k, with blue representing $k = 0.4$, green representing $k = 0.925$ and red representing $k = 1.3$.

length scales, a heatmap summarising the average D_J between problems, a t-SNE (perplexity of 5, best result from 1000 trials with a maximum of 1000 iterations) visualisation of the problem space and dendrogram are shown in Figures 7.82 to 7.84. To obtain the visualisations, the J-divergences were calculated for all pairs of problems within each of the 10 sets of problems, and the results were averaged. For example, given 10 instances of NPP at $k = 0.4$ and $k = 0.425$, $D_J(k = 0.4, k = 0.425) = \frac{1}{10}\sum_{i=1}^{10} D_J^i(k = 0.4, k = 0.425)$, where D_J^i is the J-divergence between instances i.

The heatmap in Figure 7.82 is very different to the heatmaps for the elliptical functions (Figure 7.2), Rastrigin functions (Figure 7.6), TSPLib instances (Figure 7.69) and ATSPs (Figure 7.75). The leading diagonal is very dark, indicating that problems produced by very similar values of k, such as $k = 0.4$ and $k = 0.425$, are structurally very similar. However, the D_J values slightly underneath the leading diagonal are white, indicating that the problems are structurally very different in relation to the other problems. The light colour follows linearly (as a diagonal line) down the heatmap, and is positioned approximately 5 instances away from the leading diagonal. The increment of k between problems is 0.025, meaning given a problem at k_i, problems at approximately $k_i + 5 \times 0.025$ are very different. For example, $k = 1$ and $k = 1.125$ are structurally very different. The instance pairings below the white diagonal are grey coloured, with a slightly lighter shade of grey towards the lower left corner ($k = 0.4$ and $k = 1.3$). The grey colouring suggests that the instances are moderately different compared with the other instances in the set. Therefore, instances with different k

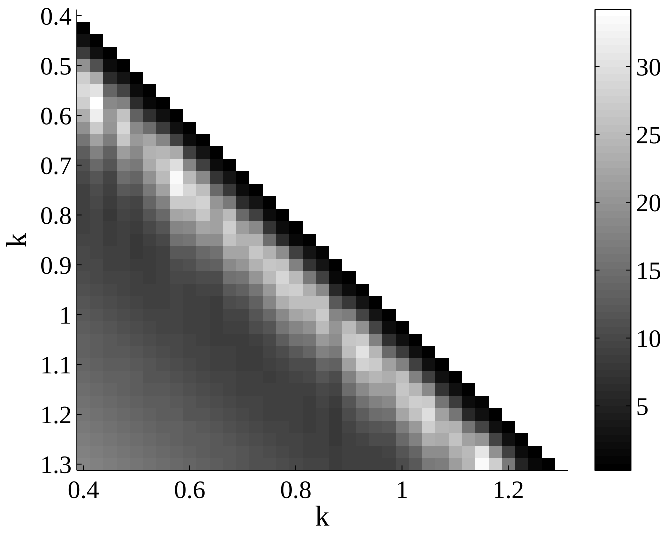


Figure 7.82: Heatmap of D_J values calculated between pairs of NPP instances with $k = 0.4, 0.425, \ldots, 1.3$.

values, such as $k = 0.4$ and $k = 1.3$ are structurally dissimilar, with a greater dissimilarity between larger discrepancies in k. Overall, the heatmap suggests that the instances are generally structurally dissimilar, with the exception of instances with close (but not *too* close) k values.

The final cost of the t-SNE optimization was 0.2272, indicating that the visualisation in Figure 7.83 is a moderately accurate representation of the J-divergences between the instances. The visualisation clearly shows a relationship with k; instances are generally ordered from light (low k values) to dark (high k values). There are two particularly interesting regions in Figure 7.83: the change in direction at approximately $k = 0.9$ and the change in direction at approximately $k = 1.15$. The phase transition between easy and hard landscapes reportedly occurs at approximately 0.9153. Remarkably, the t-SNE visualisation reflects this transition; the instances follow a linear pattern as k increase from 0 to 0.9, suggesting that the changes in problem structure are gradual and smooth. Then, at 0.9 the location of the instances changes significantly, suggesting that for $k > 0.9$, the instances are structurally very different to previous instances. The locations of the instances continues to change as k increases, indicating that the instances continue to vary significantly with structure.

The dendrogram, shown in Figure 7.84 exhibits similar relationships between instances as discussed above. Problems of similar k values have a small J-divergence between them. In particular, the J-divergences between groups of 4 instances are quite low, however merging groups of 4 problems generally requires a considerably increase in J-divergence (e.g. $D_J > 9$). The large increase in merging clusters of 4 problems likely corresponds to the large J-

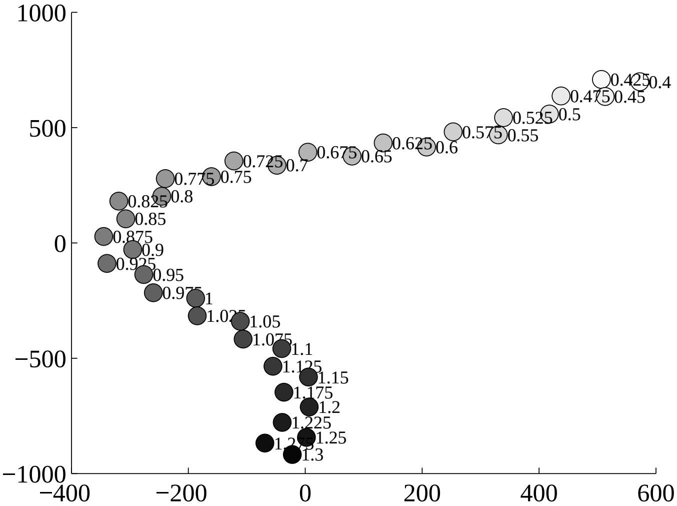


Figure 7.83: t-SNE of D_Js (cost of 0.2272) calculated between pairs of NPP instances with $k = 0.4, 0.425, \ldots, 1.3$.

divergences between problems 5 instances apart, shown by the white diagonal (below the leading diagonal) of the heatmap in Figure 7.82. Overall, the dendrogram is very balanced; problems with similar k values form small clusters, which in turn combine to form larger clusters, each containing a similar number of instances.

7.6.2 Existing Results

Unlike the Travelling Salesman Problem, the performance of polynomial-time heuristics, such as the Karmarkar-Karp Differencing Algorithm, is incredibly poor for large instances of the Number Partitioning Problem [17]. In order to try and understand why the NPP is difficult for heuristics, the landscapes of NPP instances have been analysed using techniques from statistical mechanics and traditional problem landscape analysis. However because much of the analysis relies on complete enumeration of solutions, small instances (i.e. $10 \leq n \leq 20$) are typically examined.

A barrier tree is a tree constructed for a landscape where leaves represent local minima and internal nodes represent the fittest saddle points connecting local minima [171]. Hence, barrier trees represent much of the information regarding valley structures (i.e. the path from one local optimum to another) inherent to a landscape. The construction of a barrier tree requires complete enumeration of $\mathcal{S}$, and is therefore only practically feasible for small landscapes. Stadler et al. [174] constructed barrier trees for NPP landscapes and calculated a variety of tree properties to determine if the transition between "easy" and "hard" problems could be captured. Specifically, NPP instances were generated with elements uniform

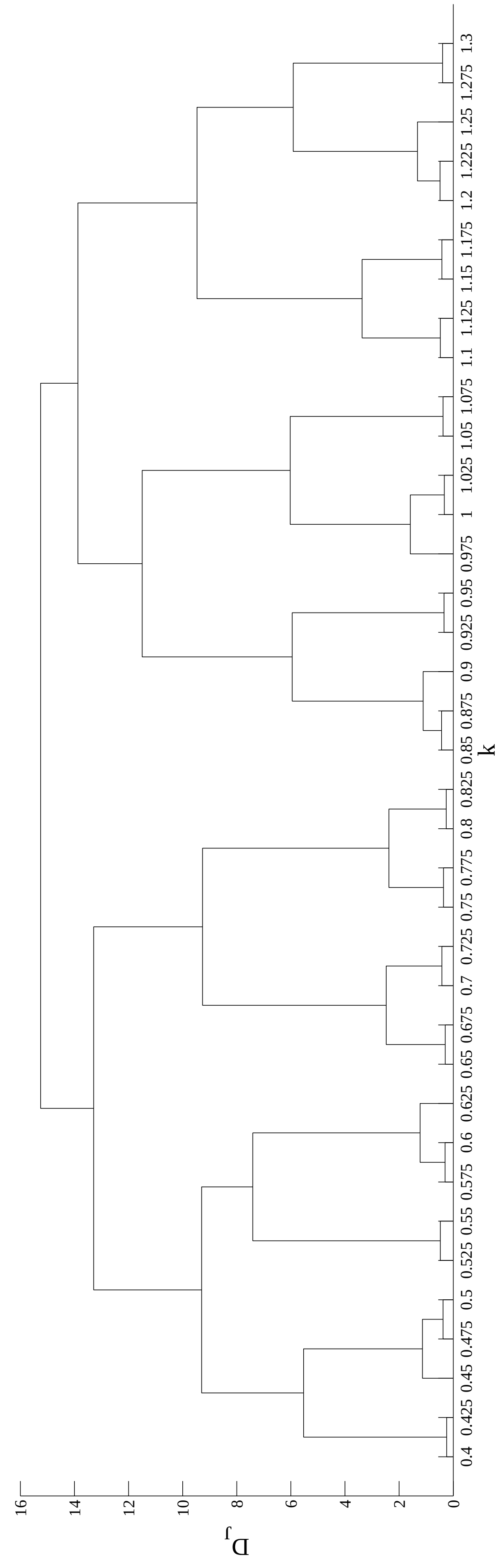


Figure 7.84: Dendrogram of D_Js (cost of 0.2041) calculated between pairs of NPP instances with $k = 0.4, 0.425, \ldots, 1.3$.

randomly sampled from an interval of integers, for problems of size $n = 12, 14, 16, 18$ and 20. Then, five measures of barrier tree symmetry and balance were calculated on 100 random instances at each n. The results showed that all five measures, along with the fraction of local minimum, remained constant across the phase transition. However, a single property, imprudently coined the *difficulty*, was able to detect traces of the phase transition. In this context, the "difficulty" is defined as the maximum of the ratio between the height of the node connecting the global optimum and a local optimum, and the objective function value difference between them, and it is directly related to the optimal convergence speed of simulated annealing [32]. Stadler et al. [174] also remarked that the barrier trees strongly resembled random trees and become completely balanced as $n \to \infty$, and that consequently, local search techniques are unlikely to produce good solutions to the NPP.

Klemm et al. [89] investigated the behaviour of local optima, with regards to their prevalence in funnels for the NPP. Problems of size $n = 8, 10$ and 12 were examined, with 30 instances generated by uniform randomly choosing integers from an interval for each n. Their results showed that the fraction of local optima within funnels is inversely proportional to the total number of local optima. The relationship between their findings and the known phase transition is not explored or discussed.

The correlation between similar objective function values and their corresponding solutions for NPP instances is theoretically and empirically examined in [8]. 10000 instances are randomly generated for $n = 20$, and the solutions of similar f-values are found to be uncorrelated. The lack of correlation means that the landscape is "locally random", and so it is difficult to generate solutions with slightly better objective function values than the current solution. Consequently, convergence is often difficult for local search heuristics on NPP instances.

Recently, Alyahya and Rowe [6] examined the number of minima and plateaus, as well as basin size and its correlation with f-values on NPP instances of size $n = 20$. Two different neighbourhood definitions are used; solutions at a Hamming distance equal to one, and less than or equal to two. 30 instances are generated for $k = 0.4$ to $k = 1.3$ at increments of 0.1, and integers are sampled randomly from 5 different distributions. The results show that the number of global minima and the number of plateaus change with respect to k. Specifically, "easy" problems have numerous global optima and large plateaus, while "hard" problems have only 2 global optima, and small plateaus.

None of the aforementioned properties are able to detect phase transitions based on black-box information. More specifically, all of the existing properties rely on knowledge of the global optimum, knowledge of all local optima and/or enumeration of all possible solutions, which is difficult and computationally time consuming in practice [6, 89, 174]. In contrast to the existing landscape analysis on the NPP, the length scale analysis performed in this section relies solely on the information available in the black-box scenario, i.e. solutions and their respective objective function values. The analysis is clearly able to detect the presence of a phase transition near the critical parameter, $k_c = 0.9153$ for $n = 20$. In addition, the analysis of the length scale distributions showed that instances prior to the phase transition have less "large" length scales than instances after the phase transition. This indicates that the "easy" problems have less sporadic changes in objective function values, and are therefore smooth in comparison to the "hard" problems. This supports the results in [6], which suggest that "easy" instances have larger plateaus, and hence less rugged regions than "hard" problems.

7.7 Summary

The length scale framework developed in Chapters 5 and 6 was used in this chapter to analyse a variety of continuous and combinatorial optimization problems. In particular, continuous artificial problems, the BBOB problem set, CiaS packing problems, TSPLib, ATSP and NPP instances were analysed using a variety of techniques including plots of length scale sets, length scale distributions, heatmaps, t-SNE problem space visualisations and dendrograms of clusters produced by hierarchical clustering.

The summaries of length scale information (such as $p(r)$ and $h(r)$) were shown to be statistically robust to different samples of given problem instances. The stability/robustness of length scale analysis can be attributed to its underlying methodology, where information regarding both the solutions *and* their objective function values are combined and effectively utilised. In comparison, many existing landscape analysis techniques ignore potentially important information. For example, information content is purely concerned with the variety of objective function fluctuations in the landscape; available information such as the magnitude of the fluctuations or location of the fluctuations is ignored. Likewise, FDC is concerned purely with the correlation between the objective function values of solutions and their distance to a reference solution. While FDC does incorporate both the solutions and their objective function values, it is with regard to a reference solution. Available information, such as the distances *between solutions*, is ignored.

The length scale analysis of the artificial continuous functions demonstrated the ability of the framework to capture important landscape structures such as eccentricity and modality. Experimental results on the BBOB and CiaS problem sets showed that the length scale analysis provides a greater ability to discriminate between the problems in comparison to seven well-known landscape analysis techniques. While all the landscape features analysed provide insight into the nature of the problems, correlation length, FDC, information content, partial information content and dispersion were found to be limited in their ability to characterise and differentiate the problems.

The length scale analysis of Travelling Salesman Problems and Number Partitioning Problems demonstrated the flexibility of the length scale framework, and in particular, how easily it can be applied to combinatorial problems with varying notions of distance. In addition to the successful application to combinatorial problems, some incredibly powerful insights were drawn. Firstly, the length scale analysis of symmetric and asymmetric instances from the TSPLib benchmarking set was able to categorise the TSPLib instances based on their source. Secondly, length scale analysis on Asymmetric TSPs (ATSPs) generated along a known phase transition was able to detect the phase transition. In a similar experiment, the well-documented phase transition of NPP instances was also captured using the length scale analysis. It is imperative to emphasize that all experiments treated the problems as a black-box, and so the categorisation of TSPLib instances and detection of the phase transitions is based *purely* on black-box information.

CHAPTER 8

Quantifying Problem Similarity with Information Distance

The eternal mystery of the world is its comprehensibility.

Albert Einstein

In Chapter 6, it was shown that the length scale analysis framework can be used effectively to explicitly quantify optimization problem similarity via the Jeffrey divergence (J-divergence) between length scale distributions. This chapter proposes an alternative utilization of the framework for quantifying problem similarity. The proposed approach is based on the notion of *Information Distance*, a universal distance function originating in Kolmogorov Complexity theory as a measure of the (dis)similarity in information between two arbitrary objects. While Information Distance is a theoretical measure, in practice it can be approximated by the *Normalised Compression Distance* (NCD). The chapter begins with a review of Kolmogorov Complexity theory, with a particular emphasis on Information Distance and the NCD. In Section 8.2, a unique methodology for calculating the NCD between optimization problems is developed, and practical considerations are discussed. The ability of the NCD to accurately measure problem similarity is experimentally investigated in Section 8.3. In particular, the NCD values are calculated between artificial continuous problems, the BBOB problem set, Circle in a Square (CiaS) packing problems, Travelling Salesman Problems (TSPs) and Number Partitioning Problems (NPPs). Both the NCD proposed in this chapter and D_J proposed in Chapter 6 are novel measures of optimization problem similarity, and so the relationship between the two measures is also theoretically and empirically examined in Section 8.4. The chapter concludes with a summary of its contributions in Section 8.5.

8.1 Kolmogorov Complexity Theory

Kolmogorov Complexity theory (also known as *algorithmic information theory*) encompasses many theories and techniques for measuring and comparing the information in objects such as pictures, music and even optimization problems. This section reviews fundamental theoretical concepts from the literature that are required for the application of Information Distance to quantifying optimization problem similarity, proposed in subsequent sections. An overview of Kolmogorov Complexity and its application in the optimization context is given, and the notion of Information Distance, a universal distance function, is formally outlined.

8.1.1 Kolmogorov Complexity

Given a finite binary string, x, the Kolmogorov Complexity, $K(x)$, quantifies the string's information content or *complexity*. Formally, K is defined as a function, $K : \{0,1\}^* \rightarrow \mathbb{N}$, from finite binary strings of arbitrary length to the natural numbers, $\mathbb{N}$ [96]. K has been extended to other representations such as sets and functions [68], however this thesis focuses on the complexity of finite binary strings. Informally, $K(x)$ is measured by the minimum number of bits required to completely and unambiguously describe x. A "description" is defined as any program running on a Universal (prefix) Turing Machine that prints the string and halts. $K(x)$ is then the length (in bits) of the *smallest* such program. Programs written in any universal programming language (e.g. C, Java or Python) can be transformed to run on a Universal Turing Machine. Formally, let p be a program that prints a binary string and halts, and let $l(p)$ be the program's length. Then, the Kolmogorov Complexity of x with respect to a Universal (prefix) Turing Machine U is:

$$K_U(x) := \min_{p:U(p)=x} l(p) \tag{8.1.1}$$

A more rigorous definition and further details can be obtained from [68, 96], but the above definition is adequate for the purpose of this work. For a simple, well-structured string, the shortest program will capture and compress exploitable structure, and so the resulting program length (and hence complexity) will be small in relation to string's length. For example, consider the string "1111111111'; such a string is intuitively simple and there exists a small program (e.g. akin to "print 10 1s') that prints the entire string and halts. Indeed, the string of 1s can be made arbitrarily large while the program remains relatively very small, e.g.

“print 2^{100} 1s’. Now consider the string “0010111010’; this string has very little exploitable structure, and so the smallest program will likely have to print the entirety of the string, e.g. “print 0010111010’. Here, the length of this program is essentially the length of the actual string, and so the complexity is high.

8.1.2 Information Distance

Information Distance is a universal distance function based on the notion that the (dis)similarity between two objects can be quantified by the minimal amount of information required to transform one object into the other, and vice versa [11, 96]. Intuitively, Information Distance is the length of the shortest program that: 1) transforms x to y when given y as auxilary input; 2) transforms string y to string x when given x as input and; 3) halts. Consider such a program in more detail; if x and y are identical strings, the program is extremely small because no conversion is required (x is already converted to y and vice versa). However, if x and y are not identical but share *some* mutual information, then the program size is moderate; the shared information can be ignored (as no conversion is required), but some conversion is required between the unshared information. Finally, if x and y differ completely, a program is required to convert all the information in x to y, and vice versa. This program is unable to utilise shared information, and as a result it is relatively large.

Formally, the Information Distance is defined as [97]:

$$ID(x, y) = \min \{l(p) : U(p, x) = y, U(p, y) = x\} \tag{8.1.2}$$

where $U(p, x) = y$ denotes running program p on Universal Prefix Turing Machine U with input string x, such that the output is string y.

The Information Distance is an absolute distance, and indicates the total difference in the information between strings x and y. If two strings comprised of 10^6 bits differ by only 100 bits, intuitively, they are relatively quite similar. In contrast, if two strings comprised of 100 bits differ in all 100 bits, they are intuitively very different. The Normalised Information Distance (NID) normalises the Information Distance to allow relative comparisons [97]:

$$NID(x, y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}} \tag{8.1.3}$$

where x and y are two finite binary strings and $K(x|y)$ and $K(y|x)$ are the (conditional) Kolmogorov Complexities of x given y as input, and y given x as input respectively.

The NID is a relative distance metric, meaning two strings of differing sizes can be compared and the resulting distance is in $[0, 1]$. A value of 0 indicates that the strings are identical in terms of information, while 1 indicates the strings are maximally different.

8.1.3 Normalised Compression Distance

Kolmogorov Complexity is uncomputable and thus cannot be directly used in practice. Information Distance and Normalised Information Distance rely on K, and so they too cannot be directly used in practice. However, because K relies on the smallest description of x, *any* description of x (including itself) will form an upper bound on its Kolmogorov Complexity. Therefore, approximations of K can be made using *lossless* compression algorithms. Given an input string, lossless compression algorithms produce an encoded string that can be unambiguously decoded without loss of information [104]. The encoding is essentially a description of the string, and hence the encoding's length forms an upper bound on the string's complexity. To achieve unambiguous decoding, lossless compression algorithms map each unique input string to a unique encoding, meaning there is a one-to-one mapping between input strings and encodings. Consequently, for each compressible string (i.e. its encoding is smaller than itself), there exists an incompressible encoding (i.e. its encoding is larger than itself). However, it is important to note that for incompressible input strings where the encoding is larger than the original, most compressors will output the original string, rather than the (larger) encoding.

Given a lossless compressor, Z, $K(x)$ can be approximated by the length of the compressed x, denoted by $|Z(x)|$. Hence, by substituting $K(\ldots)$ with $|Z(\ldots)|$, the Normalised Compression Distance (NCD) can be used to approximate the Normalised Information Distance [36, 97]:

$$NCD(x, y) = \frac{|Z(xy)| - \min\{|Z(x)|, |Z(y)|\}}{\max\{|Z(x)|, |Z(y)|\}} \tag{8.1.4}$$

where xy is the concatenation of strings x and y. Here, the term $\max\{K(x|y), K(y|x)\}$ from (8.1.3) has been substituted with $(K(xy) - \min\{K(x), K(y)\})$ as they are equal up to an additive logarithmic term, $O(\log K(xy))$, which is commonly ignored [36, 37, 194]. This effectively means that provided a suitable compressor and concatenation operator exist, the NCD can be used to approximate the NID between two finite arbitrary binary strings.

NCD operates under the assumption that the compressor used is *normal*, meaning that Z satisfies the following axioms up to an additive term $O(\log n)$, where n is the maximum length of the input:

1. Idempotency: $|Z(xx)| = |Z(x)|$ and $|Z(\epsilon)| = 0$, where ϵ is the empty string
2. Monotonicity: $|Z(xy)| \geq |Z(x)|$
3. Symmetry: $|Z(xy)| = |Z(yx)|$
4. Distributivity: $|Z(xy)| + |Z(w)| \leq |Z(xw)| + |Z(yw)|$

It is assumed that most popular compression algorithms in use closely adhere to the normality properties, though this is rarely tested experimentally. Cebrian et al. [33] investigated the normality of GZIP, BZIP2 and PPMZ on the Calgary Corpus and it was found that certain block-sizes of GZIP and window sizes of BZIP2 can violate the idempotency axiom. The Calgary Corpus is mainly comprised of text data, and so the degree to which the compressors adhere to the normality properties on other data types (e.g. images or songs) is unknown.

The accuracy of NCD can also be affected by how well the string representation captures the information in the object and the ability of the compressor to exploit structural regularities in the representation. For example, consider computing the NCD between two pieces of piano music. Each piece can be represented in a variety of ways; one could use a sound recording of the piece, a listing of the notes played, or even an image of the sheet music. These representations induce specific structural features in their respective binary representations, and so it is important to use a compressor that is capable of exploiting such structures. For example, if a sound recording representation is chosen, compressors specifically designed for sound data, such as the Free Lossless Audio Compressor (FLAC), are likely to obtain good compression. In certain cases, satisfactory results may be achieved using general-purpose compressors. For example, [194] used GZIP in computing the NCD between a heterogeneous dataset consisting of four mitochondrial gene sequences, four excerpts from a novel, four MIDI files, two Linux executables and two compiled Java classes. The resulting distances clustered objects into their respective "types"*and* distributed the objects according to their inherent similarities. For example, their results showed that Musical Instrument Digital Interface (MIDI) files of music were clustered together, and within this cluster, MIDI files of Jimi Hendrix were well-separated from MIDI files of movements from Debussy's "Suite Bergamasque").

NCD has been found to be competitive with (and often superior to) many state-of-the-art specialised techniques in anomaly detection, classification, and clustering [86]. NCD has been successfully used in applications from a wide variety of areas including bioinformatics [52, 90, 97], linguistics [36, 97], music classification [37] and plagiarism detection [35].

8.2 Normalised Compression Distance Between Optimization Problems

The Normalised Information Distance is a universal distance metric, meaning that it is applicable to any two objects, and in practice it is approximated by the NCD. The following section proposes the application of the NCD to quantify the similarity between optimization problems. Crucial components of NCD include a suitable, binary representation of the objects (e.g. ASCII for texts) and a compressor (e.g. GZIP). Therefore to begin, a suitable representation, based on the *length scale*, r, developed in Chapter 5, is described. Following this, a range of generalised and type-specific compressors are experimentally investigated to evaluate their ability to compress length scale information. The methodology developed in this section is a significant and novel contribution, as it allows the explicit quantification of problem similarity, based purely on the information available in the black-box setting.

8.2.1 Representing Optimization Problems with Length Scale

In the black-box scenario, information regarding a problem is limited to the solutions in a search space, $\mathbf{x} \in \mathcal{S}$, and their respective objective function values, $f(\mathbf{x}) \in \mathbb{R}$. Using this information, an objective function can be unambiguously described via a string of concatenated objective function values, evaluated on a finite, indexed sample of the search space [20, 50]. The indexing of the sample is very important, as it provides an explicit ordering of the solutions and ensures that the description string is unique for each objective function. Without an indexing, problems with a common set of f values will yield identical representations, regardless of the *distribution* of the f values over $\mathcal{S}$ (and hence problem structure). The description resulting from an indexed enumeration of f facilitates Kolmogorov Complexity analysis, and has subsequently been used in the development of No Free Lunch theorems [47, 177] and investigating the connection between problem complexity and algorithmic performance [20, 50]. Hence, while Kolmogorov Complexity analysis has been previously used in the optimization context [20, 50], it is largely restricted to theoretical applications and the analysis of simple artificial problems.

Determining an index/order of solutions that is appropriate for a wide variety of problems is difficult in practice. Such an index must be appropriate for problems with varying search space bounds, dimensionality and solution representation. A fixed index, such as lexicographic ordering of solution vectors, can be used for problems with common dimensionalities and solution representations and where bounds can easily be re-scaled/normalised. However, such a method precludes comparisons of problems with differing dimensionality, bounds and potentially, representation. Furthermore, the index would ideally produce a sequence of objective function values that maintains the regularities and structure within the problem, but for many problems (e.g. black-box), no such index is known a priori [20]. Indeed, the index can potentially change or obfuscate problem regularities and structures. For example, lexicographic ordering of problems differing by a simple transformation of the search space - such as a rotation or reflection - will yield different string representations (and hence differ in their NCD), despite the fact that they are identical from an information perspective.

In order to quantify the distance between *arbitrary* optimization problems, a representation is required that is not reliant on the search space bounds, dimensionality or sample type/order. Therefore, it is highly advantageous that the length scale (Definition 5.1) satisfies these properties (see Section 5.2). Length scale is an indicator of functional equivalence over pairs of solutions. Problems with local regions of equivalence will have a partition of r values in common, and so problem similarity can be measured by the degree to which problems share common r values. In Section 6.1.2, this idea was explored using the notion of the Variation of Information (VI) between length scale multisets. The VI is similar to NID, and theoretical and empirical comparisons in the context of clustering validation and comparison concluded that NID satisfies more desirable properties (i.e. NID is a normalised metric with tight theoretical bounds) than other commonly used clustering validation and comparison measures including VI, the Rand index and Mutual Information [192].

Perhaps the most important and useful property of the length scale multiset is that it provides a description of the landscape that is invariant to the order in which solutions are sampled. In other words, the length scale multiset can be used as a unique representation of optimization problems, without requiring an index/order over the solutions sampled. Consequently, the r values are well-suited to Kolmogorov Complexity analysis, and in particular, the application of the NCD.

Algorithm 8.1 proposes a novel the methodology for calculating the NCD between two optimization problems. The methodology requires only a sample of solutions from each problem, their respective objective function values and a lossless compressor. The samples of each function are used to generate length scale multisets (denoted $\mathbf{r}_a$ and $\mathbf{r}_b$). The ordering of length scale multisets is arbitrary by definition, and so $\mathbf{r}_a$, $\mathbf{r}_b$ and $\mathbf{r}_a \cup \mathbf{r}_b$ are sorted to aid in compression. The resulting length scale multisets (and their union) are compressed, and the NCD is calculated by substituting the size of the compressed-binaries into Equation 8.1.4.

Algorithm 8.1 Normalised Compression Distance (NCD)

```
Input:
    Sample of solutions, $\mathcal{S}'_a \leftarrow [\mathbf{x}^1, \ldots, \mathbf{x}^n]$
    Sample of solutions, $\mathcal{S}'_b \leftarrow [\mathbf{x}^1, \ldots, \mathbf{x}^m]$
    Objective function, $f_a : \mathcal{S}'_a \rightarrow \mathbb{R}$
    Objective function, $f_b : \mathcal{S}'_b \rightarrow \mathbb{R}$
    Lossless normal compressor, $Z$
$\mathbf{r}_a \leftarrow LengthScales(\mathcal{S}'_a, f_a)$
$\mathbf{r}_b \leftarrow LengthScales(\mathcal{S}'_b, f_b)$
$\mathbf{b}_a \leftarrow SaveAsBinary(Sort(\mathbf{r}_a))$
$\mathbf{b}_b \leftarrow SaveAsBinary(Sort(\mathbf{r}_b))$
$\mathbf{b}_{ab} \leftarrow SaveAsBinary(Sort(\mathbf{r}_a \cup \mathbf{r}_b))$
$ncd \leftarrow \frac{|Z(b_{ab})| - \min\{|Z(b_a)|, |Z(b_b)|\}}{\max\{|Z(b_a)|, |Z(b_b)|\}}$
return $ncd$
```

8.2.2 Suitable Compression Algorithms

While arbitrary data can be compressed using general-purpose compression algorithms (e.g. GZIP), many lossless compressors are designed for particular types of data (e.g. TIFF is specifically designed for image data). Type-specific compressors achieve good compression by exploiting structures and regularities that are commonly found within the targeted data type [148]. As a result, input strings from the intended domain are generally mapped to small encodings, while strings outside the intended domain may be mapped to large encodings. Hence good compression is generally achieved in practice, and incompressibility often occurs when atypical data is given as input (e.g. TIFF applied to text data), or when the input data is unstructured/random.

By definition, $r \in [0, \infty]$, but in practice, the length scale value between two solutions is calculated and stored using a finite precision floating point number. In this thesis, length scales are represented by the IEEE Standard 754 for double-precision floating points, meaning the length scale multiset is an ordered sequence of IEEE 754 double-precision (64-bit) floating point values.

Both generalised and type-specific compression algorithms are available for compressing sequences of 64-bit floating point values. Current popular and state-of-the-art generalised compression algorithms include GZIP , BZIP2, LZMA and PPMd [148]. While generalised compressors have no explicit heuristics to detect and uncover complex numerical patterns and regularities in the length scale multiset, they may be successful at capturing more general regularities, such as repeated bit patterns. For example, length scales of similar magnitudes will yield similar exponents in the IEEE 754 double-precision representation. Under these assumptions, a generalised compressor could potentially achieve good compression by exploiting repeated bit-patterns (stemming from similar exponents) in the bit-string representation of the length scale multiset.

Few lossless compression algorithms have been developed specifically for sequences of 64-bit floating point numbers; FSD [49] is particularly suited to compressing gradually changing sequences, PLMI [98] was developed specifically for compressing 2D and 3D sequences, while DFCM [134] focuses on compressing arbitrary sequences quickly. An extension of the work in Ratanaworabhan et al. [134] is FPC: a lossless, single-pass, linear time compressor [25]. FPC is parameterised by a single parameter, *table size*, that controls the size of two predictors (that are effectively hash tables) used by the algorithm. Increasing the table size generally achieves better compression, but with the caveat of slower speed. Parallelised and self-tuning versions of FPC have also been developed [26, 27], although initial explorative experiments conducted for the experiments in this chapter showed little improvements in compression, at a greater effort in implementation and computational cost. FPC has been compared to BZIP2, GZIP, DFCM, FSD and PLMI on a variety of datasets, and it generally achieves competitive or superior compression ratios using a factor of 2 to 300 less time [25].

Given that there are many lossless compression algorithms available, it is not immediately clear which is the most practically suitable for the length scale data (i.e. a multiset of increasing double-precision floating point values). The degree of compression achieved, computational resources required and ease of implementation are all important considerations when selecting a compressor. Therefore, the following experiment investigates the ability of GZIP, BZIP2, LZMA, PPMd and FPC to compress length scale data. The degree to which each compressor satisfies the *normality* properties defined in Section 8.1.2 is also assessed. To obtain a reasonable representation of length scale multisets, length scales are calculated from the 10 dimensional Griewank, Michalewicz ($m = 20$), Rastrigin ($A = 10$) and Rosenbrock problems, as defined in Table A.1. For each problem, 30 Lévy random walks of length $2.5 \times 10^5 D^2$ are used to generate 30 length scale multisets. Thus there are

	GZIP	**BZIP2**	**LZMA**	**PPMd**	**FPC**
Griewank	0.6433	0.7716	0.5608	0.8729	0.6836
Michalewicz	0.6374	0.7679	0.5550	0.8700	0.6829
Rastrigin	0.6352	0.7661	0.5529	0.8684	0.6826
Rosenbrock	0.6351	0.7661	0.5526	0.8687	0.6826

Table 8.1: Compression Ratios

a total of 4 × 30 representative length scale multisets. The walks were parameterised by $\gamma = 0.0005 \times range(\mathcal{S})$ and $\delta = 0$.

The compression ratio for a given input string x is defined as $\frac{|Z(x)|}{|x|}$, that is, the length of the encoding divided by the length of the input [148]. Small compression ratios indicate that the compressor is able to exploit structural regularities in the data. Using each compressor, the compression ratio was calculated for each of the 30 length scales multisets across the problems. To achieve maximum compression, FPC's table size was set to 2^{30} bytes, and 7zip's [127] implementations of GZIP, BZIP2, LZMA and PPMd were used with the "ultra"compression setting. Table 8.1 contains the average compression ratio for each compressor (on each problem) across the 30 walks.

The standard deviations for the compression ratios across the walks were very low and ranged between 5.9784×10^{-6} (FPC over the Griewank walks) and 4.1977×10^{-4} (LZMA over the Michalewicz walks), indicating that the compressors give consistent results across the different walks. The compression ratios for each compressor do not vary significantly between problems, and this is likely because all of the inputs are the same length and consist of increasing sequences of 64-bit floating points. LZMA achieved superior compression ratios compared to the other compressors. Despite being designed specifically for 64-bit floating point data, FPC achieved only the third best compression ratio.

As previously discussed in Section 8.1.2, the accuracy of the NCD also relies on the degree to which the normality properties are satisfied. For an idempotent compressor, $|Z(\epsilon)| = 0$ and $\frac{|Z(xx)|}{|Z(x)|} = 1$ for input x. Likewise, a symmetrical compressor will yield $\frac{|Z(xy)|}{|Z(yx)|} = 1$ for inputs x and y. There are 4 different problems, each with 30 walks in these experiments. Using the length scale multiset for each walk as inputs, there are 4 × 30 different combinations of xx and $4 \times 30 \times 29$ different combinations of xy (and yx). The degree to which each compressor is idempotent and symmetric is calculated by averaging $|Z(\epsilon)|$, $\frac{|Z(xx)|}{|Z(x)|}$ and $\frac{\max\{|Z(xy)|,|Z(yx)|\}}{\min\{|Z(xy)|,|Z(yx)|\}}$ over the different samples. The monotonicity and distributivity

	GZIP	BZIP2	LZMA	PPMd	FPC
$\|Z(\epsilon)\|$ **(bytes)**	30	14	86	86	1
$\frac{\|Z(xx)\|}{\|Z(x)\|}$	2.0000	2.0000	1.9999	2.0008	1.9995
$\frac{\max\{\|Z(xy)\|,\|Z(yx)\|\}}{\min\{\|Z(xy)\|,\|Z(yx)\|\}}$	1.0000	1.0000	1.0000	1.0000	1.0018
Monotonicity violations (%)	0	0	0	0	0.0287
Distributivity violations (%)	0	0	0	0	0.0287

Table 8.2: Evaluation of the normality properties

properties are tested in a similar manner; all possible combinations of samples are generated and the number of times each property is violated is counted. Table 8.2 displays the ratios for the idemonpotent and symmetry properties as well as the monotonicity and distributivity violations as a percentage of the total number of samples.

Table 8.2 shows varying levels of compliance to the idempotent property; all of the compression algorithms produced small outputs for the empty string (the largest being LZMA and PPMd at 86 bytes), however all failed to detect that xx was a repeated dataset. Symmetry was observed very well across the compressors; all compressors produced outputs that differed in size by less than 1%. Encouragingly, GZIP, BZIP2, LZMA and PPMd all completely complied with the monotonicity and distributivity axioms, and although FPC violated the axioms, it did so in only 0.0287% of the datasets.

Overall, GZIP, BZIP2 and LZMA achieved good compression ratios and closely adhered to the normality properties. While PPMd complied with normality, its compression ratios were not as competitive as the other compressors. In contrast, FPC achieved good compression ratios, but slightly violated both the monotonicity and distributivity properties. All of compressors tested in these experiments are arguably suitable for compressing the length scale data, particularly with the intention of calculating NCDs. Given LZMA's ability to compress the data and and adhere to most of the normality properties, it is used in the experiments throughout the remainder of this chapter.

8.2.3 Computational Complexity

Consider calculating the NCD between two optimization problems. The proposed approach requires storing both problems' length scale multisets ($\mathbf{r}_a$ and $\mathbf{r}_b$), as well as the union of the multisets ($\mathbf{r}_a \cup \mathbf{r}_b$). Assuming a total of n length scale values are sampled for each problem and p is the number of bits required to represent a single r value ($p = 64$ bits in this thesis), the total storage requirement is $4np$: np for $\mathbf{r}_a$, np for $\mathbf{r}_b$, and $2np$ for $\mathbf{r}_a \cup \mathbf{r}_b$. Hence, the total space requirements to calculate the NCD values between two problems is $O(np)$.

Calculation of n length scales for a D-dimensional problem takes $O(nD)$ time (D time is required to compute Euclidean distance between D-dimensional solutions, and this is repeated for each of the n solution pairs). Most modern compressors run in $O(n)$ time [25, 148], and so assuming $O(n)$ time to compress n length scale values, the total running time to calculate NCD is $O(nD)$.

Given a set of problems, $\mathcal{P}$, there are $\frac{|\mathcal{P}|(|\mathcal{P}|-1)}{2}$ pairs of problems, and hence to compute the NCD value between each pair of problems, a total of $\frac{|\mathcal{P}|(|\mathcal{P}|-1)}{2}$ calculations is required. Naively, each NCD value can be computed one at a time: n length scales are sampled for each problem, and the resulting length scale multisets and concatenated length scale multiset are compressed (the lengths of the compressions are then used to calculate NCD). There are clear inefficiencies in this design; the length scale sets are continually re-sampled and compressed multiple times (e.g. the multiset for a problem is sampled and compressed $|\mathcal{P}| - 1$ times.). Efficient experimental design can therefore reduce the running time of the calculation of NCD between pairs of problems in $\mathcal{P}$. Instead of generating and saving the length scale sets for each NCD calculation *one at a time*, firstly *all* length scale sets (and the combinations of their concatenations) are compressed and saved. Then, the NCD between all combinations of problems in $\mathcal{P}$ is simply calculated using a look-up table of the lengths of the compressed multisets. This approach essentially minimises the number of saves to disk and compressions of length scale multisets, however it does require all $|\mathcal{P}|$ compressed-multisets to be stored, as well as all $\frac{|\mathcal{P}|(|\mathcal{P}|-1)}{2}$ combinations of the multiset concatenations, requiring a total of $O(|\mathcal{P}|^2 np)$ space. Hence while this experimental design can improve the speed of NCD calculations between problems in a set, there is a significant trade-off in the amount of storage required to do so.

8.3 Results

The following experiments implement and execute the methodology proposed in Section 8.2 to calculate the NCD between optimization problem instances. The following experiments analyse the same problems as the length scale analysis conducted in Chapter 7, with the addition of the Griewank function. These experiments are directly focused on calculating and analysing the NCD values between problems, as well as analysing the resulting NCD values in conjunction with known problem similarities.

To begin, artificial continuous problems, namely the elliptical, Rastrigin and Griewank functions, are analysed to evaluate the ability of the NCD to reflect known structural similarities and differences between problems. A similar experiment is also conducted for the BBOB problems, and the NCD results are compared with existing knowledge and intuition regarding problem structures. Circle in a Square packing problems are also analysed in order to provide insight into real-world problem relationships. Then, instances from TSPLib are analysed in order to evaluate the efficacy of NCD applied to combinatorial problems with known similarities. In addition, two well-known combinatorial optimization problems that exhibit phase transitions, the Asymmetric Travelling Salesman Problem and Number Partitioning Problem, are also analysed to evaluate whether or not the phase transitions are detectable from the NCD values.

The NCD calculated between two given optimization problems is a scalar value representing a similarity-based feature between problems, and is therefore a similar type of feature as the J-divergence (Equation 6.1.8). Hence, the techniques used to analyse J-divergence data can also be readily applied to NCD data. Consequently, the following experimental results are summarised using three main visualisations; 1) a heatmap of the explicit NCD values, 2) a dendrogram of the clusters produced by hierarchical clustering (with unweighted average distance linkages) and 3) a two-dimensional visualisation of the problems spaced at distances reflective of their respective NCD values. Due to the symmetrical properties of NCD, heatmaps depict only the lower triangle of the NCD matrix. Problem-space visualisations are constructed using t-SNE [186]. In each experiment, an exhaustive search is conducted to determine the perplexity value (from the set $[1, 1.5, \ldots, 50]$) that minimises the average cost of 100 trials of t-SNE on the data, with a maximum of 1000 iterations.

8.3.1 Elliptical Functions

The 2-D elliptical function, defined in Table A.1, is essentially a quadratic bowl with elliptical contours, where the eccentricity of the contours is defined by a constant, $a \in \mathbb{R}$. For $a = 1$, the contours are circular, and as a increases, the contours become increasingly more elliptical. The elliptical function provides a simple and intuitive landscape from which the similarity between instances can be directly controlled via a. Therefore, this experiment investigates the relationship between problems of varying a values and the resulting NCD between them. In this experiment, $\mathcal{S} = [-1, 1]^2$ and $a \in [1, 1.25, \ldots 10]$ (i.e. a total of 37 elliptical functions are analysed). At each value of a, $2.5 \times 10^5 D = 10^6$ length scales are

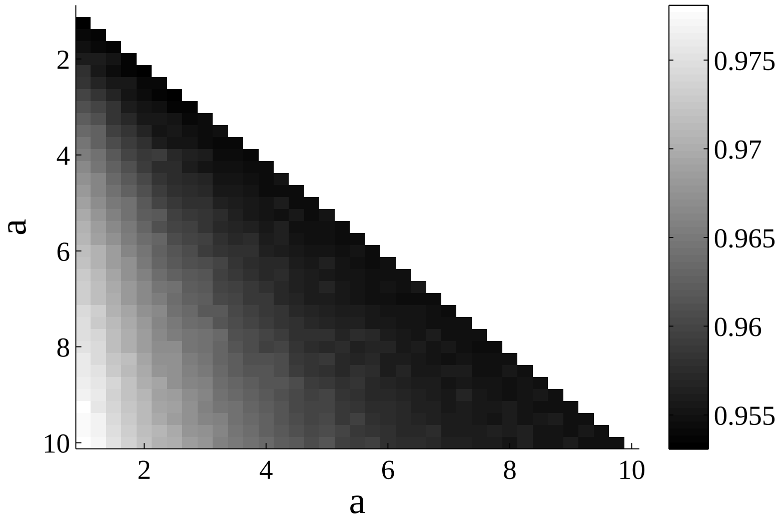


Figure 8.1: Heatmap of NCD values calculated (using LZMA) between pairs of 2-D elliptical functions, where $a = 1, 1.25, \ldots, 10$.

generated using Algorithm 6.1 with samples generated from a Lévy random walk parameterised by $\gamma = 10^{-3}$ and $\delta = 0$. LZMA with the "ultra"compression setting (a dictionary size of 64MB, BT4 matchfinder and BCJ2 filter) is used to compress the length scale multisets, and a perplexity of 19.5 (cost of 0.1451) is used in the t-SNE dimensionality reduction. Heatmaps summarising the NCD between problems as well as t-SNE visualisations of the problem space are shown in Figures 8.1 to 8.3.

The NCD values shown in the heatmap (Figure 8.1) are all very high; while NCD is defined over the interval $[0, 1]$, the values between elliptical functions are in the range $[0.9535, 0.9779]$. An *absolute* interpretation suggests that the problems are all (approximately equally) different, however, the *relative* differences between the NCD values are much more interesting and provides insight into the relationships between problems. Experimental results conducted with more length scale samples (not shown) produced a significantly larger range in NCD values, however the relative differences remained the same. This suggests that the range of NCD values can be improved by additional sampling, however this does not affect relative conclusions drawn from the results.

The light/white colouring of the lower left-hand corner of the heatmap shows that the greatest NCD is between problems of low and high eccentricities (e.g. the NCD between $a = 1$ and $a = 10$ is the largest value). In contrast, the leading diagonal in the heatmap is very dark, indicating that the most similarity between the elliptical functions occurs between problems of similar a values. The shading of the leading diagonal is very consistent, meaning that problems with similar eccentricities have similar NCD values, regardless of where

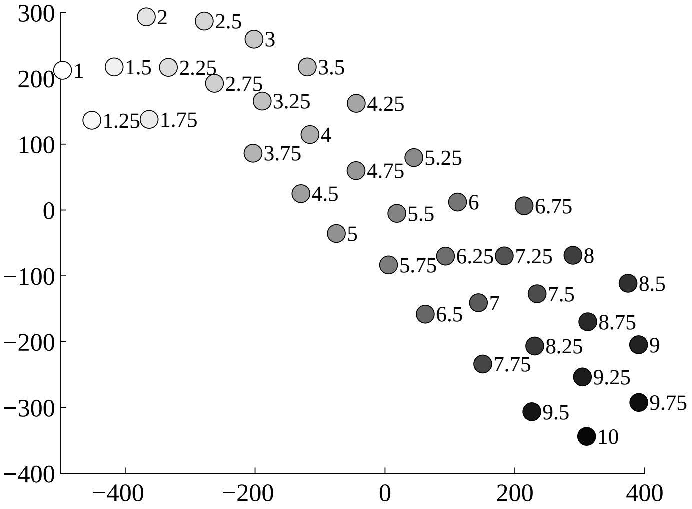


Figure 8.2: t-SNE of NCD values (cost of 0.1451) calculated (using LZMA) between pairs of 2-D elliptical functions, where $a = 1, 1.25, \ldots, 10$.

on the eccentricity spectrum they are. For example, the NCD between problems $a = 1$ and $a = 1.25$ is approximately the same as the NCD between $a = 9.75$ and $a = 10$. This is true across all of a, that is, the NCD between a_n and a_{n+1} remains constant and suggests that the structural changes caused by small increases in eccentricity are quite regular.

Figure 8.2 shows the t-SNE visualisation of the problem similarities (and hence, "problem space" according to the NCD values). The cost value of 0.1451 from t-SNE indicates that the two-dimensional visualisation is an accurate representation of the NCDs between problems. Problems in Figure 8.2 are labelled with their respective a value, and the markers are shaded from white to black as a transitions from 1 to 10. Figure 8.2 captures the general trend that problems with similar eccentricities are similar to each other; the problems are spatially ordered according approximately to a. Furthermore, the positioning of the problems ensures that the largest spatial distance is between low eccentricities and high eccentricities.

Viewing the dendrogram in Figure 8.3 from the bottom, it is clear that problems with similar a values are more similar than problems with very different a values. By examining the dendrogram from the top, it is clear that the problems form two major clusters; $1 \leq a \leq 3$ and $3.25 \leq a \leq 10$. The large NCD connecting these clusters indicates that they are quite well-separated. Moving downwards, the problems can be further clustered into three clusters by using a NCD threshold of approximately 0.958: $1 \leq a \leq 3$, $3.25 \leq a \leq 7$, and $7.25 \leq a \leq 10$. Moving downwards again, the three clusters disintegrate into numerous sub-clusters, suggesting that for the elliptical functions considered, there is at most 3 clear clusters according to hierarchical clustering.

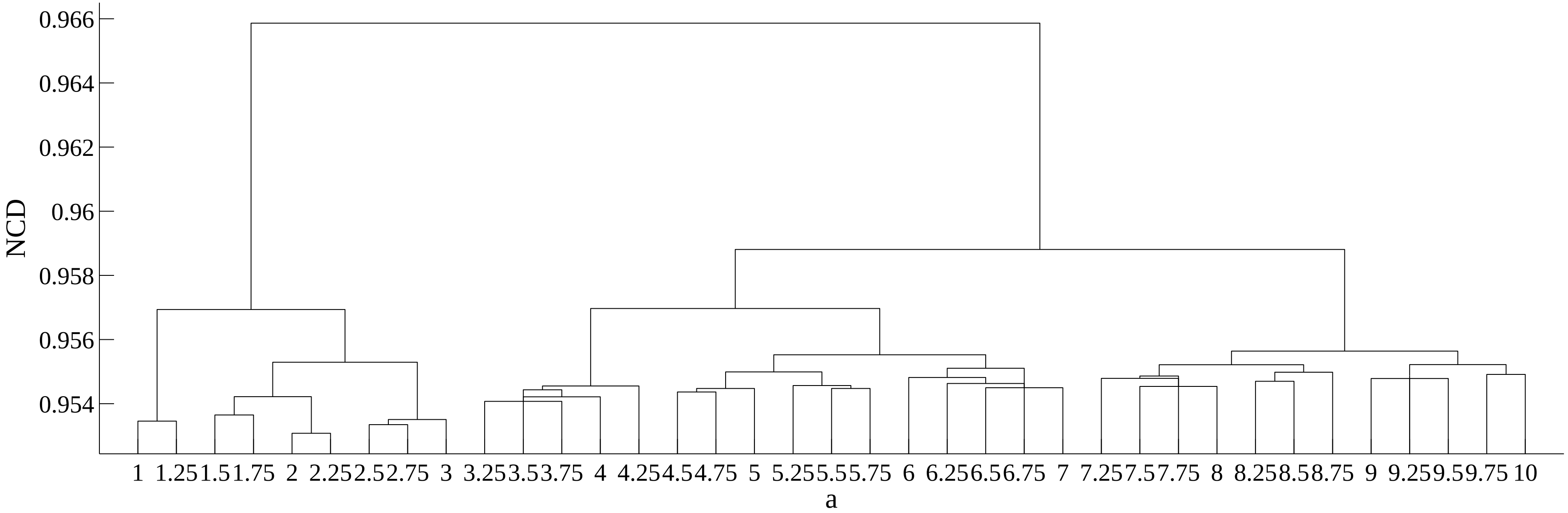


Figure 8.3: Dendrogram of NCD values calculated (using LZMA) between pairs of 2-D elliptical functions, where $a = 1, 1.25, \ldots, 10$.

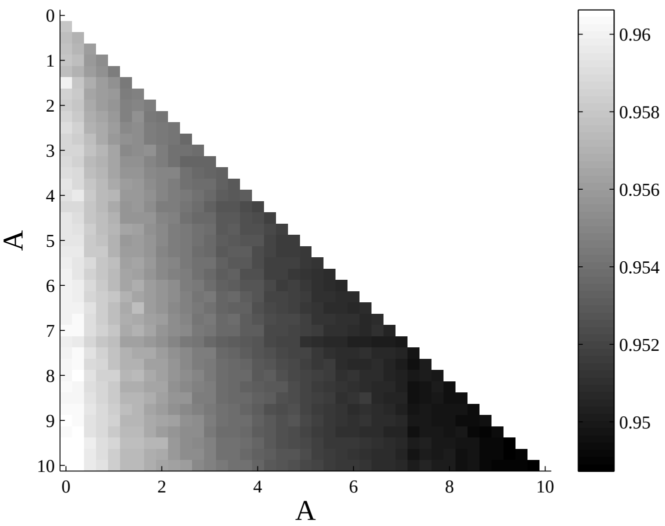


Figure 8.4: Heatmap of NCD values calculated (using LZMA) between pairs of 1-D Rastrigin functions, where $A = 0, 0.25, \ldots, 10$.

8.3.2 Rastrigin

The Rastrigin function (see Appendix A.1.1) also provides a simple, intuitive landscape that can be changed using one parameter; the perturbation term A. The goal of this experiment is to investigate how the change in perturbation affects the similarity between the problems. Results in Section 7.1.2 showed that problems with similar perturbation values were more similar than problems with very different levels of perturbation. Similar to the experiments in Section 7.1.2, the dimensionality is fixed at $D = 1$, and $A = 0, 0.25, \ldots, 10$. A total of 2.5×10^5 solutions are sampled from $\mathcal{S} = [-5.12, 5.12]$ using a Lévy random walk parameterised by $\gamma = 0.005$ and $\delta = 0$, and length scales are calculated using Algorithm 6.1. Figures 8.4 to 8.6 show the resulting heatmap, t-SNE (perplexity of 32) problem space visualisation and dendrogram of the NCD values between problems.

The heatmap in Figure 8.4 shows that the explicit NCD values are quite high across the set and are arguably quite similar. Despite their similarities, clear trends are evident. As expected, the largest distance between problems is generally between the smoothest and most rugged problems: $NCD(A = 0, A = 10) = 0.9606$. Looking along the diagonal of Figure 8.4, as A increases, the shading becomes darker. This indicates that problems with small A values have a slightly larger distance between them than problems at large A values. The dendrogram in Figure 8.6 also confirms this; as A increases, the NCD between problems generally decreases. t-SNE's final cost value of 0.3205 suggests that the visualisation in Figure 8.5 is not a very accurate summary of the problem space, however it generally reflects that the largest differences occur between low and high values of A.

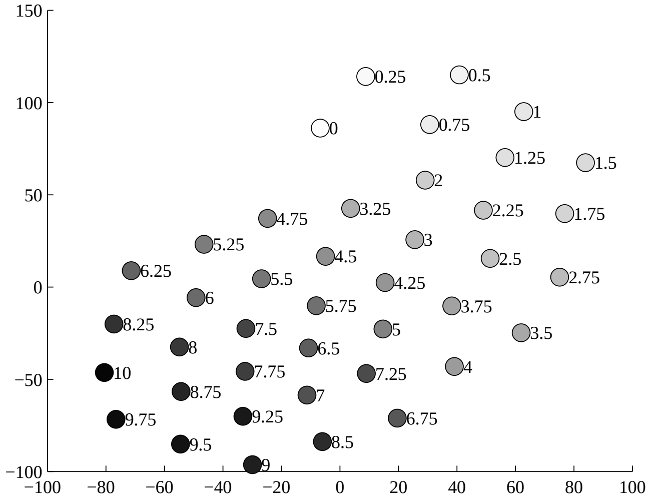


Figure 8.5: t-SNE of NCD values (cost of 0.3205) calculated (using LZMA) between pairs of 1-D Rastrigin functions, where $A = 0, 0.25, \ldots, 10$.

8.3.3 Griewank

The structure of the Griewank problem (Appendix A.1.1) becomes increasingly more convex as dimensionality increases [100], and so this experiment investigates whether the NCD values can reflect changes in the landscape structure that occur as dimensionality increases, particularly, the increasing similarity to a convex function. Specifically, the NCD between Griewank functions from 1 to 15 dimensions is calculated. To evaluate whether the NCD can detect an increasing resemblance to a convex function as dimensionality increases, the NCD between Griewank functions and the convex component, $1 + \frac{1}{4000}\sum_{i=1}^{D} x_i^2$, is also calculated. $2.5 \times 10^5 D^2$ solutions are sampled from $\mathcal{S} = [-600, 600]^D$ using a Lévy random walk parameterised by $\gamma = 0.5$ and $\delta = 0$. Heatmaps summarising the NCD results between problems, a t-SNE (with perplexity of 19.5) visualisation of the problem space and a dendrogram of the clustering of problems are shown in Figures 8.8 to 8.9.

t-SNE obtained a cost of 0.0497, indicating that the discrepancy of distances between points in the original data and reduced data is very low and that the visualisation is a reliable summary of the problem space. The t-SNE visualisation in Figure 8.7 nicely illustrates the behaviour of the Griewank problem; Griewank is always very similar to its convex component, and problems of high dimensions (e.g. $D = 14$ and 15) are more similar than problems of lower dimensions (e.g. $D = 1$ and 2). The heatmap and dendrogram shown in Figures 8.8 and 8.9 provides greater insight into Griewank's behaviour ("G" denotes Griewank, while "C" denotes the convex component). The heatmap shows that the largest dissimilarity to $D = 1$ Griewank is $D = 15$ Griewank. Using the dendrogram, clustering Griewank

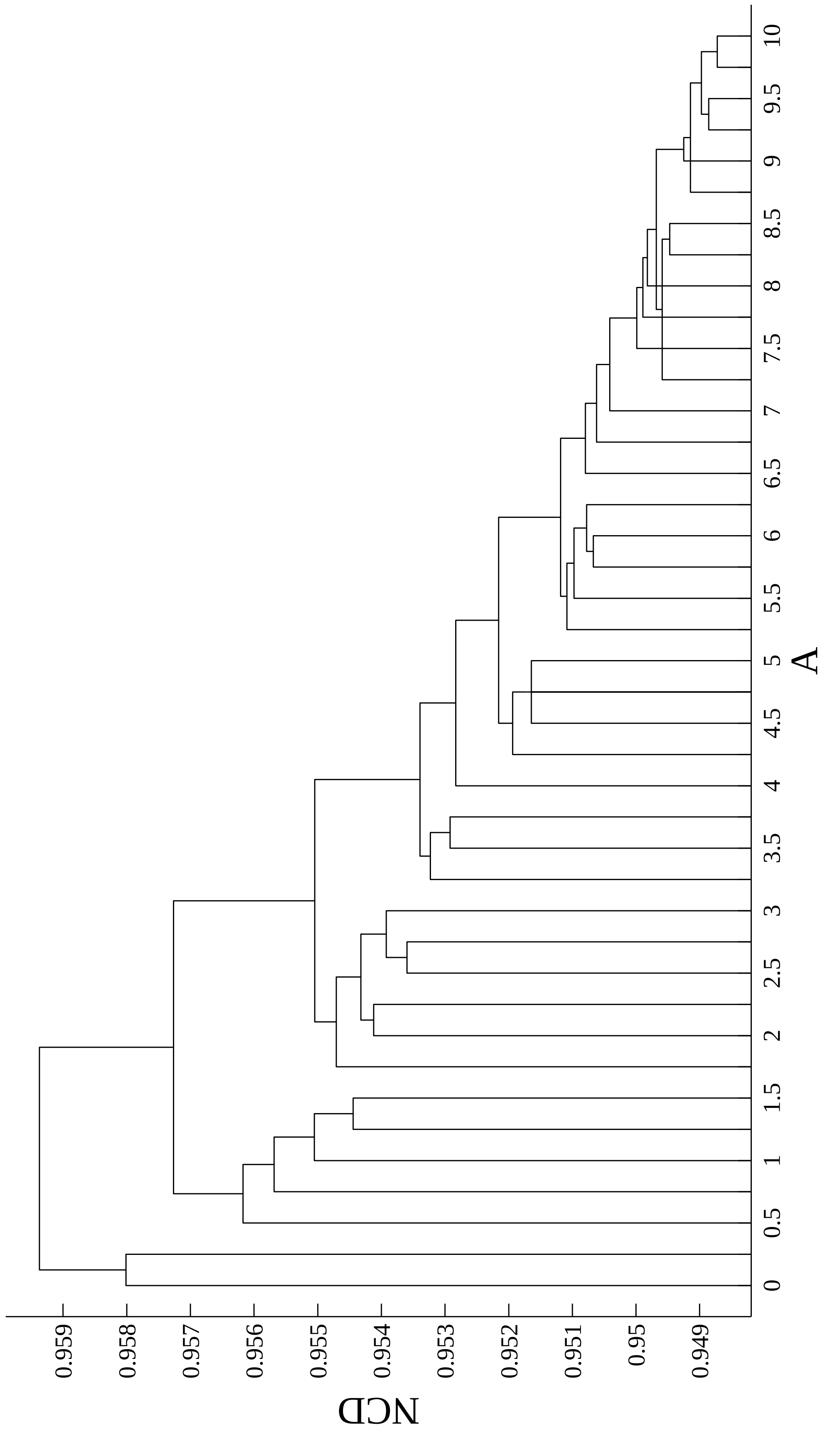


Figure 8.6: Dendrogram of NCD values calculated (using LZMA) between pairs of 1-D Rastrigin functions, where $A = 0, 0.25, \ldots, 10$.

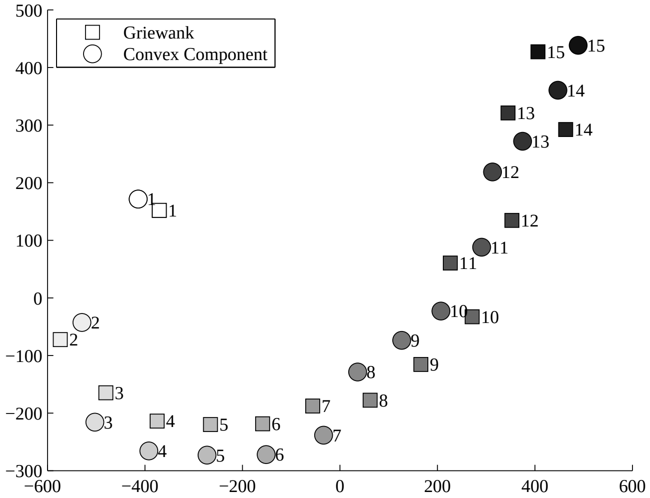


Figure 8.7: t-SNE of NCD values (cost of 0.0497) calculated (using LZMA) between pairs of Griewank instances, where $D = 1, 2, \ldots, 15$. "G"denotes Griewank functions, while ′C"denotes the convex component.

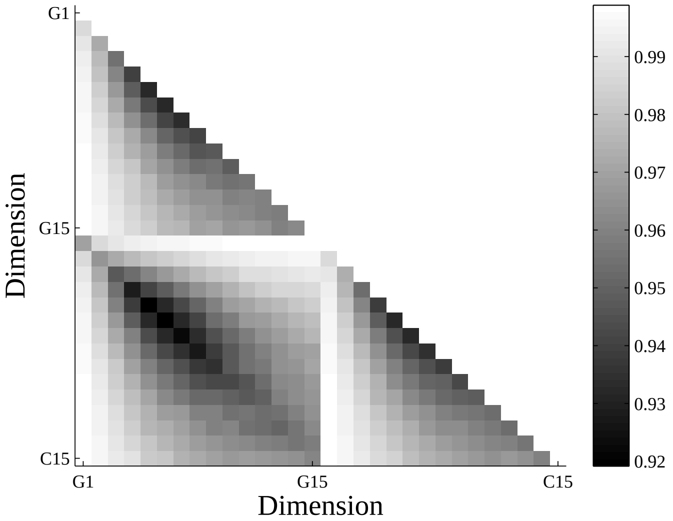


Figure 8.8: Heatmap of NCD values calculated (using LZMA) between pairs of Griewank instances, where $D = 1, 2, \ldots, 15$. "G"denotes Griewank functions, while ′C"denotes the convex component.

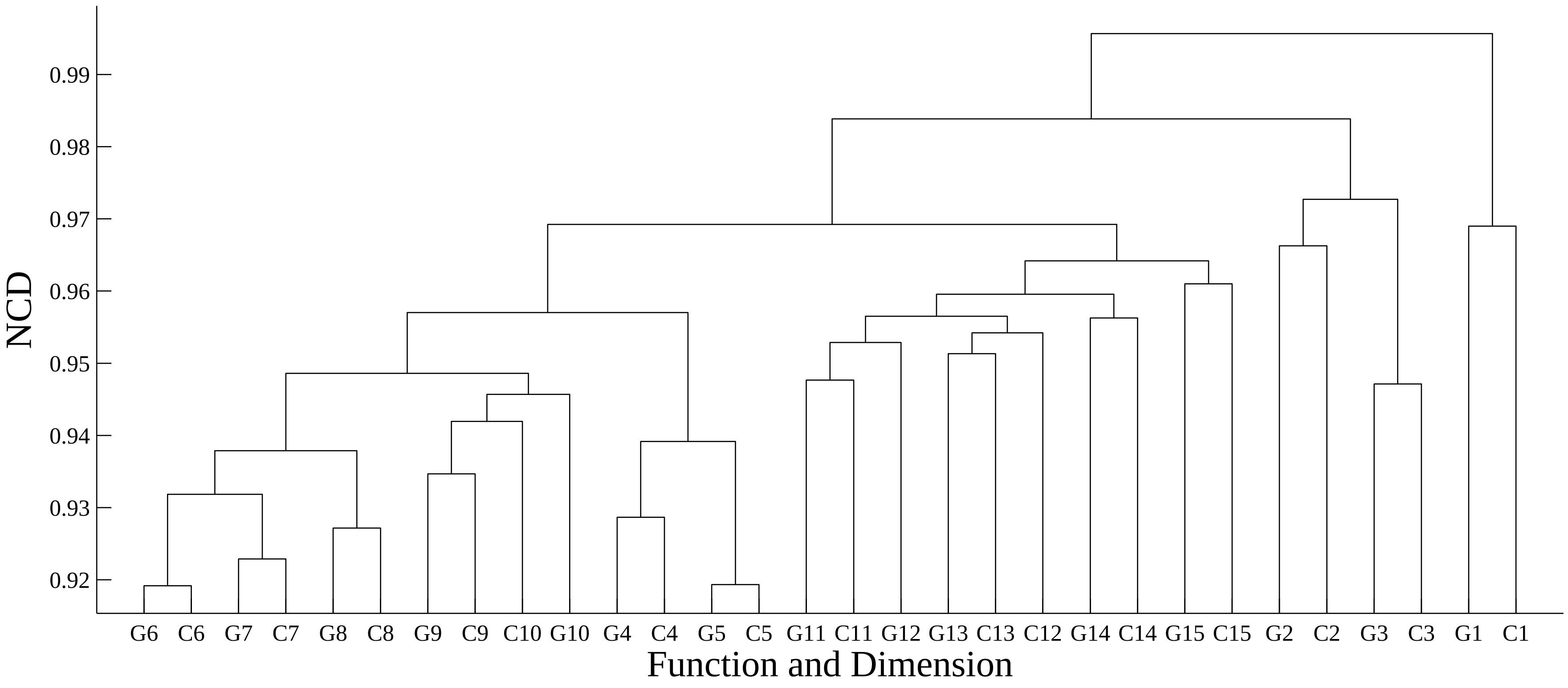


Figure 8.9: Dendrogram of NCD values calculated (using LZMA) between pairs of Griewank instances, where $D = 1, 2, \ldots, 15$. "G"denotes Griewank functions, while 'C"denotes the convex component.

into two clusters results in $\{D = 1\}$, and $\{D > 1\}$, while clustering into three clusters yields $\{D = 1\}$, $\{D = 2, D = 3\}$ and $\{D > 3\}$. This pattern generally continues and illustrates that as D increases, the Griewank problems become more similar. The heatmap and dendrogram also confirm the expectation that lower dimensional problems (e.g. $D = 1$) are less convex than higher dimensional problems. However, it appears that as D increases, the similarity between Griewank and its convex component peaks at approximately $D = 6$, and for $D > 6$, Griewank and its convex component become increasingly more dissimilar.

8.3.4 Black-Box Optimization Benchmarking Problems

One of the major motivations behind the development of the BBOB problem set is to provide a set of problems that reflect a diverse range of structural features that are thought to be challenging for algorithms to solve in practice [72]. Hence, the BBOB problems have been purposefully constructed using predefined notions of challenging structures. As previously argued in Section 7.2, there is little doubt that the BBOB problem set contains a variety of challenging structures, however the amount of variety has not been explicitly quantified. The BBOB problems are therefore an interesting problem set over which to calculate NCD values. The aim of this experiment is to estimate the NCD values between BBOB instances and assess the degree to which the known and conjectured similarities and differences in the BBOB problems are reflected by the resulting NCD values.

These experiments are based on the experimental procedure in Section 7.2. Specifically, a Lévy random walk (parameterised by $\gamma = 10^{-3}$ and $\delta = 0$) of $50000D$ steps is used to sample $\mathcal{S} = [-5, 5]^D$, and length scales are calculating using Algorithm 6.1. In these experiments, 2, 5, 10 and 20-D problems are analysed, and 30 instances are generated for each problem by supplying seeds 1 to 30 to the BBOB generator. NCD is calculated on the (sorted) length scale values, and LZMA used to compress the values. The results report the mean NCD value between problem pairs (averaged over the 30 instances). Heatmaps summarising the mean NCD between problems, t-SNE visualisations of the problem space and dendrograms of the clustering of problems are shown in Figures 8.10 to 8.21.

The NCD values between BBOB problems displayed in the heatmaps (Figures 8.10 to 8.13) are generally quite high (ranging between 0.93 and 1), and so *relative* comparisons between values, indicated by the shading of heatmap cells, gives better insight into the relationships between problems. For example, F8 and F9 are shaded black in all heatmaps, indicating that there is a stronger similarity between them than any other prob-

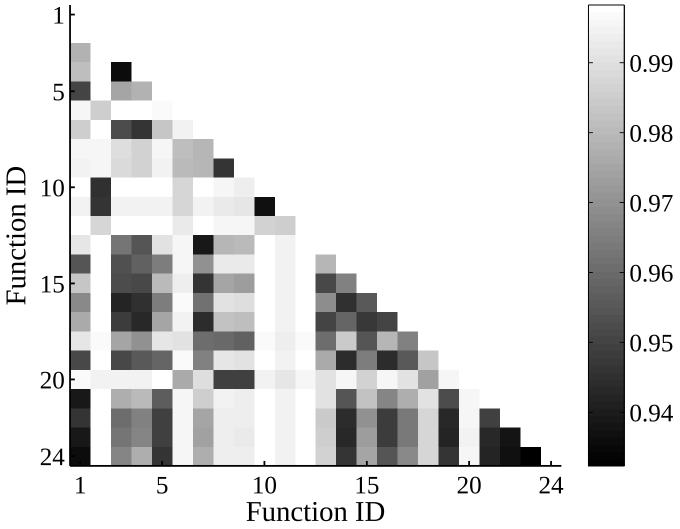


Figure 8.10: Heatmap of NCD values calculated (using LZMA) between pairs of 2-D BBOB instances.

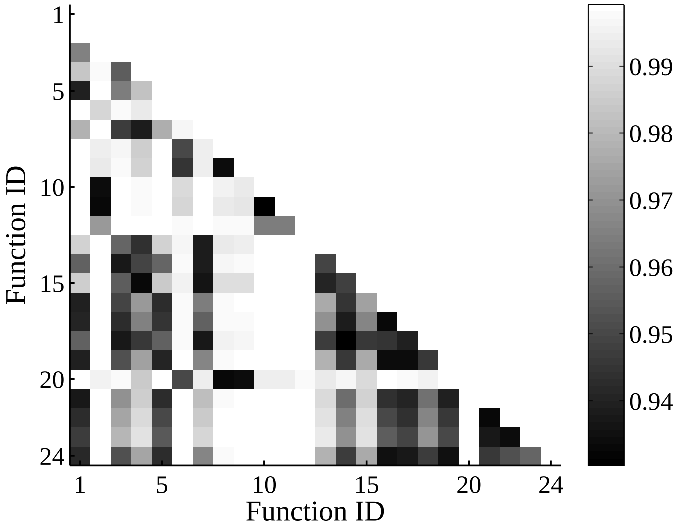


Figure 8.11: Heatmap of NCD values calculated (using LZMA) between pairs of 5-D BBOB instances.

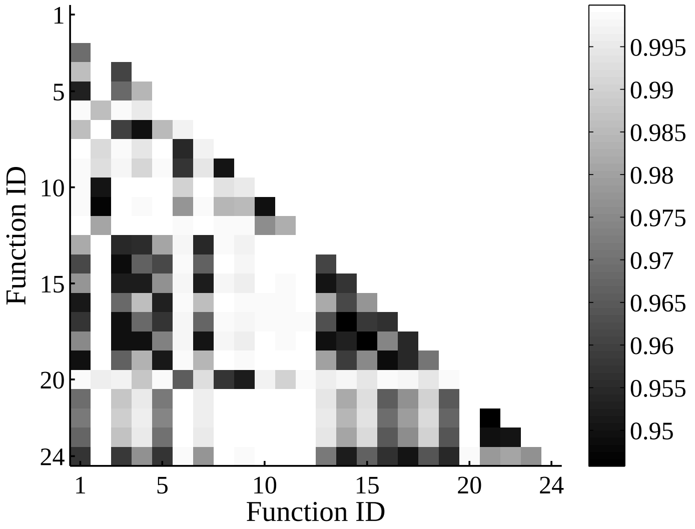


Figure 8.12: Heatmap of NCD values calculated (using LZMA) between pairs of 10-D BBOB instances.

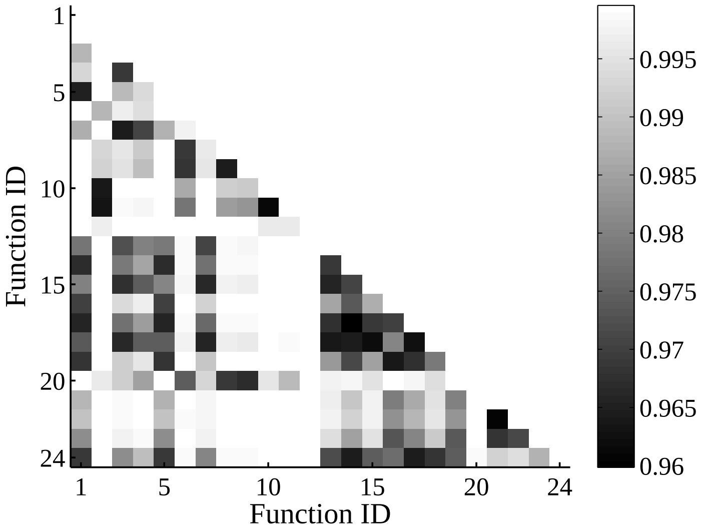


Figure 8.13: Heatmap of NCD values calculated (using LZMA) between pairs of 20-D BBOB instances.

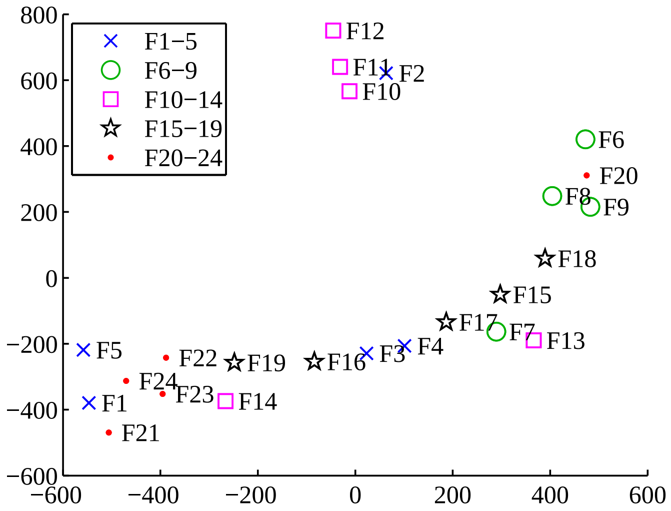


Figure 8.14: t-SNE of NCD values (cost of 0.0611) calculated (using LZMA) between pairs of 2-D BBOB instances.

lem pairing. F8 and F9 are Rosenbrock functions differing by a rotation of the search space, and hence they are identical from an information/structural perspective. The heatmaps also display many problems for which there is a high dissimilarity (indicated by light/white shading). For example, F12 (Bent Cigar function) is highly dissimilar to all of the other BBOB problems, even across D.

The shading patterns, and hence relative similarities, reflected by the heatmaps are generally quite consistent across D, suggesting that the landscape structures scale with D, and that this is captured by the NCD analysis. There are, however, a few NCD values that are exceptions to this general trend; for example, F10 increases its similarity with respect to both F8 and F9 as D increases, while F3 decreases its similarity with respect to F21, F22 and F23. The change in similarity is likely due to subtle changes in landscape structure as D increases. For example, F8 and F9 have a ridge that changes its orientation $D - 1$ times, and so this may affect the similarity between F8 (or F9) and problems that remain structurally consistent across D.

Figures 8.14 to 8.17 show the two-dimensional representations of the NCD values, determined by t-SNE. Each problem is marked according to its respective problem classification defined by the BBOB developers. The problems exhibit no clear clustering according to their respective BBOB classes. However, there are clearly strong relationships between the problems, and these relationships persist as D increases. For example, problems within the subsets $\{F1, F5, F19\}$, $\{F2, F10, F11, F12\}$, $\{F3, F4, F15\}$, $\{F6, F8, F9\}$ and $\{F21, F22, F23\}$ are positioned in close proximity to each other. Many of the problems in the subsets are known

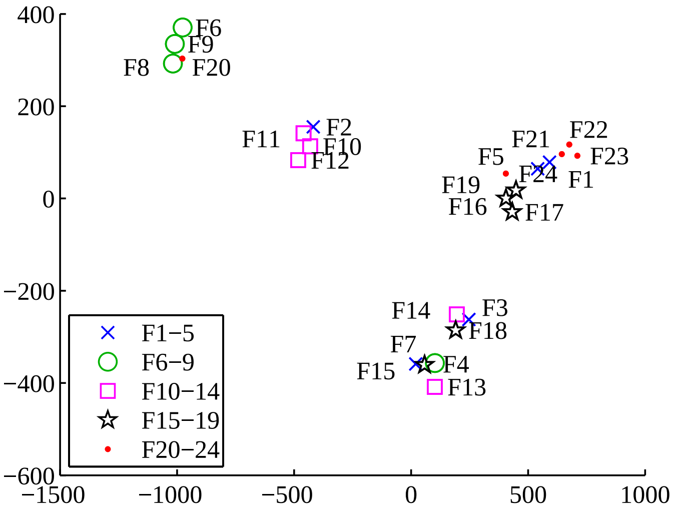


Figure 8.15: t-SNE of NCD values (cost of 0.0534) calculated (using LZMA) between pairs of 5-D BBOB instances.

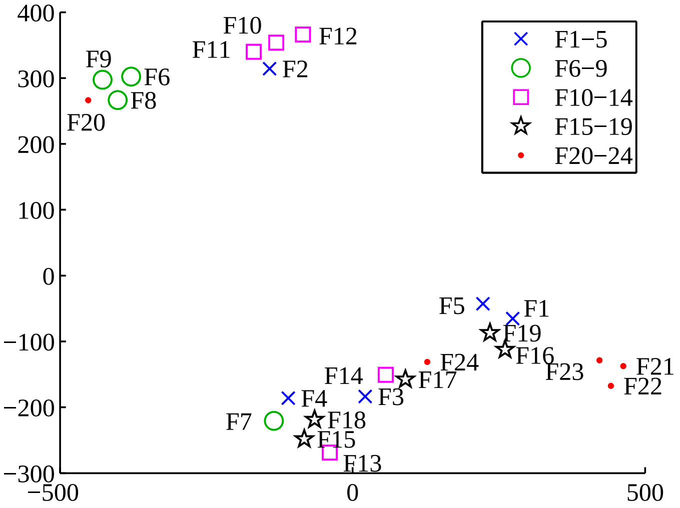


Figure 8.16: t-SNE of NCD values (cost of 0.0443) calculated (using LZMA) between pairs of 10-D BBOB instances.

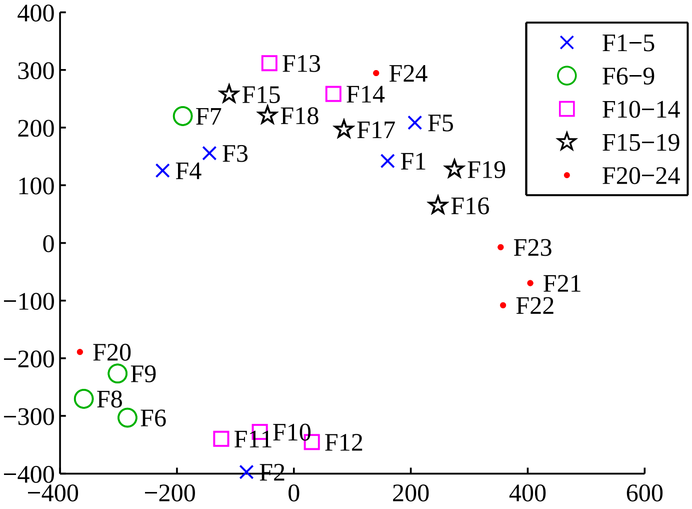


Figure 8.17: t-SNE of NCD values (cost of 0.0370) calculated (using LZMA) between pairs of 20-D BBOB instances.

to have similar landscape structures: F2, F10 and F11 are ellipsoidal functions, F3, F4 and F15 are variants of the Rastrigin function, F8 and F9 are (rotationally different) Rosenbrock functions, and F21 and F22 contain Gaussian-shaped modes. Overall, the t-SNE visualisations suggests that there are a variety of landscape structures in the BBOB problem set, and that the NCD methodology is capable of capturing known landscape similarities and differences.

Dendrograms of the hierarchical clustering of NCD values (for each D) are shown in Figures 8.18 to 8.21. The dendrograms exhibit similar groupings across D, and are quite well-balanced. By analysing the dendrograms from the bottom, only a few problems have a relatively small NCD value between them (specifically, F10 and F11, F14 and F17, F15 and F18, and F21 and F22). Moving further up, it is clear that problems with known similarities form clusters. For example, F8 and F9 have a relatively low NCD compared to other problems. Top-down analysis of the dendrograms suggests that there are generally three main clusters; $\{F2, F10, F11, F12\}$, $\{F6, F8, F9, F20\}$ and the remaining problems. Interestingly, clustering the problems into 5 clusters does not yield the 5 pre-defined BBOB classes (indeed, F12 forms its own cluster entirely).

Due to the generality and simplicity of the NCD methodology, comparisons between problems differing in their type and/or dimensions can easily be made. To further investigate the landscape changes as D increases, the NCD values between BBOB problems in *different* dimensions is calculated, and the resulting heatmap and t-SNE visualisation is shown in Figures 8.22 and 8.23 respectively.

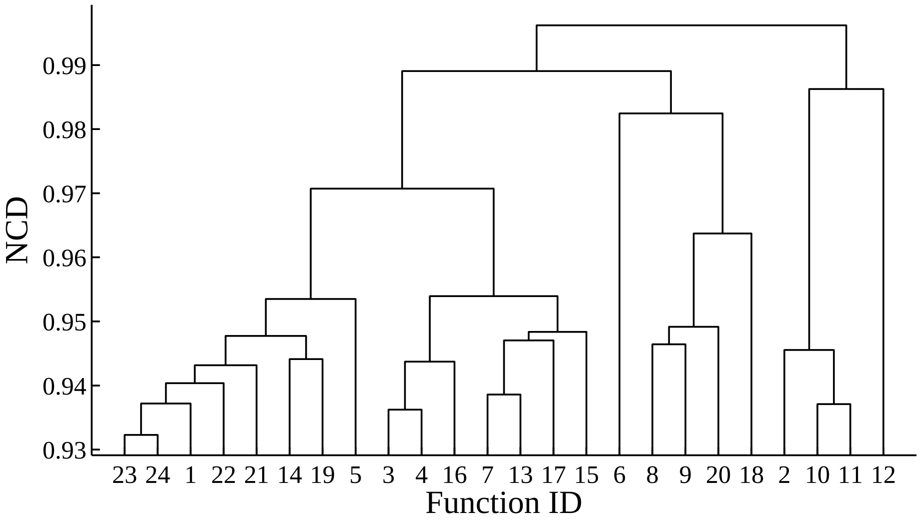


Figure 8.18: Dendrogram of NCD values calculated (using LZMA) between pairs of 2-D BBOB instances.

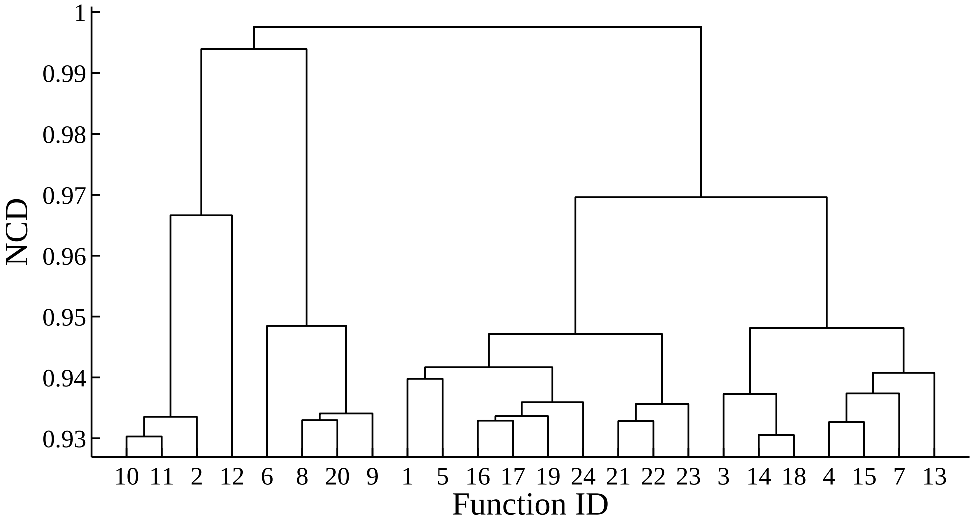


Figure 8.19: Dendrogram of NCD values calculated (using LZMA) between pairs of 5-D BBOB instances.

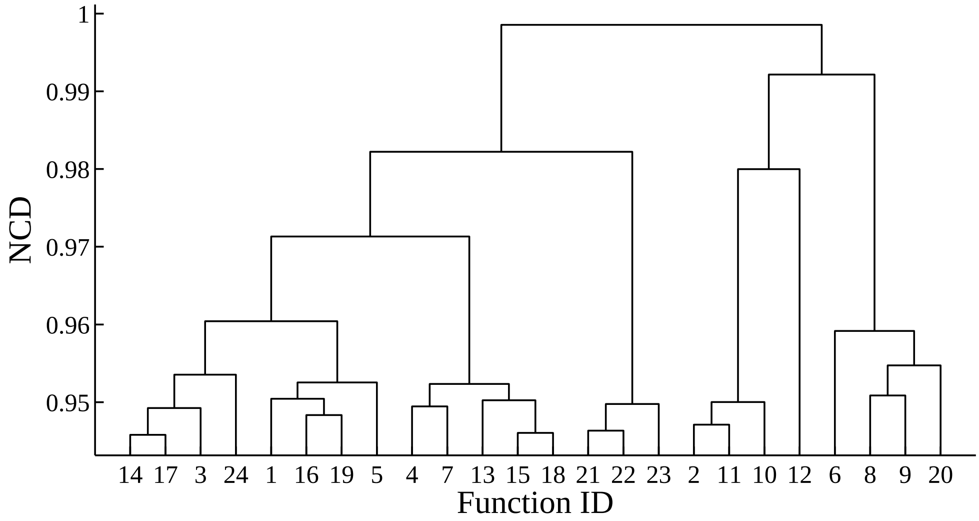


Figure 8.20: Dendrogram of NCD values calculated (using LZMA) between pairs of 10-D BBOB instances.

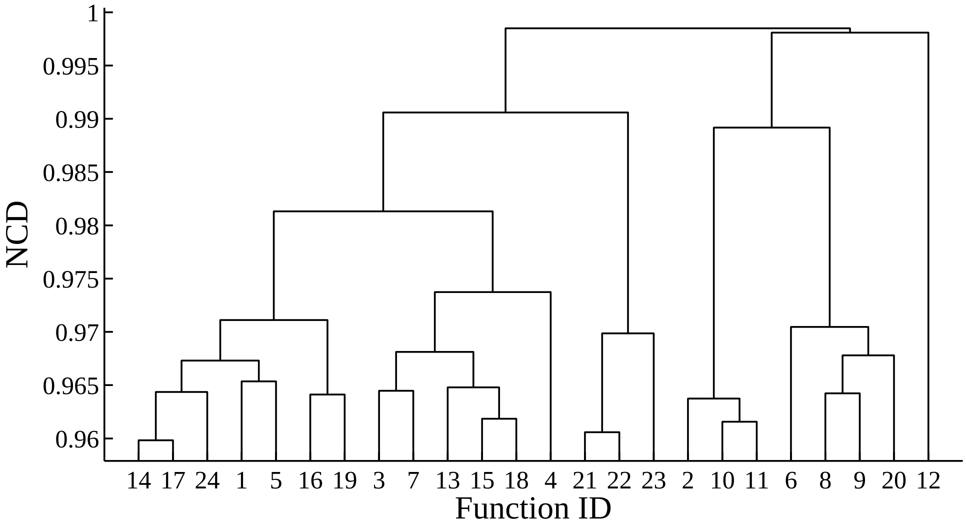


Figure 8.21: Dendrogram of NCD values calculated (using LZMA) between pairs of 20-D BBOB instances.

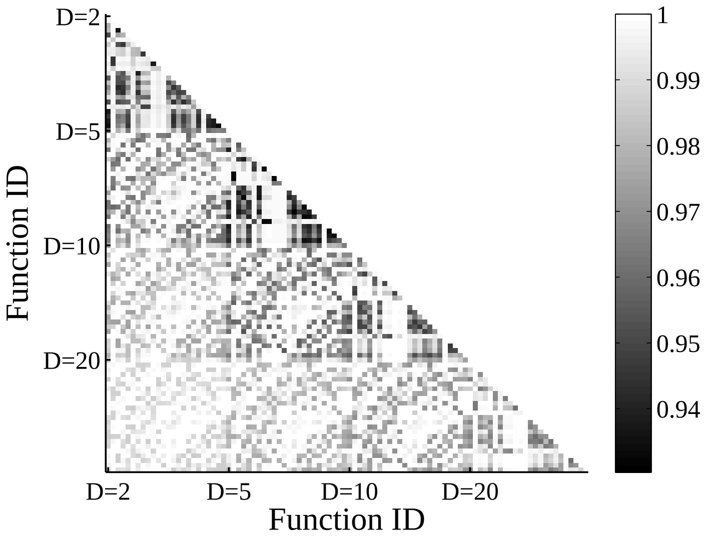


Figure 8.22: Heatmap of NCD values calculated (using LZMA) between pairs of 2, 5, 10 and 20-D BBOB instances.

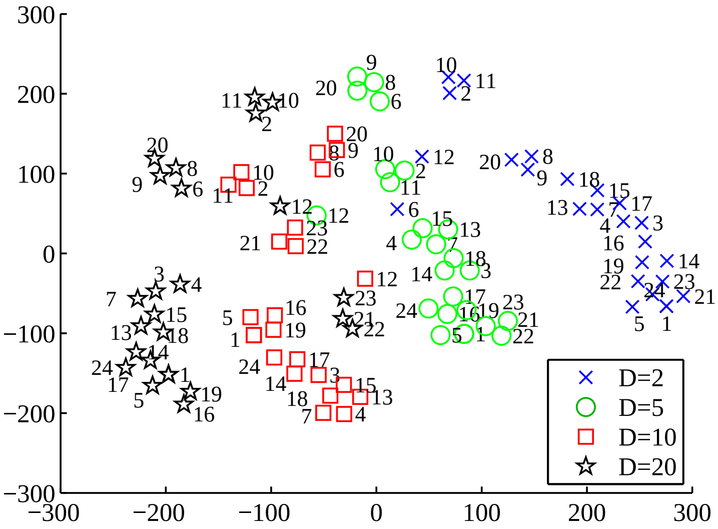


Figure 8.23: t-SNE of NCD values (cost of 0.4136) calculated (using LZMA) between pairs of 2, 5, 10 and 20-D BBOB instances.

The heatmap (Figure 8.22) nicely illustrates the consistency of the shading patterns across D, shown by the repetitive pattern near the leading diagonal. More importantly, it also shows that there is a greater dissimilarity between problems of widely varying D, than problems of similar D. For example, the region pertaining to 2 and 20-D problems is much lighter than the region associated with 2 and 5-D problems. One possible explanation is that similar problems within a given dimension share a common structure (e.g. a ridge), however because structures can change significantly with D, they bear little resemblance between D, and so comparisons of a particular structure over D will yield dissimilar values.

The t-SNE visualisation (Figure 8.23) also reflects the large dissimilarity between largely varying D; the problems are positioned horizontally in the space in the order of their dimensionality. Furthermore, the problems appear to be largely clustered according to there respective D. There are notably a few problems that do not adhere to this strict clustering; F12 is positioned between the clusters, and is rather invariant to D. The 20-D F21, F22 and F23 problems are also separated from the main 20-D cluster, and instead are located between the 5-D and 10-D clusters. Clearly, the NCD values capture the changes in landscape structure over D, and analysis of the NCD values shows that some problems are more affected by D than others.

In summary, the analysis of NCD values has provided some valuable insights into the relationships between the BBOB problems. Specifically, the relative relationships between problems tends to remain the same as D increases, suggesting that the structural information within the BBOB problems scales with D. However, as shown in Figures 8.22 and 8.23, the problems increasingly differ *between dimensions* as D increases, meaning that a function in 2-D is unlikely to bear similarity to the same function in 20-D. The NCD analysis conducted in this section also shows that the BBOB problem set contains a variety of structural features, however there are also a few strikingly similar problems, namely, F2/F10, F8/F9 and F21/F22. Given the duplication of information within these problems, F10, F9 and F22 could potentially be removed without affecting the integrity of the benchmark set.

8.3.5 Circle in a Square Problems

The results of the Circle in a Square (CiaS) packing problems in Section 7.3 suggest that problems with similar numbers of circles, n_c, are more similar than problems at highly different values of n_c, and that as n_c increases, the problems become increasingly more similar. This experiment investigates the relationship between CiaS problems of varying values of n_c, by

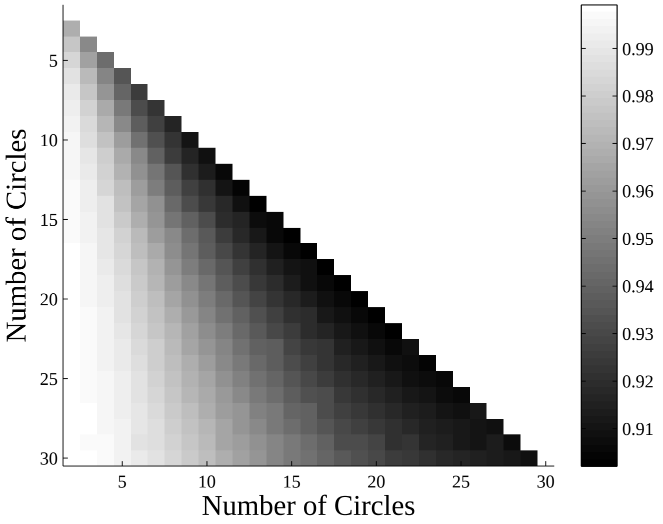


Figure 8.24: Heatmap of NCD values calculated (using LZMA) between pairs of CiaS instances, where $n_c = 2, 3, \ldots, 30$.

calculating and analysing the NCD between pairs of CiaS problems. The CiaS analysis in Section 7.3 indicates that problems where $n_c > 30$ are highly similar, and so this experiment analyse problems where $n_c = 2, 3, \ldots, 30$. A total of $5 \times 10^5 n_c$ solutions were sampled from $\mathcal{S} = [0, 1]^{2n_c}$ using a Lévy random walk parameterised by $\gamma = 5 \times 10^{-4}$ and $\delta = 0$. Length scales are calculated using Algorithm 6.1. A heatmap summarising the NCD between problems, t-SNE (with perplexity of 4) visualisations of the problem space and a dendrogram of the clustering of problems are shown in Figures 8.24 to 8.26.

The heatmap of NCD values in Figure 8.24 agrees with the results in Section 7.3; as n_c increases, the problems become increasingly more similar. In addition, the dark colouring along the diagonal and white colouring on the lower left corner shows that problems at similar values (e.g. $n_c = 10$ and $n_c = 11$) of n_c are similar, particularly in comparison to problems at largely differing values of n_c (e.g. $n_c = 2$ and $n_c = 30$). The cost of 0.0885 from t-SNE indicates that the discrepancy of distances between points in the original data and reduced data is very low and that the visualisation of the problem space in Figure 8.25 is a good summary of the problem space. The sharp decrease in NCD values between problems in the dendrogram (Figure 8.26) also indicates that the problems becoming increasingly more similar as n_c increases.

8.3.6 TSPLib

TSPLib contains Travelling Salesman Problems gathered from a variety of sources and applications, such as printed circuit board routing and geographical city layouts. In addition

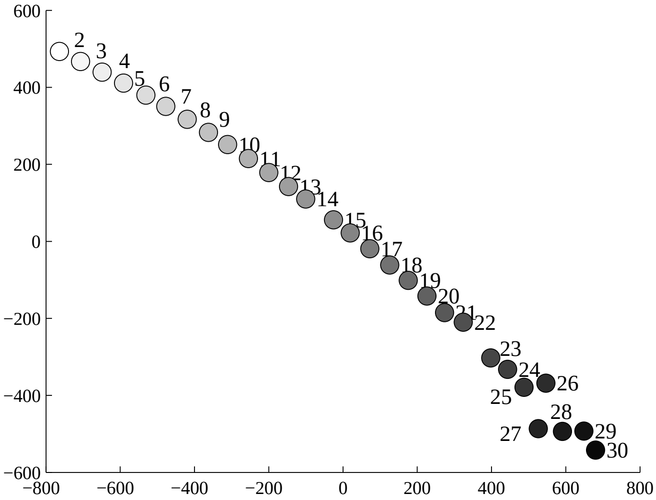


Figure 8.25: t-SNE of NCD values (cost of 0.0885) calculated (using LZMA) between pairs of CiaS instances, where $n_c = 2, 3, \ldots, 30$.

to the underlying source problem, the problems differ mainly in the number of cities, type of distance between cities and symmetry of their distance matrix (symmetrical vs asymmetric). The aim of this experiment is to calculate the NCDs between TSPLib instances and evaluate whether the NCD values correlate with similarities known regarding the source, size, distance type and symmetry. The same subset of TSPLib instances examined in Section 7.4 (and summarised in Table A.3) is used in this experiment. Similar to previous experiments, $2.5 \times 10^5 \times n$ random solutions (i.e. tours) are generated for each problem, and length scale values are calculated using Algorithm 6.1. Figures 8.27 to 8.29 show the heatmap, t-SNE visualisation (perplexity of 10) and dendrogram of the TSPLib instances.

The NCD values between TSPLib instances ranged between 0.1123 and 1, which is a much wider range than the NCD values resulting from the continuous optimization problems. The heatmap in Figure 8.27 shows that the "kro"-type instances are all very similar to each other, and are very different to the remaining benchmark problems. The NCD values show that the asymmetric kro124p instance is less similar to the other "kro"-type problems, suggesting that the symmetry is captured by the NCD values. The bayg29 and bays29 instances are also very similar to each other. Problems from similar sources are generally more similar than problems of differing sources. For example, ulyssess16 and ulyssess22, as well as the "ftv"-type problems. Out of all of the TSPLib instances examined, the heatmap indicates that p43 is a highly unique problem, which agrees with the analysis in Section 7.4.

The t-SNE visualisation in Figure 8.28 (cost of 0.1026) accurately represents the NCDs between instances. There are no trends related to the size of the instances. Remarkably,

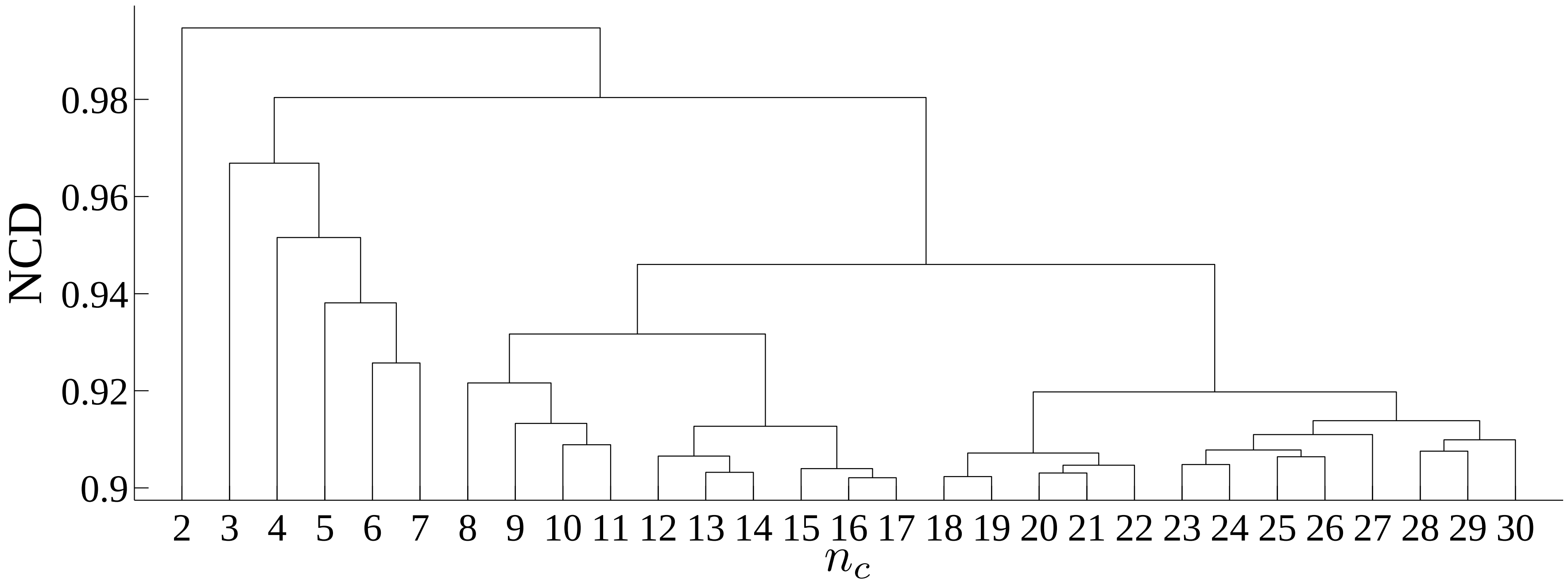


Figure 8.26: Dendrogram of NCD values calculated (using LZMA) between pairs of CiaS instances, where $n_c = 2, 3, \ldots, 30$.

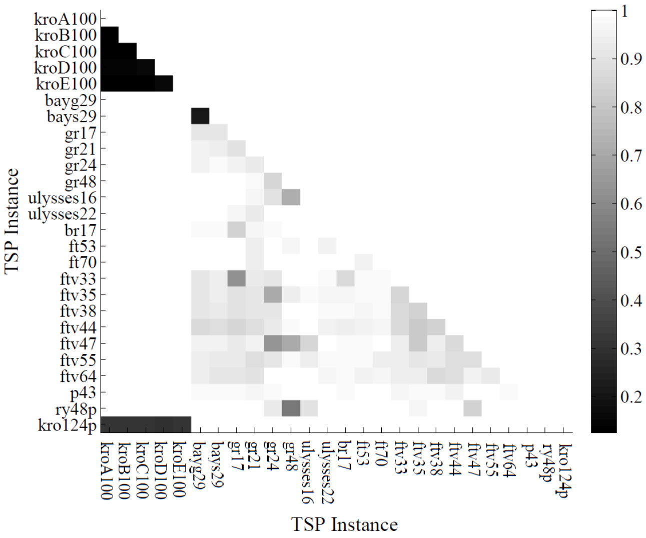


Figure 8.27: Heatmap of NCD values calculated (using LZMA) between pairs of TSPLib instances.

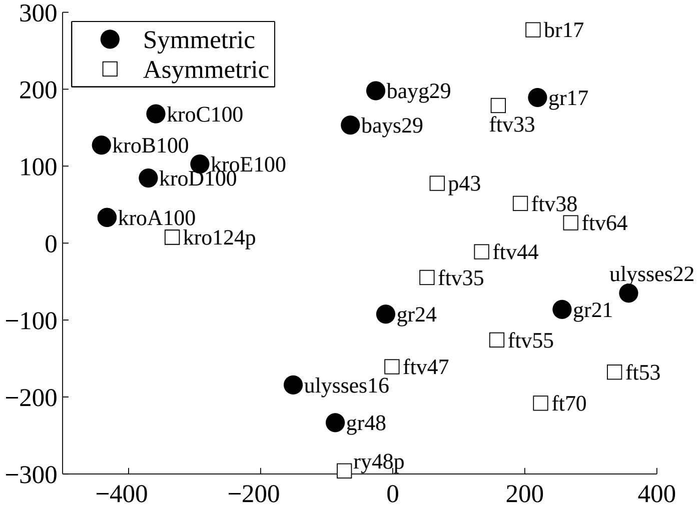


Figure 8.28: t-SNE of NCD values (cost of 0.1026) calculated (using LZMA) between pairs of TSPLib instances.

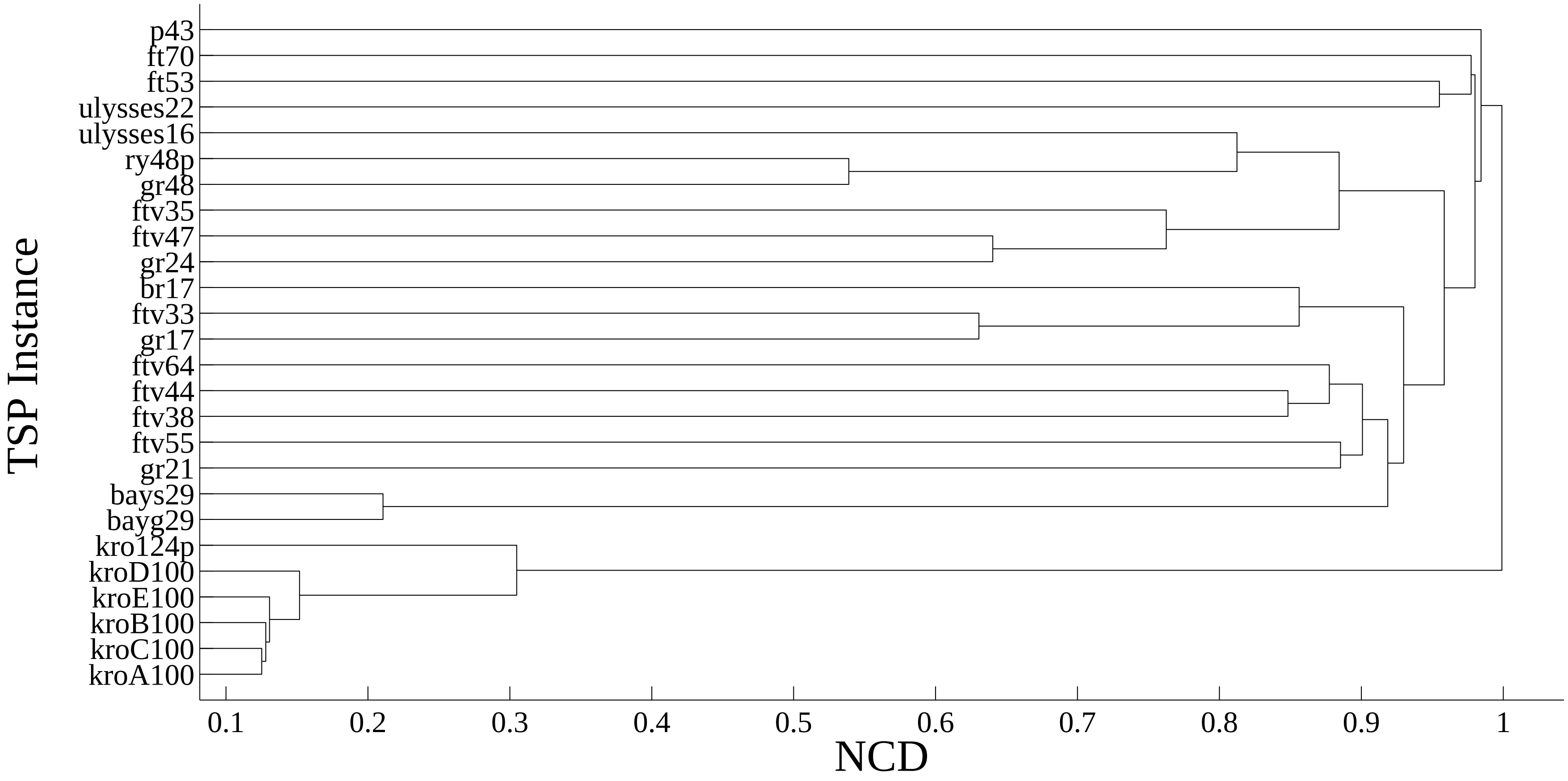


Figure 8.29: Dendrogram of NCD values calculated (using LZMA) between pairs of of TSPLib instances.

the problems are quite well clustered according to their source and symmetry. The asymmetric problems are generally located in the bottom-right of the space, while the symmetric are quite dispersed. The "kro"-type problems form a clear cluster, as do the "bay"-type problems. This clustering is also exhibited in the dendrogram in Figure 8.29. Overall, the dendrogram shows that apart from the 2 clusters formed by the "kro"-type and "bay"-type problems, the problems are all quite different to each other, which is a desirable property of benchmarking problem sets.

8.3.7 Asymmetric Travelling Salesman Problems

Instances of Asymmetric Travelling Salesman Problems (ATSPs) randomly generated with distances sampled uniformly from integers in the range $[0, \ldots, \lfloor 10^b \rfloor]$ (where $b > 0$) have been shown to exhibit a phase transition in the performance of exact solvers, as well as specific TSP-related landscape properties such as the size of backbones [206]. The aim of this experiment is to analyse the NCDs between ATSP instances generated throughout the phase transition, and to evaluate whether the phase transition behaviour is reflected in the estimated NCD values between instances. The experimental setup is identical to the experiments in Section 7.5; 10-city problems are randomly generated using $b = 0, 0.1, \ldots 6.5$ (thus a total of 66 problems)[1]. For each value of b, ten TSP instances are generated using ten separate distance matrices, where entries in each matrix (excluding the leading diagonal) are sampled uniform randomly from integers in the range $[0, 1, \ldots, \lfloor 10^b \rfloor]$. All 9! solutions are enumerated for each problem instance and the length scale values are calculated using Algorithm 6.1. Because there are ten TSP instances for each value of b, the NCD between TSP instances at two different b values is estimated by averaging the NCD for 10 combinations of instances.

Figure 8.31 shows the NCD values between problem instances parameterised by b, averaged over the ten trials. The NCD values spanned from 0.3150 to 1, indicating that there are a wide range of strong similarities and differences in the set. The heatmap shows very interesting behaviour between particular combinations of b. Firstly, comparing instances from low values of b ($b < 1.5$) to *any* other instance results in high dissimilarity (i.e. they are structurally very different). This holds true for high values of b ($b > 5.5$) as well. Indeed, most problem pairs have quite high dissimilarity, except for an area along the diagonal where $2.5 \leq b \leq 4.5$. Problem pairs off the diagonal in this range (meaning quite different b values)

[1] Figure 7.72 shows that the phase transition occurs between approximately $1.4 \leq b \leq 3.7$.

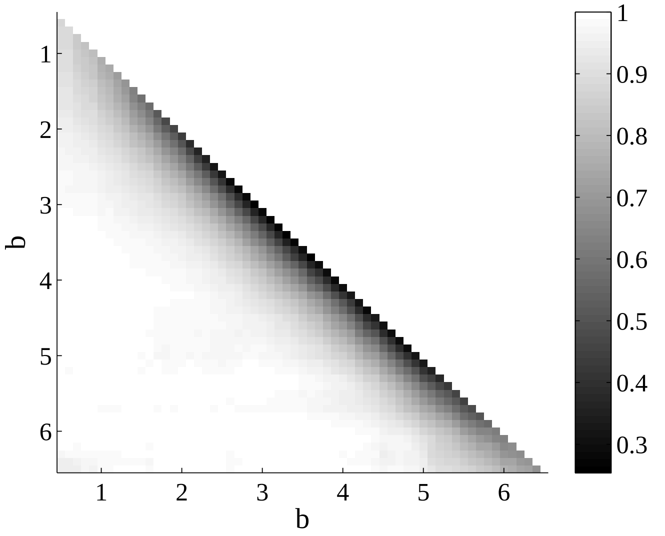


Figure 8.30: Heatmap of the mean NCD values calculated (using LZMA) between pairs of random ATSP instances, generated with $b = 0, 0.1, \dots 6.5$.

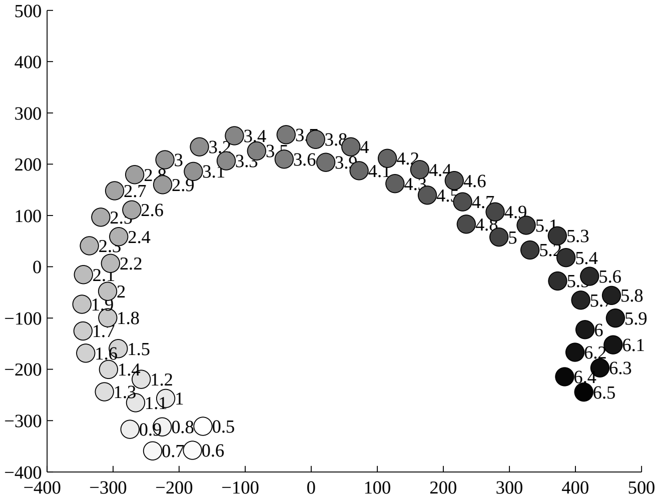


Figure 8.31: t-SNE of the mean NCD values (cost of 0.0419) calculated (using LZMA) between pairs of random ATSP instances, generated with $b = 0, 0.1, \dots 6.5$.

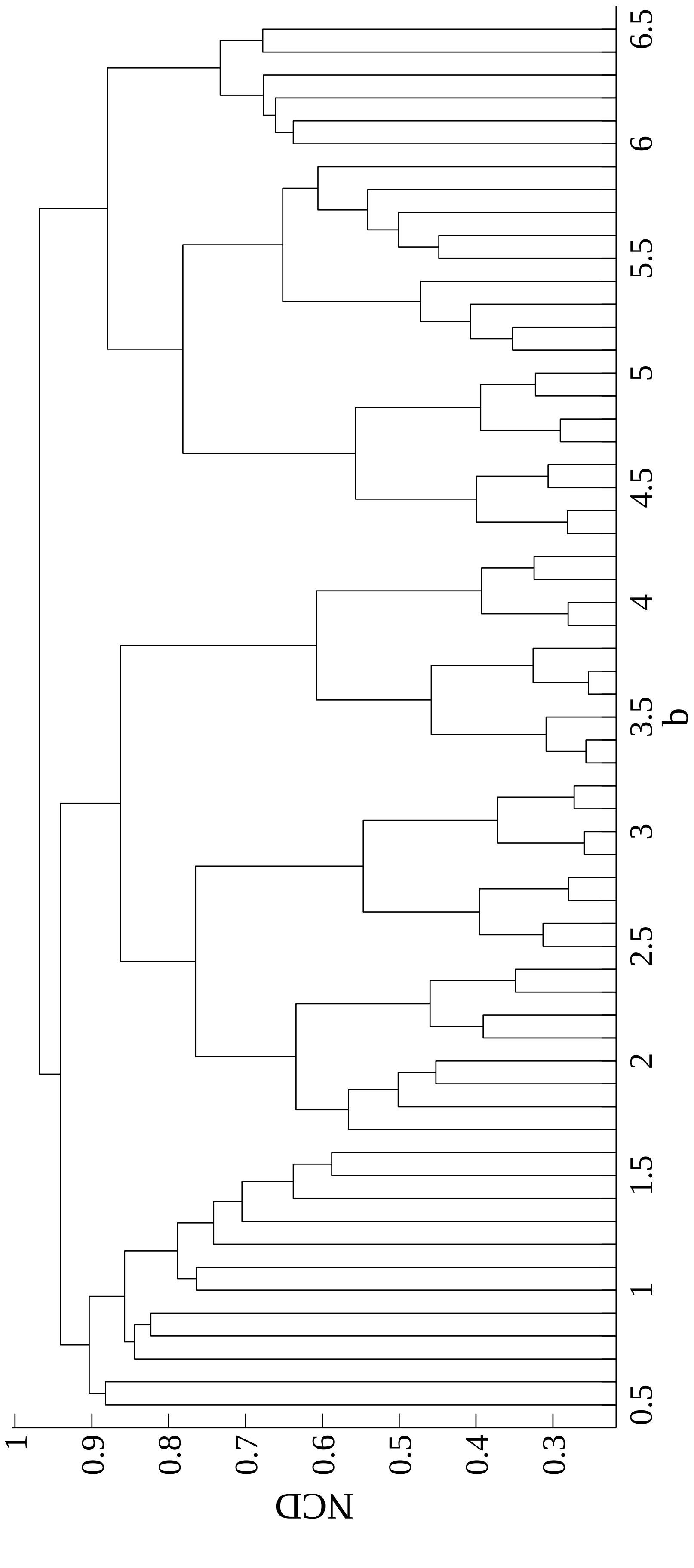


Figure 8.32: Dendrogram of the mean NCD values calculated (using LZMA) between pairs of random ATSP instances, generated with $b = 0, 0.1, \ldots 6.5$.

have a much lower NCD. This indicates that for the entire problem set, the only problems that are structurally related (as measured by NCD) are problems with similar values of b in the range $2.5 \leq b \leq 4.5$ (e.g. $b = 3$ and $b = 3.1$).

The problem space visualisation resulting from applying t-SNE (with perplexity 15) to the average NCD values is shown in Figure 8.31. The points are shaded according to their b values, with light shades corresponding to low values, and dark shades to high values. Figure 8.31 displays a progression of very structured, incrementally shaded points; problems clearly transition from white to black across the problem space. The positioning of the problems in the space forms a "U"-shape; problems after the critical point[2] in the phase transition form a "bend', while problems at either end of the transition are at the "tips" of the "U"-shape.

The dendrogram in Figure 8.32 also illustrates phase-transition behaviour. As b increases, the NCD values between problems clearly transition from high to low, and then once $b \approx 3.5$, the problems' NCD values transition from low to high. The dendrogram also shows that the problems are partitioned into three major clusters ($0.5 \leq b \leq 1.6$, $1.7 \leq b \leq 4.2$ and $4.3 \leq b \leq 6.5$), which closely correspond to the different stages of the phase transition.

8.3.8 Number Partitioning Problems

The Number Partitioning Problem (NPP), described in Appendix A.2.2, is another combinatorial optimization problem that exhibits phase transitions in the difficulty of exact solvers, as well as the number of global optima and size of plateaus [17, 22, 174]. Instances can be generated at a particular stage of the phase transition through the control parameter, k. Specifically, $k = \frac{1}{n}\log_2 m$, where n is the number of elements to partition, and the elements are integers uniform randomly chosen from $\{1, \ldots, m\}$, where $m = 2^{nk}$. Instances before the critical point $k_c = 1 - \frac{\ln\left(\frac{10\pi}{3}\right)}{40\ln(2)} \approx 0.9153$ are generally easy, while instances after k_c are hard. Therefore, the NPP provides a very good opportunity to evaluate the ability of the NCD to detect structural similarities between instances. Hence, this experiment calculates and analyses the NCDs between randomly generated instances throughout the phase transition. Ten random instances of size $n = 20$ are generated at each k, where $k = 0.4, 0.425, \ldots, 1.3$. The NCD between NPP instances at two different k values is estimated by averaging the NCD for ten combinations of instances. Figures 8.33 to 8.35 show the heatmap, t-SNE visualisation (perplexity of 4) and dendrogram of the resulting NCD values.

[2] The critical point is 2, as the expected number of distinct distances in D approaches ∞ [206].

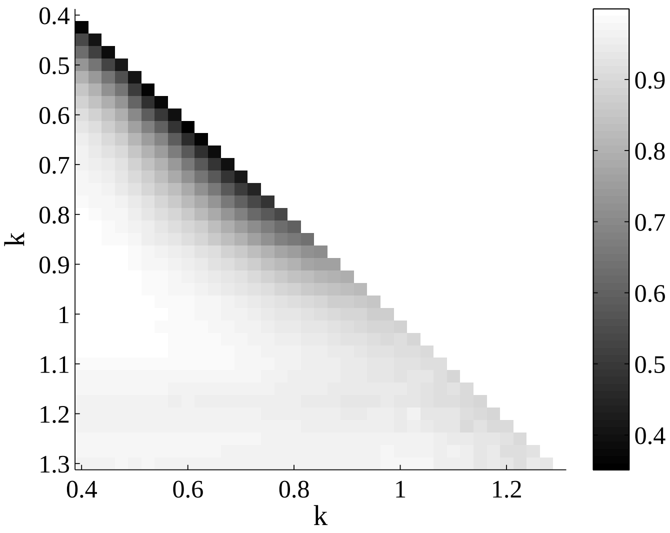


Figure 8.33: Heatmap of the NCD values calculated (using LZMA) between pairs of NPP instances, where $k = 0.4, 0.425, \ldots, 1.4$.

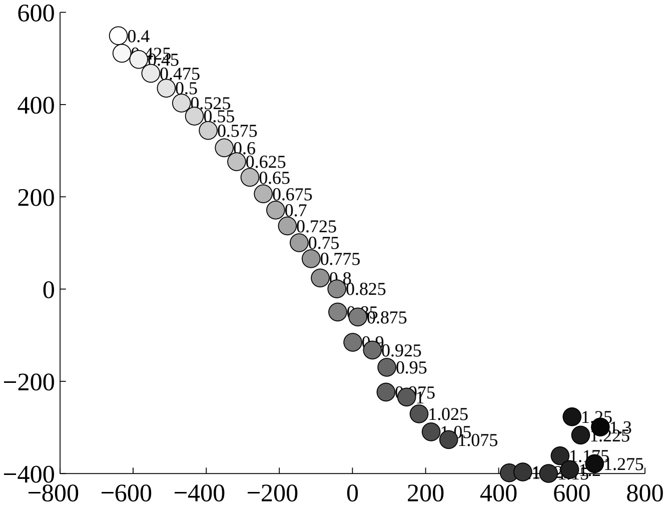


Figure 8.34: t-SNE of the NCD values (cost of 0.1158) calculated (using LZMA) between pairs of NPP instances, where $k = 0.4, 0.425, \ldots, 1.4$.

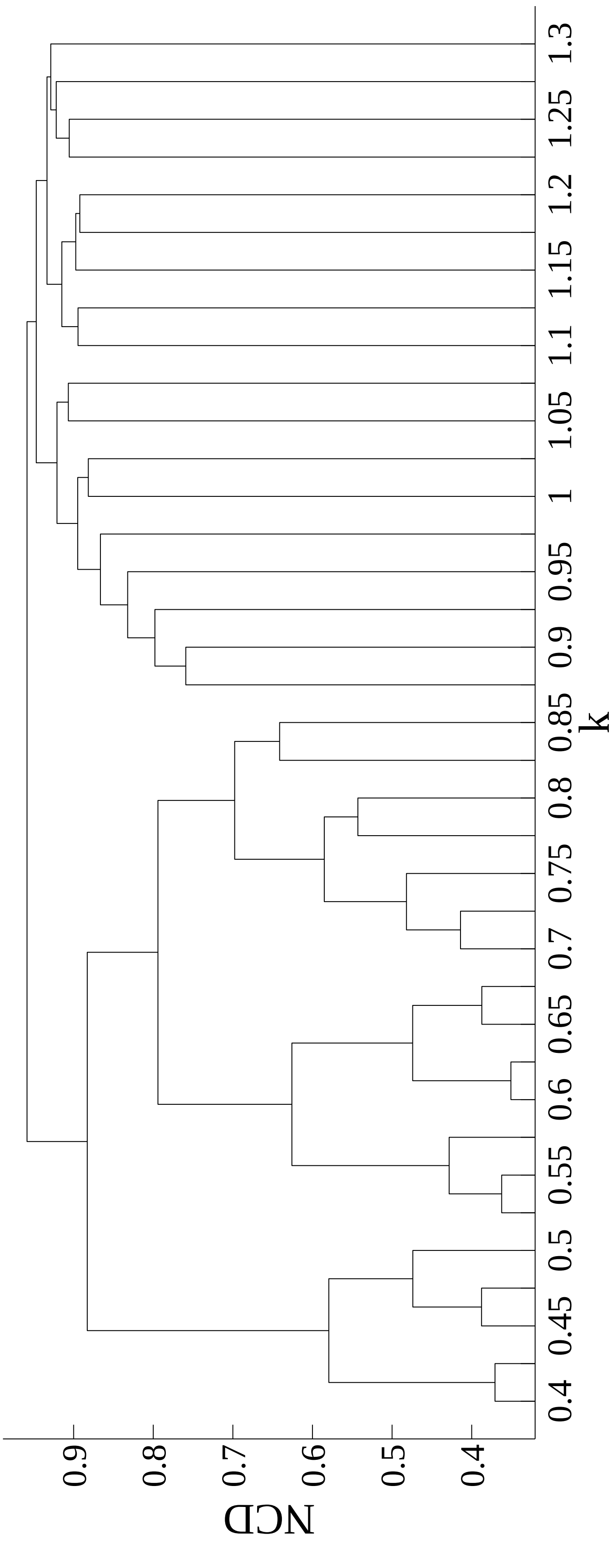


Figure 8.35: Dendrogram of the NCD values calculated (using LZMA) between pairs of NPP instances, where $k = 0.4, 0.425, \ldots, 1.4$.

The NCD values ranged between 0.3508 to 0.9994, indicating a wide range of similarities and differences between problems. The dark region along the leading diagonal of the heatmap in Figure 8.33 shows that the strong similarities occur for instances with similar values of k, where $k < 0.9$. Notably, instances along the leading diagonal where $k > 0.9$ are lightly-coloured, and hence very different. The phase transition for $n = 10$ occurs at approximately 0.9153, and so the sudden change in NCD between instances along the leading diagonal closely matches the phase transition. The results confirm that on a second class of combinatorial optimization problems (i.e. in addition to the ATSP instances in Section 8.3.7), NCD is a feature that can capture a known structural property of the problem landscape. To reiterate, this is achieved using only black-box sample information.

The t-SNE visualisation in Figure 8.34 is an accurate representation of the NCD values (i.e. cost of 0.1158). The visualisation resembles the elliptical function's t-SNE visualisation in Figure 8.5; as k increases, the problems are located in a linear trend, with the exception of instances $k > 1.075$, which curve away from the remaining problems. These instances belong to the "hard" stage of the phase transition, and so the change in trend is likely because the problems are increasingly more different. The phase transition occurs at 0.9153, however there is no noticeable change in the location of the problems in the t-SNE at this point. Therefore, the t-SNE visualisation reflects that the hard problems are quite different to the problems before the phase transition, however the location of the phase transition is not clearly discernible from Figure 8.34.

In contrast, the change in problem similarity as the phase transition occurs is nicely illustrated from the dendrogram (Figure 8.35). Analysing the dendrogram from the top, the instances can be clustered into two major groups; $k \leq 0.85$ and $k > 0.875$. This is very close to the theoretical phase transition. In addition, the cluster containing instances where $k \leq 0.85$ generally shows that for low k, the instances are very similar to each other (e.g. the NCD connecting 0.4 and 0.45 is very low). In contrast, the cluster containing instances $k > 0.875$ shows that the NCD values between instances is very high (e.g. the NCD connecting 1 and 1.025 is very high). This suggests that instances in the former cluster are quite similar, while instances in the latter are quite different. The results in the dendrogram corroborates well with the interpretations of the heatmap and t-SNE visualisation.

8.4 Quantifying Problem Similarity: Kolmogorov vs Shannon

This chapter proposed and demonstrated the utility of the Normalised Compression Distance (NCD) as a practical measure of black-box problem similarity. The calculation of the NCD requires estimates of the Kolmogorov Complexities of problems' *length scale multisets* (denoted by $\mathbf{r}$), which in this work are approximated using a lossless compressor, Z. In Chapter 6, the Shannon entropy of the *length scale distribution* (denoted by $h(r)$) is used to quantify the amount of information in an optimization problem. Restated, $Z(\mathbf{r})$ and $h(r)$ each analyse the information in optimization problems using Kolmogorov Complexity theory and Shannon information theory respectively. From a philosophical perspective on information, Kolmogorov Complexity theory and Shannon information theory are related. This section theoretically and experimentally explores the relationship between Kolmogorov Complexity theory and Shannon information theory, with a particular focus on the novel contributions proposed in this thesis.

8.4.1 Kolmogorov Complexity vs Shannon Entropy

Kolmogorov Complexity theory and *Shannon information theory* (often referred to as simply *information theory* or *classical information theory*) are both theoretical frameworks concerned with analysing and quantifying the information in objects [68]. While the theories share a common aim, the underlying assumptions regarding the objects of interest are quite different. In particular, Shannon's theory assumes that an object is an outcome of a known random source (i.e. a known distribution), and the Shannon entropy aims to quantify the minimum information required to communicate an arbitrary object, given that the recipient has knowledge of the source. In contrast, Kolmogorov Complexity is concerned with the amount of information required to describe an object, and no prior knowledge regarding an object's source/distribution is assumed [68]. Consider a data source that emits only two messages with equal probability; $m1$ and $m2$. In the Shannon framework, it is assumed that the recipient of the message knows all possible messages that can be sent (thus while the outcome is random, it is from a known distribution of outcomes). Under this assumption, a message can be communicated using a single bit by allowing 0 to denote $m1$ and 1 to denote $m2$. Hence in this scenario, the Shannon entropy for communicating an arbitrary message, out of only two possible messages, is 1 bit. However, the amount of information within the messages themselves can be made arbitrarily large. For example, let $m1$ be "I will be late for dinner" and $m2$ be "0111100011101010". The amount of information in each of these messages, and hence their respective Kolmogorov Complexity, is clearly more than 1 bit. Thus in general, Shannon information theory is predominately applied to quantifying the information required to unambiguously communicate objects, while Kolmogorov Complexity theory is concerned with unambiguous descriptions of the objects themselves.

Kolmogorov Complexity and Shannon entropy have an interesting theoretical relationship; assuming that a given object comes from a computable distribution, the Shannon entropy is (loosely speaking) approximately equal to the object's *expected* Kolmogorov Complexity. Formally, this is described by the following Theorem [68, 96]:

Theorem 8.1. *Let P be a computable probability distribution on the set of binary strings of arbitrary length, {0,1}*. Then,*

$$0 \leq \sum_{x} P(x)K(x) - H(P) \leq K(P) + O(1) \tag{8.4.1}$$

Applied to the concept of length scale (Section 6.1.3), the Shannon entropy of the length scale distribution quantifies the information in the random variable, r. Hence, $h(r)$ in Chapter 6 quantifies the information in a *single* length scale value, with respect to the distribution of r values sampled. In contrast, the lossless compression of the length scale multiset (used to approximate the Kolmogorov Complexity) quantifies the information in the *multiset* of length scale values. Therefore, the objects analysed in this thesis by the Shannon and Kolmogorov Complexity theories are not the same; $h(r)$ quantifies the information in a single length scale, while $Z(\mathbf{r})$ quantifies the information in the length scale multiset, $\mathbf{r}$. Consequently, Theorem 8.1 is not directly applicable in theoretically relating $h(r)$ and $Z(\mathbf{r})$.

8.4.2 Results: Normalised Compression Distance vs Jeffrey Divergence

While the Shannon entropy and Kolmogorov Complexity approaches proposed in this thesis to characterise and compare optimization problems operate on slightly different objects, there is still clearly a deep philosophical relationship between them. For example, the NCD and D_J are both information-theoretic based measures of optimization problem similarity. Specifically, given two optimization problems, NCD and D_J quantify their similarity using Kolmogorov Complexity theory and Shannon information theory respectively. To investigate the relationship between the two measures, the NCD and D_J values from problems

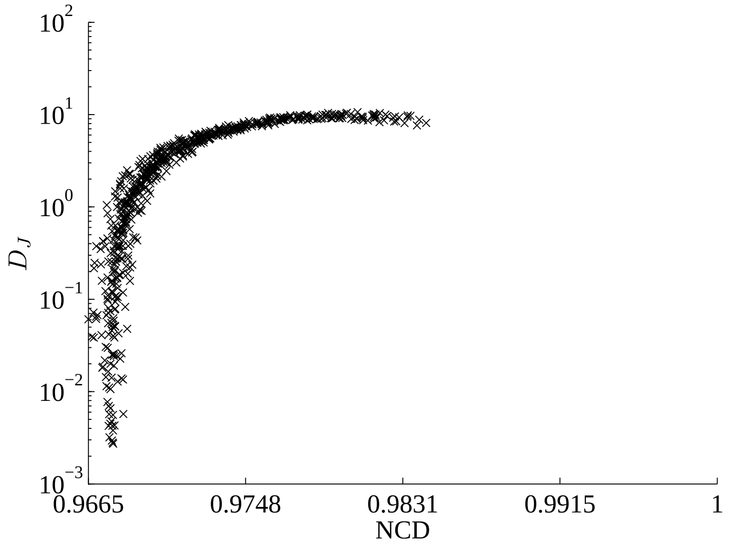


Figure 8.36: NCD vs D_J for 2-D elliptical functions, where $a = 1, 1.25, \ldots, 10$.

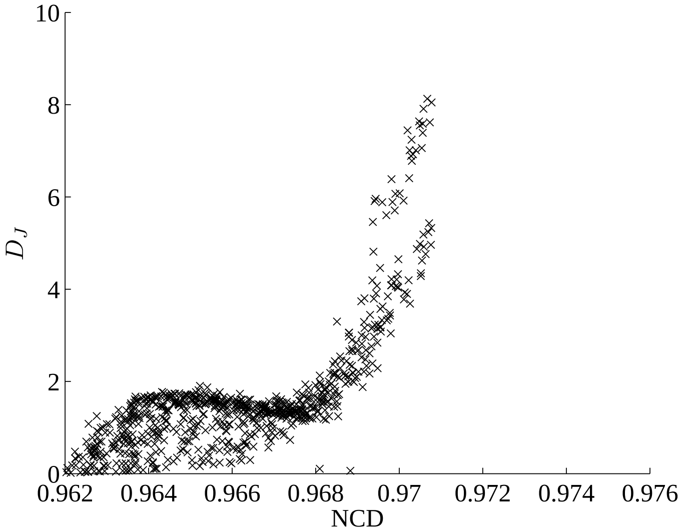


Figure 8.37: NCD vs D_J for 1-D Rastrigin functions, where $A = 0, 0.25, \ldots, 10$.

analysed in this chapter and Chapter 7 are empirically compared. In particular, the resulting NCD and D_J values for the elliptical functions, Rastrigin problems, BBOB problems, CiaS problems, TSPLib instances, ATSPs and NPPs are plotted against each other in Figures 8.36 to 8.42.

The relationships between the NCD and D_J values calculated for the 2-D elliptical functions (Figure 8.36), 1-D Rastrigin functions (Figure 8.37), and CiaS problems (Figure 8.39) are non-linear and monotonic. There is a much more complex relationship between the NCD and D_J values calculated between instances of BBOB problems (Figure 8.38), TSPLib (Figure 8.40), the ATSP (Figure 8.41) and the NPP (Figure 8.42). Despite the non-linearity and complexity of the relationships between D_J and the NCD displayed in Figures 8.36 to 8.42,

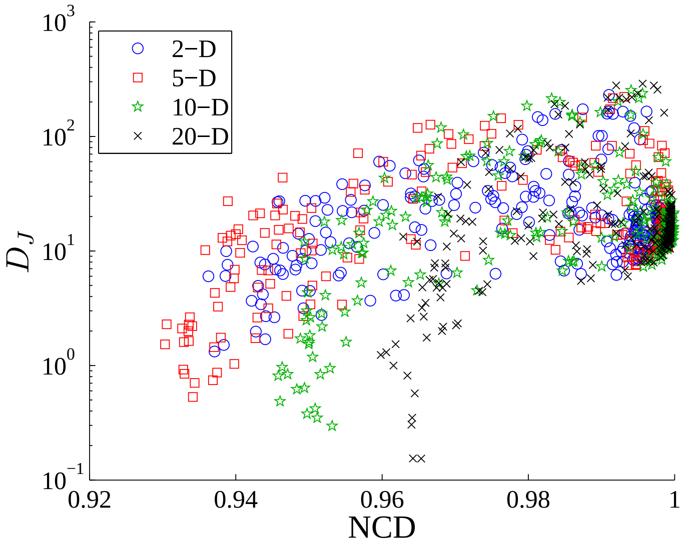


Figure 8.38: NCD vs D_J for 2, 5, 10 and 20-D BBOB problems.

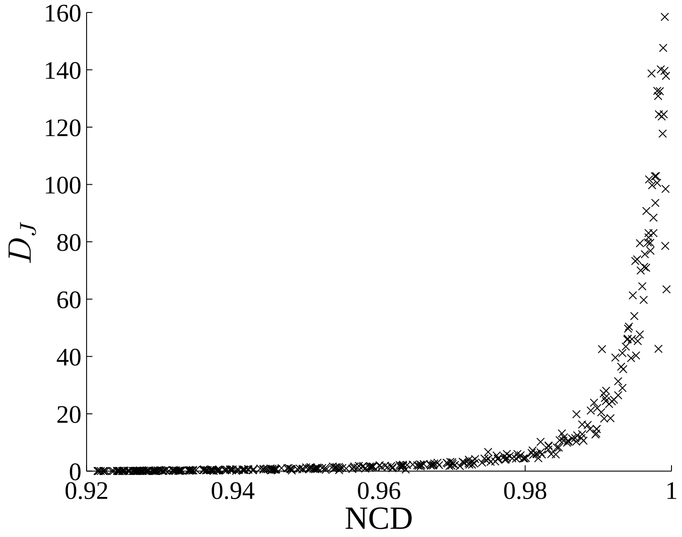


Figure 8.39: NCD vs D_J for Circle in a Square problems, where $n_c = 2, 3, \ldots, 100$.

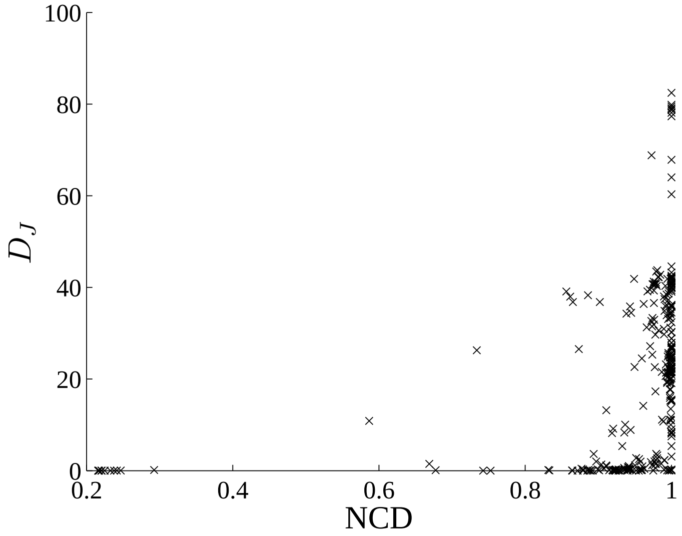


Figure 8.40: NCD vs D_J for TSPLib instances.

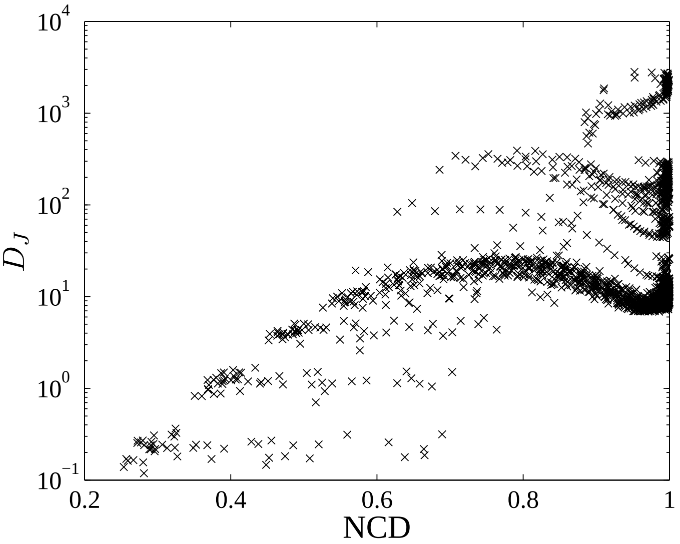


Figure 8.41: NCD vs D_J for Asymmetric Travelling Salesman Problem instances, randomly generated with $b = 0, 0.1, \ldots 6.5$.

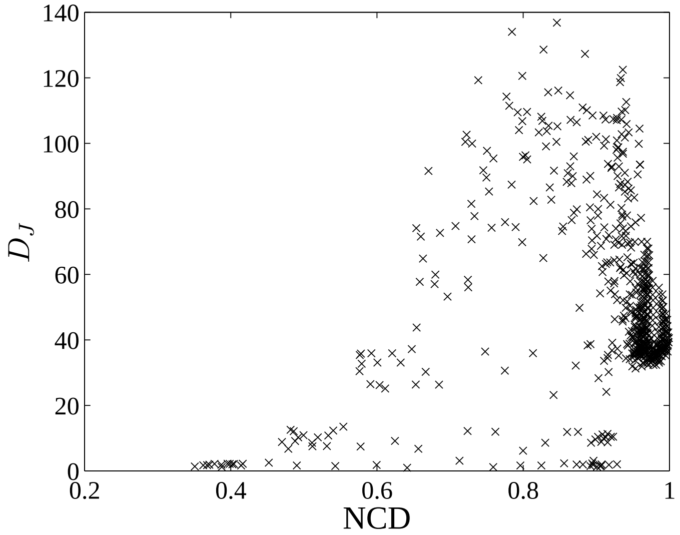


Figure 8.42: NCD vs D_J for Number Partitioning Problem instances, where $k = 0.4, 0.425, \ldots, 1.4$.

Problem	**Pearson's ρ_p**	**Spearman's ρ_s**	**Kendall's τ**
Elliptical Function	0.9469	0.9726	0.8618
Rastrigin	0.7241	0.6840	0.5070
Circle in a Square	0.6515	0.9863	0.9067
BBOB 2D	0.1590	0.1472	0.1057
BBOB 5D	0.1490	0.2522	0.1741
BBOB 10D	0.088 (0.1591)	0.1879	0.1356
BBOB 20D	0.0684 (0.2787)	0.2337	0.1892
TSPLib	0.3442	0.5721	0.4203
ATSP	0.1707	0.2242	0.1522
NPP	0.1919	-0.1161	-0.1107

Table 8.3: Correlation coefficients between NCD and D_J with $\alpha = 0.05$

the relationship is clearly non-random. This suggests that the two measures share similarities as well as differences in the structural features that they capture.

In order to explicitly quantify the correlation between NCD and D_J, the Pearson's ρ_p, Spearman's ρ_s and Kendall's τ correlation coefficients are calculated using a significance level of $\alpha = 0.05$. Pearson's ρ_p coefficient measures the linear correlation between the explicit NCD and D_J values. To quantify the extent of non-linear correlation, ρ_s and τ utilise only rank information regarding the NCD and D_J values, thereby measuring how well the NCD and D_J can be represented with a monotonic function. The resulting correlation coefficients are summarised in Table 8.3. The resulting p-values are all less than 0.05 (with the exception of ρ_s for 10-D and 20-D BBOB problems, whose p-values are reported in brackets), meaning that the correlations are statistically significant.

The 2-D elliptical and 1-D Rastrigin functions yielded high ρ_p coefficients in Table 8.3, indicating that there is a linear correlation between the NCD and D_J values. The remaining problems have relatively small ρ_p coefficients, however the ρ_s and τ coefficients for the CiaS problems indicates that there is a strong non-linear relationship between the NCD and D_J values. The TSPLib instances also have moderate ρ_s and τ values, suggesting some degree of non-linear correlation. The NPP instances yield a positive ρ_p value, yet negative ρ_s and τ values, indicating that linear correlation is not a good summary of this data.

8.4.3 Discussion

Theoretical analysis into the relationship between the NCD and D_J measures of problem similarity investigated in Section 8.4.1 showed that the measures cannot be directly related. Subsequent comparisons of empirical results in Section 8.4.2 suggested that the relationship between the measures is generally very complex and highly non-linear. Hence while both NCD and D_J are measures of problem similarity, an important research direction is to determine the differences between them, and in particular, the structural features that each of the measures use to discriminate between problem landscapes.

There are no sets of problems for which the "ground truth" similarity is known, and so a rigorous comparison of the similarity measures' accuracies cannot be made. However, it must be stressed that *both* the D_J and NCD measures of problem similarity produced significant results on the continuous and combinatorial problems analysed in this thesis. Clearly, both measures are valuable, complementary measures of problem landscape similarity. Thus in practice, both measures (or a measure based on some combination of the two) can be used to analyse and compare optimization problems.

The NCD and D_J approaches both require a sample of n length scale values. The experiments in this thesis used the same sampling technique to obtain r values for both of the measures. The calculation of D_J requires estimation of Equation 6.1.8, which is performed in practice by evaluating m points on kernel density estimators constructed from each of the problems' multisets of (sampled) r values. As previously discussed in Section 6.3, the J-divergence between two D-dimensional problems requires $O(mn^3D)$ time (where n is the size of the sample, and m is the number of evaluation points), and $O(n)$ space if the length scales and/or kernel density estimators are stored.

The main disadvantage to the D_J measure is its large running time. That said, one major advantage of the D_J approach is that it computes length scale distributions, which can be subsequently analysed to gain additional insights into problem structure (e.g. see Section 6.1.3). Hence, while the estimation of D_J may be computationally expensive, it has the inherent benefit of additionally analysing the computed models for valuable insights.

As outlined in Section 8.2.3, the NCD methodology runs in $O(n)$ time. Hence, the NCD methodology may be preferable in situations where running time is an important consideration and computation of D_J is infeasible. The NCD approach is also arguably simpler to implement; it relies solely on samples of length scale values from the problems and a suitable compressor for the length scale data. In addition, the NCD approach can theoretically be used to compare optimization problems with completely different types of objects (e.g. music, images and text files).

The NCD and D_J developed in this thesis measure problem similarity, and they differ in their resulting values, computational complexities, ease of implementation and capacity for additional analysis. Hence, to answer the question proposed at the beginning of this section, *both* measures are useful, but each methodology comes with different benefits and limitations that may be important or irrelevant in particular scenarios and applications.

8.5 Summary

This chapter proposed using the (Normalised) Information Distance as a measure of black-box optimization problem similarity. Normalised Information Distance relies on the Kolmogorov Complexity of the problems, and thus is purely a theoretical notion. In practice, lossless compression algorithms can be used to approximate Kolmogorov Complexity. Hence by utilising lossless compression, the Normalised Compression Distance (NCD) can be used to quantify problem similarity in practice.

The success of the NCD relies on two major components; the binary string used to represent a given optimization problem and the compressor used to compress the binary string. The limitations of numerous binary representations were discussed and the multiset of sampled length scales was proposed as a suitable representation. In addition, the suitability of numerous well-known general-purpose compressors as well as compressors suitable for length scale data were discussed and empirically compared using length scale data. The results suggest that many of the compressors are suitable for length scale data, and that the general-purpose compressor LZMA is slightly superior.

The NCD was then used to measure the similarity between a wide variety of continuous and combinatorial optimization problems. The problems consisted of artificial problems (with known structural features), benchmark problems (with known/conjectured structural features) and real-world-like problems. The problems were treated as a black-box; only the solutions and their respective objective function values were available to the NCD methodology The resulting similarity values were subsequently analysed and interpreted using heatmaps, dendrograms of hierarchical clustering and visualisations of the problem space produced by t-SNE. The results clearly demonstrated that the NCD is able to capture known structural similarities as well as phase-transition behaviour, purely via black-box information.

Previous work in this thesis proposed using the J-divergence to measure problem similarity. At a fundamental level, both of the NCD and D_J measures use information to quantify problem similarity. In addition to exploring the philosophical relationship between NCD and D_J, this chapter also empirically investigated the relationship between NCD and D_J in practice. Resulting NCD and D_J values for many of the problems analysed in this thesis suggests that the two measures have a complex, non-linear relationship and are therefore complementary to each other. From a practical perspective, NCD is arguably easier to implement and has a faster running time than D_J, but D_J utilises models of the length scale distributions, which can be additionally analysed for further insights into the problems.

Overall, the NCD is a general, yet powerful similarity measure; it relies purely on black-box information and can be readily applied to both continuous and combinatorial optimization problems. Experimental analysis of the similarities between artificial, benchmark and real-world-like problems demonstrated a strong ability to capture known structural similarities, dissimilarities and phase transitions.

CHAPTER 9

Conclusion

Science never solves a problem without creating ten more.
George Bernard Shaw

This chapter concludes the thesis by reflecting on the work's novel contributions. Section 9.1 summarises the contributions of each chapter by highlighting important arguments, concepts, methodologies and results. Limitations of the work and potential avenues for future work are discussed in Section 9.2.

9.1 Summary and Conclusions

Chapter 2 formally introduces the optimization problem, and outlines several relevant concepts and definitions, including the notion of a *landscape* as a model of the relationship between candidate solutions and their objective function values. A review of landscape definitions in the literature shows that there are several competing definitions with varying degrees of rigor. In order to consolidate the literature, a landscape definition - based on the objective function, search space and a suitable distance function - is provided. An examination of the topological concepts and notions of ruggedness, ridges, valleys, plateaus and funnels shows that many of these concepts lack rigorous definitions. In addition, it is unclear how the topological notions scale with dimensionality, and in particular, whether the landscape descriptors are able to capture the complexity of high dimensional structures. An argument is made for the development of problem features and properties that are not derived from two and three dimensional landscape intuition.

Chapter 3 reviews the ability of problem landscape features and analysis techniques proposed in the literature to analyse and characterise black-box optimization problems. While techniques from the evolutionary computation literature are predominately reviewed, related analysis in the geography, ecology, biology, chemistry and physics literature is also discussed. In contrast to continuous optimization, a considerable amount of landscape analysis has been developed for combinatorial optimization. There are many important differences between combinatorial and continuous optimization that can negatively affect the adaptation of techniques between the domains. Despite this, many techniques applied to continuous problems originate from combinatorial optimization. Regardless of whether problem features originate in the combinatorial or continuous domain, it is shown that existing techniques are quite limited. One significant limitation is the inability of the techniques to fully utilise all of the black-box information available. Furthermore, many techniques compress landscape information into a single scalar value, which in turn leads to information loss and a lack of discriminatory power.

Landscape analysis techniques rely heavily on finite samples of solutions, and Chapter 4 reviews the methodologies commonly employed to sample continuous optimization problems. The review raises serious concerns regarding the efficacy of the sampling methodologies commonly used in high dimensional continuous problem analysis. In order to address the concerns, a Lévy random walk is proposed for the analysis of high dimensional continuous problems. Two case studied are conducted to investigate the affect of the concerns and sampling adequacy of the Lévy random walk. The first study, conducted on the *dispersion* metric, shows that the use of uniform random sampling in conjunction with Euclidean distance (as is often used in the literature) is flawed and results in convergent dispersion values as dimensionality increases. Encouragingly, the Lévy random walk reduces the convergence of the dispersion values. In the second case study, *fitness distance correlation* is shown to be similarly affected in the black-box scenario. As for dispersion, the negative effects are reduced by the Lévy random walk.

Chapter 5 proposes the notion of *length scale* as a fundamental feature of problem landscapes. The length scale intuitively measures the magnitude of objective function difference with respect to a step between two solutions in the search space. Several important properties and summaries of length scale information are described, including the *length scale distribution*. In practice, length scale values can be calculated from a finite sample of candidate solutions and their objective functions values, and the length scale distribution can be estimated using kernel density estimation. While length scale is related to the notion of *finite differences* and the *Lipschitz constant*, it uniquely captures information regarding *all* rates of change, over a wide variety of intervals (distances) on the problem.

In Chapter 6, analysis techniques from set theory, statistics, machine learning and visualisation are proposed to analyse and compare problems based a sample of length scale values. One major contribution of this chapter is the application of the *Jeffrey divergence* (J-divergence) between length scale distributions to explicitly quantify the similarity between optimization problems. That is, given two optimization problems, the J-divergence between the problems' respective length scale distributions is a proxy for problem similarity. A unique methodology based on the length scale distribution is also developed in order to assess the adequacy of a sample of solutions and respective objective function values. Practical considerations are discussed, including the time and space complexities of sampling, density estimation, and the calculation of the J-divergence.

Chapter 7 investigates the ability of the length scale framework to analyse and compare optimization problems in practice. Specifically, continuous artificial problems with adjustable landscape structures, a popular benchmarking set, real-world representative geometric packing problems, Traveling Salesman Problems (TSP) and Number Partitioning Problems (NPP) are analysed and compared using the length scale distribution and J-divergence. Visualisation of the distributions, as well as heatmaps, hierarchical clustering and dimensionality reduction of J-divergences clearly shows known structural features and similarities between problems. Remarkably, the length scale analysis of TSP and NPP instances generated along known phase transitions is able to detect the phase transitions. The patterns and trends evident in the analyses are consistent, indicating that any limitations of the analysis techniques (e.g. poor parameter settings) do not significantly impact on the results. A comparison with state-of-the-art landscape features (correlation length, dispersion, fitness distance correlation, information content, partial information content and information stability) shows that the length scale framework provides valuable insights into the nature of the problems, is statistically robust, and adept at characterising and differentiating between problems. The variety of problems analysed demonstrates the flexibility of the length scale framework, and in particular, how easily it can be applied to both continuous and combinatorial problems.

While the J-divergence can be used to explicitly quantify optimization problem similarity, Chapter 8 proposes an alternative similarity measure, based on a universal distance function in Kolmogorov Complexity theory known as *Information Distance*. Information Distance is a theoretical measure, but in practice is approximated by the *Normalised Compression Distance* (NCD). A review of Kolmogorov Complexity theory and Information Distance is provided, and limitations of the existing Kolmogorov Complexity analysis in the optimization literature are identified. To overcome these limitations, a unique methodology for calculating the NCD between optimization problems is developed based on finite sets of length scale values. Practical considerations regarding the compression of length scale values are discussed, and experimental results suggest that the general compressor, LZMA, is suitable for compressing length scale values. The NCD methodology is applied to the continuous artificial problems, benchmarking set, geometric packing problems, TSPs and NPPs. The results clearly demonstrate that the NCD is able to capture known structural similarities as well as phase-transition behaviour, purely via black-box information. At a fundamental level, both the NCD and J-divergence problem similarity measures developed in this thesis use the length scale *information* to quantify problem similarity. The philosophical relationship between NCD and J-divergence is discussed, and theoretical analysis shows that while the measures are similar, they operate on subtly different summaries of length scale information. A comparison of the resulting NCD and J-divergence values for many of the problems analysed in this thesis suggests that the two measures have a complex, non-linear relationship and may provide complementary information. From a practical perspective, NCD is arguably easier to implement and has a faster running time than the J-divergence, but J-divergence utilises important models that can be additionally analysed for further insights into the problems.

Overall, the contributions in this thesis together form a framework and practical techniques to study the structural characteristics of a problem landscape, independent of any particular optimization algorithm. Results on a variety of continuous and combinatorial optimization problems clearly demonstrate the ability of the framework to detect important structural features from purely black-box information (i.e. finite samples of candidate solutions and their respective objective function values). Importantly, the developed framework is easy to implement, applicable to **both** continuous and combinatorial problems, and highly amenable to the incorporation of additional analysis techniques.

9.2 Limitations and Future Work

The length scale framework is clearly able to capture the structural features that are required to distinguish and differentiate between problems. It would be interesting to relate particular summaries of length scale values to well-defined topological properties. For example, it should be possible to explore the relationship between landscape modality and the shape of the length scale distribution.

While the techniques used to experimentally analyse length scale values provided valuable insights, the effectiveness of the techniques are influenced by many factors, including parameterisations and the overall appropriateness of the techniques for the task at hand. For example, the use of t-SNE for producing a two-dimensional representation of the relationships between problems (according to their J-divergences) assumes that the relationships can adequately be represented in two-dimensional space. While the majority of the resulting t-SNE reductions in this thesis produced accurate (i.e. low-cost) representations (e.g. Figures 8.14 to 8.17), inaccurate results were produced in certain instances (e.g. Figure 8.5).

The ordering and organisation of problems in the heatmaps and dendrograms can be difficult for instances with no obvious ordering, such as the Black-Box Optimization Benchmarking (BBOB) problem set and TSPLib. For example, in the TSPLib problem set, problems are grouped along the heatmap axes according to their source (see Figures 7.69 and 8.27), however the groups are ordered arbitrarily. In such circumstances, trends in heatmaps can be difficult to observe. Encouragingly, the conclusions drawn from the distribution plots, heatmaps, dendrograms and t-SNE visualisations were largely consistent across the visualisations. This suggests that the techniques are reliable and that the inherent limitations of each methodology do not significantly impact on results.

There is considerable scope to apply a range of other techniques for modelling and summarising length scale information. The experiments in this thesis predominately used entropy to summarise the length scale distribution, but other ideas from statistics and information theory (including those already discussed in Section 6.1) deserve investigation. In particular, the current methods used to analyse length scales ignore spatial information, such as the locations of the solutions. More discriminative ability may be possible by combining complementary summaries of length scale information, as well as existing landscape features.

The set of solutions evaluated by an algorithm during a run could also be analysed in the length scale framework. Comparisons between the landscape's length scales and the algorithm's resulting length scales can be made, and possible insights into algorithm behaviour may be drawn. In addition, this information could potentially be used to make online algorithm parameter adjustments. Similar to the comparisons of problem length scale distributions conducted in this thesis, the length scale distributions of the solutions produced by algorithm instances could also be directly compared to each other. Exploration of the relationship between algorithm performance and length scale metrics (e.g. the entropy of the length scale distribution, $h(r)$) is also an interesting avenue for future work. In Section 7.2.4, algorithm performance results for the BBOB problems were examined against $h(r)$, however it would be very interesting to investigate the ability of $h(r)$ to discriminate or predict algorithm results for the *other* problems in this thesis, such as TSP or NPP instances along the phase transitions.

The experiments conducted in this thesis indicate that the J-divergence and NCD measures are powerful problem discriminators. Algorithm prediction models often utilise feature ensembles to compare and discriminate between problems, and hence the use of the J-divergence and/or NCD as problem discriminators in algorithm prediction models is a promising avenue for future work.

In order to adequately yet efficiently test the performance of algorithms, benchmark problem sets ideally contain a wide variety of landscape structures with only a modest number of problems. The J-divergence and NCD results for the BBOB and TSPLib problems show that there are some similarities, and hence, redundancies in these benchmark sets. By removing the redundant problems, the size of a benchmark set may be reduced without significantly affecting the coverage of problem structures. Hence, the J-divergence and NCD measures developed could be used to identify benchmark redundancies, leading to more efficient benchmark testing. The J-divergence and NCD measures could also be used to investigate the degree to which benchmark functions reflect real-world problems, which is a research area of interest and debate in the optimization community.

Thousands of new metaheuristic algorithms are proposed each year with the aim of improving the performance of existing algorithms and solving new/unsolved problems. A greater understanding of the structural features in the optimization problems of interest will likely lead to better-designed algorithms for solving them. Landscape features, such as the length scale framework developed in this thesis, shed light into the nature of black-box problems and may be used to analyse algorithm behaviour, design better heuristics and influence/guide heuristics *during search*. Certain algorithm frameworks already utilise basic landscape features (e.g. hyperheuristics and algorithm portfolios), and will greatly benefit from the development of more descriptive and discriminating features.

References

[1] D. Achlioptas, A. Naor, and Y. Peres. Rigorous location of phase transitions in hard optimization problems. *Nature*, 435(7043):759–764, 2005.

[2] B. Addis, M. Locatelli, and F. Schoen. Disk packing in a square: a new global optimization approach. *INFORMS Journal on Computing*, 20(4):516–524, 2008.

[3] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the International Conference on Database Theory (ICDT'01)*, pages 420–434, London, UK, 2001. Springer.

[4] E. Alpaydin. *Introduction to machine learning*. MIT press, 2nd edition, 2010.

[5] L. Altenberg. Fitness distance correlation analysis: An instructive counterexample. In *Proceedings of the International Conference on Genetic Algorithms*, pages 57–64, San Francisco, USA, 1997. Morgan Kaufmann.

[6] K. Alyahya and J. E. Rowe. Phase transition and landscape properties of the number partitioning problem. In *Evolutionary Computation in Combinatorial Optimisation*, pages 206–217. Springer, 2014.

[7] T. Bartz-Beielstein, M. Chiarandini, L. Paquete, and M. Preuss. *Experimental methods for the analysis of optimization algorithms*. Springer, 2010.

[8] H. Bauke, S. Franz, and S. Mertens. Number partitioning as a random energy model. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(04):P04003, 2004.

[9] O. M. Becker and M. Karplus. The topology of multidimensional potential energy surfaces: Theory and application to peptide structure and kinetics. *The Journal of Chemical Physics*, 106(4):1495–1517, 1997.

[10] G. Beliakov. Interpolation of Lipschitz functions. *Journal of Computational and Applied Mathematics*, 196(1):20 – 44, 2006.

[11] C. H. Bennett, P. Gács, M. Li, P. M. B. Vitányi, and W. H. Zurek. Information distance. *IEEE Transactions on Information Theory*, 44(4):1407–1423, 1998.

[12] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(1):281–305, 2012.

[13] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is ″nearest neighbor″ meaningful? In *Proceedings of the International Conference on Database Theory (ICDT'99)*, pages 217–235, London, UK, 1999. Springer.

[14] L. T. Biegler. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*, volume 10. SIAM, 2010.

[15] B. Bischl, O. Mersmann, H. Trautmann, and M. Preuss. Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'12)*, pages 313–320, New York, USA, 2012. ACM.

[16] K. Boese, A. Kahng, and S. Muddu. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 16(2):101–113, 1994.

[17] S. Boettcher and S. Mertens. Analysis of the Karmarkar-Karp differencing algorithm. *The European Physical Journal B-Condensed Matter and Complex Systems*, 65(1):131–140, 2008.

[18] Y. Borenstein and R. Poli. Fitness distributions and GA hardness. In *Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *Lecture Notes in Computer Science*, pages 11–20. 2004.

[19] Y. Borenstein and R. Poli. Information landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'05)*, pages 1515–1522, New York, USA, 2005. ACM.

[20] Y. Borenstein and R. Poli. Kolmogorov complexity, optimization and hardness. In *IEEE Congress on Evolutionary Computation (CEC'06)*, pages 112–119, 2006.

[21] Y. Borenstein and R. Poli. Decomposition of fitness functions in random heuristic search. *Foundations of Genetic Algorithms*, pages 123–137, 2007.

[22] C. Borgs, J. Chayes, and B. Pittel. Phase transition and finite-size scaling for the integer partitioning problem. *Random Structures & Algorithms*, 19(3-4):247–288, 2001.

[23] P. A. N. Bosman, J. Grahl, and D. Thierens. Enhancing the performance of maximum-likelihood Gaussian EDAs using anticipated mean shift. In *Parallel Problem Solving from Nature (PPSN X)*, pages 133–143. Springer, 2008.

[24] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[25] M. Burtscher and P. Ratanaworabhan. FPC: A high-speed compressor for double-precision floating-point data. *IEEE Transactions on Computers*, 58(1):18–31, 2009.

[26] M. Burtscher and P. Ratanaworabhan. pFPC: A parallel compressor for floating-point data. In *Data Compression Conference (DCC'10)*, pages 43–52, 2009.

[27] M. Burtscher and P. Ratanaworabhan. gFPC: A self-tuning compression algorithm. In *Data Compression Conference (DCC'10)*, pages 396–405, 2010.

[28] P. Caamaño, A. Prieto, J. Becerra, F. Bellas, and R. Duro. Real-valued multimodal fitness landscape characterization for evolution. In *Neural Information Processing. Theory and Algorithms*, volume 6443 of *Lecture Notes in Computer Science*, pages 567–574. Springer, 2010.

[29] P. Caamaño, F. Bellas, J. A. Becerra, V. Diaz, and R. J. Duro. Experimental analysis of the relevance of fitness landscape topographical characterization. In *IEEE Congress on Evolutionary Computation (CEC'12)*, pages 1–8. IEEE, 2012.

[30] P. Caamaño, F. Bellas, J. A. Becerra, and R. J. Duro. Evolutionary algorithm characterization in real parameter optimization problems. *Applied Soft Computing*, 13(4): 1902–1921, 2013.

[31] I. Castillo, F. J. Kampas, and J. D. Pintér. Solving circle packing problems by global optimization: numerical results and industrial applications. *European Journal of Operational Research*, 191(3):786–802, 2008.

[32] O. Catoni. Rough large deviation estimates for Simulated Annealing: Application to exponential schedules. *The Annals of Probability*, pages 1109–1146, 1992.

[33] M. Cebrian, M. Alfonseca, and A. Ortega. Common pitfalls using the normalized compression distance: What to watch out for in a compressor. *Communications in Information & Systems*, 5(4):367–384, 2005.

[34] P. Cheeseman, B. Kanefsky, and W. Taylor. Where the really hard problems are. In *International Joint Conference on Artificial Intelligence (IJCAI'91)*, pages 331–337. Morgan Kaufmann, 1991.

[35] X. Chen, B. Francia, M. Li, B. Mckinnon, and A. Seker. Shared information and program plagiarism detection. *IEEE Transactions on Information Theory*, 50(7):1545–1551, 2004.

[36] R. Cilibrasi and P. M. Vitányi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.

[37] R. Cilibrasi, P. Vitányi, and R. De Wolf. Algorithmic clustering of music based on string compression. *Computer Music Journal*, 28(4):49–67, 2004.

[38] P. Collard, S. Vérel, and M. Clergue. Local search heuristics: Fitness cloud versus fitness landscape. *European Conference on Artificial Intelligence*, pages 973–974, 2004.

[39] D. P. Costa, G. A. Breed, and P. W. Robinson. New insights into pelagic migrations: implications for ecology and conservation. *Annual Review of Ecology, Evolution, and Systematics*, 43:73–96, 2012.

[40] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, USA, 1991.

[41] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems*, pages 2933–2941, 2014.

[42] V. C. David Applegate, Robert E. Bixby and W. J. Cook. Concorde TSP solver. http://www.math.uwaterloo.ca/tsp/concorde/index.html, 2014.

[43] Y. Davidor. Epistasis variance: Suitability of a representation to Genetic Algorithms. *Complex Systems*, 4(4):369–384, 1990.

[44] E. Deza, M. M. Deza, M. M. Deza, and E. Deza. *Encyclopedia of Distances*. Springer, 2009.

[45] J. P. Doye. Physical perspectives on the global optimization of atomic clusters. In *Global Optimization*, volume 85 of *Nonconvex Optimization and Its Applications*, pages 103–139. Springer, 2006.

[46] B. Draskoczy. Fitness distance correlation and search space analysis for permutation based problems. In *Evolutionary Computation in Combinatorial Optimization*, pages 47–58. Springer, 2010.

[47] S. Droste, T. Jansen, and I. Wegener. Perhaps not a free lunch but at least a free appetizer. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, Lecture Notes in Computer Science, pages 833–839. Springer, 1999.

[48] H. Emmons and S. Rai. Computational complexity theory. In C. Floudas and P. Pardalos, editors, *Encyclopedia of Optimization*, number 1, pages 310–315. Springer, 2008.

[49] V. Engelson, D. Fritzson, and P. Fritzson. Lossless compression of high-volume numerical data from simulations. In *Proceedings of Data Compression Conference (DCC'00)*, page 574, 2000.

[50] T. English. Optimization is easy and learning is hard in the typical function. In *IEEE Congress on Evolutionary Computation (CEC'00)*, volume 2, pages 924–931, 2000.

[51] T. M. English. Practical implications of new results in conservation of optimizer performance. In *Parallel Problem Solving from Nature (PPSN VI)*, volume 1917 of *Lecture Notes in Computer Science*, pages 69–78. Springer, 2000.

[52] P. Ferragina, R. Giancarlo, V. Greco, G. Manzini, and G. Valiente. Compression-based classification of biological sequences and structures via the universal similarity metric: experimental assessment. *BMC Bioinformatics*, 8(1):1–20, 2007.

[53] A. Forrester, A. Sóbester, and A. Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[54] M. Gallagher. *Multi-layer perceptron error surfaces: visualization, structure and modelling*. PhD thesis, Dept. Computer Science and Electrical Engineering, University of Queensland, 2000.

[55] M. Gallagher. Fitness distance correlation of neural network error surfaces: A scalable, continuous optimization problem. In *European Conference on Machine Learning*, volume 2167 of *Lecture Notes in Artificial Intelligence*, pages 157–166, Singapore, 2001.

[56] J. Gamier and L. Kallel. How to detect all maxima of a function. *Theoretical Aspects of Evolutionary Computing, Natural Computing*, pages 343–370, 2001.

[57] R. W. Garden and A. P. Engelbrecht. Analysis and classification of optimisation benchmark functions and benchmark suites. In *IEEE Congress on Evolutionary Computation (CEC'14)*, pages 1641–1649. IEEE, 2014.

[58] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, USA, 1990.

[59] M. Gendreau and J. Potvin. *Handbook of Metaheuristics*. International Series in Operations Research & Management Science. Springer, 2010.

[60] I. Gent and T. Walsh. The TSP phase transition. *Artificial Intelligence*, 88(1-2):349–358, 1996.

[61] I. P. Gent and T. Walsh. Analysis of heuristics for number partitioning. *Computational Intelligence*, 14(3):430–451, 1998.

[62] G. Giorgi, A. Guerraggio, and J. Thierfelder. *Mathematics of Optimization: Smooth and Nonsmooth Case*. Elsevier Science, 2004.

[63] H. Gould, J. Tobochnik, and W. Christian. *An Introduction to Computer Simulation Methods: Applications to Physical Systems*. Pearson Addison Wesley, 2007.

[64] J. Grahl, S. Minner, and F. Rothlauf. Behaviour of UMDAc with truncation selection on monotonous functions. In *IEEE Congress on Evolutionary Computation (CEC'05)*, volume 3, pages 2553–2559, 2005.

[65] H. Greenside, A. Wolf, J. Swift, and T. Pignataro. Impracticality of a box-counting algorithm for calculating the dimensionality of strange attractors. *Physical Review A*, 25(6):3453–3456, 1982.

[66] C. M. Grinstead and J. L. Snell. *Introduction to Probability*. American Mathematical Society, 2012.

[67] A. Grosso, A. Jamali, M. Locatelli, and F. Schoen. Solving the problem of packing equal and unequal circles in a circular container. *Journal of Global Optimization*, 47(1): 63–81, 2010.

[68] P. D. Grünwald and P. M. B. Vitányi. Algorithmic information theory. In *Philosophy of Information*, pages 281–317. Elsevier, 2008.

[69] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[70] N. Hansen. Invariance, self-adaptation and correlated mutations in evolution strategies. In *Parallel Problem Solving from Nature (PPSN VI)*, volume 1917 of *Lecture Notes in Computer Science*, pages 355–364. Springer, 2000.

[71] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-Parameter Black-Box Optimization Benchmarking: Experimental Setup. Technical report, INRIA, 2013. URL `http://coco.lri.fr/downloads/download13.09/bbobdocexperiment.pdf`.

[72] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2010: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2013.

[73] L. Hernando, J. A. Pascual, A. Mendiburu, and J. A. Lozano. A study on the complexity of TSP instances under the 2-exchange neighbor system. In *Foundations of Computational Intelligence (FOCI'11)*, pages 15–21, 2011.

[74] W. Hordijk. A measure of landscapes. *Evolutionary Computation*, 4(4):335–360, 1996.

[75] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer, 1996.

[76] R. Horst, P. M. Pardalos, and H. Romeijn. *Handbook of Global Optimization*. Number 2 in Handbook of Global Optimization. Springer, 2002.

[77] F. Hutter, Y. Hamadi, H. Hoos, and K. Leyton-Brown. Performance prediction and automated tuning of randomized and parametric algorithms. In *Principles and Practice of Constraint Programming (CP'06)*, volume 4204 of *Lecture Notes in Computer Science*, pages 213–228. Springer, 2006.

[78] A. J. Izenman. Recent developments in nonparametric density estimation. *Journal of the American Statistical Association*, 86(413):pp. 205–224, 1991.

[79] A. J. Izenman. *Modern multivariate statistical techniques: regression, classification, and manifold learning*. Springer, 2009.

[80] T. Jansen. Black-box complexity for bounding the performance of randomized search heuristics. In *Theory and Principled Methods for the Design of Metaheuristics*, pages 85–110. Springer, 2014.

[81] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft computing*, 9(1):3–12, 2005.

[82] T. Jones. *Evolutionary algorithms, fitness landscapes and search*. PhD thesis, The University of New Mexico, 1995.

[83] T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for Genetic Algorithms. In *Proceedings of the International Conference on Genetic Algorithms*, pages 184–192, San Francisco, USA, 1995. Morgan Kaufmann.

[84] L. Kallel, B. Naudts, and C. Reeves. Properties of fitness functions and search landscapes. In *Theoretical Aspects of Evolutionary Computing*, Natural Computing Series, pages 175–206. Springer, 2001.

[85] S. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128(1):11–45, 1987.

[86] E. Keogh, S. Lonardi, C. A. Ratanamahatana, L. Wei, S.-H. Lee, and J. Handley. Compression-based data mining of sequential data. *Data Mining and Knowledge Discovery*, 14(1):99–129, 2007.

[87] P. Kerschke, M. Preuss, C. Hernández, O. Schütze, J.-Q. Sun, C. Grimme, G. Rudolph, B. Bischl, and H. Trautmann. Cell mapping techniques for exploratory landscape analysis. In *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, volume 288 of *Advances in Intelligent Systems and Computing*, pages 115–131. Springer, 2014.

[88] P. Kilby, J. Slaney, and T. Walsh. The backbone of the travelling salesperson. In *International Joint Conference on Artificial Intelligence (IJCAI'05)*, volume 19, page 175. Morgan Kaufmann, 2005.

[89] K. Klemm, C. Flamm, and P. F. Stadler. Funnels in energy landscapes. *The European Physical Journal B - Condensed Matter and Complex Systems*, 63(3):387–391, 2008.

[90] N. Krasnogor and D. A. Pelta. Measuring the similarity of protein structures by means of the universal similarity metric. *Bioinformatics*, 20(7):1015–1021, 2004.

[91] D. P. Kroese, T. Taimre, and Z. I. Botev. *Handbook of Monte Carlo Methods*, volume 706. John Wiley & Sons, 2011.

[92] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[93] J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer, 2007.

[94] M. Levandowsky and D. Winter. Distance between sets. *Nature*, 234(5323):34–35, 1971.

[95] K. Leyton-Brown, E. Nudelman, G. Andrew, J. McFadden, and Y. Shoham. A portfolio approach to algorithm selection. In *International Joint Conference on Artificial Intelligence (IJCAI'03)*, volume 1543, page 2003. Morgan Kaufmann, 2003.

[96] M. Li and P. M. B. Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer, 2008.

[97] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitányi. The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264, 2004.

[98] P. Lindstrom and M. Isenburg. Fast and efficient compression of floating-point data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1245–1250, 2006.

[99] S. Lloyd. Ultimate physical limits to computation. *Nature*, 406(6799):1047–1054, 2000.

[100] M. Locatelli. A note on the Griewank test function. *Journal of Global Optimization*, 25 (2):169–174, 2003.

[101] S. Lohr. *Sampling: Design and Analysis*. Advanced Series. Cengage Learning, 2009.

[102] D. G. Luenberger and Y. Ye. *Linear and nonlinear programming*, volume 116. Springer, 2008.

[103] M. Lunacek and D. Whitley. The dispersion metric and the CMA evolution strategy. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'06)*, pages 477–484, New York, USA, 2006. ACM.

[104] D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[105] C. Macnish. Towards unbiased benchmarking of evolutionary and hybrid algorithms for real-valued optimisation. *Connection Science*, 19(4):361–385, 2007.

[106] W. Macready and D. Wolpert. What makes an optimization problem hard. *Complexity*, 5:40–46, 1996.

[107] K. Malan and A. Engelbrecht. Ruggedness, funnels and gradients in fitness landscapes and the effect on PSO performance. In *IEEE Congress on Evolutionary Computation (CEC'13)*, pages 963–970, 2013.

[108] K. Malan and A. Engelbrecht. A progressive random walk algorithm for sampling continuous fitness landscapes. In *IEEE Congress on Evolutionary Computation (CEC'14)*, pages 2507–2514, July 2014.

[109] K. M. Malan and A. P. Engelbrecht. Quantifying ruggedness of continuous landscapes using entropy. In *IEEE Congress on Evolutionary Computation (CEC'09)*, pages 1440–1447, 2009.

[110] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(1):239–245, 1979.

[111] M. Meilă. Comparing clusterings by the variation of information. In *Learning theory and kernel machines*, pages 173–187. Springer, 2003.

[112] O. Mersmann, M. Preuss, and H. Trautmann. Benchmarking evolutionary algorithms: Towards exploratory landscape analysis. In *Parallel Problem Solving from Nature (PPSN XI)*, volume 6238 of *Lecture Notes in Computer Science*, pages 73–82. Springer, 2010.

[113] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph. Exploratory landscape analysis. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'11)*, pages 829–836, New York, USA, 2011. ACM.

[114] P. Merz. Advanced fitness landscape analysis and the performance of memetic algorithms. *Evolutionary Computation*, 12(3):303–325, 2004.

[115] P. Merz and B. Freisleben. Memetic algorithms for the traveling salesman problem. *Complex Systems*, 13(4):297–346, 2001.

[116] J. V. Michalowicz, J. M. Nichols, and F. Bucholtz. *Handbook of Differential Entropy*. Taylor & Francis, 2013.

[117] R. Morgan and M. Gallagher. Using landscape topology to compare continuous metaheuristics: A framework and case study on EDAs and ridge structure. *Evolutionary computation*, 20(2):277–299, 2012.

[118] M. Muñoz, M. Kirley, and S. Halgamuge. Exploratory landscape analysis of continuous space optimization problems using information content. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2014.

[119] M. A. Muñoz, M. Kirley, and S. Halgamuge. A meta-learning prediction model of algorithm performance for continuous optimization problems. In *Parallel Problem Solving from Nature (PPSN XII)*, volume 7491 of *Lecture Notes in Computer Science*, pages 226–235. Springer, 2012.

[120] M. A. Muñoz, M. Kirley, and S. K. Halgamuge. Landscape characterization of numerical optimization problems using biased scattered data. In *IEEE Congress on Evolutionary Computation (CEC'12)*, pages 1180 –1187, 2012.

[121] C. Müller, B. Baumgartner, and I. Sbalzarini. Particle swarm CMA evolution strategy for the optimization of multi-funnel landscapes. In *IEEE Congress on Evolutionary Computation (CEC'09)*, pages 2685–2692, 2009.

[122] C. L. Müller and I. F. Sbalzarini. Global characterization of the CEC 2005 fitness landscapes using fitness-distance analysis. In *Applications of Evolutionary Computation*, volume 6624 of *Lecture Notes in Computer Science*, pages 294–303. Springer, 2011.

[123] B. Naudts and L. Kallel. A comparison of predictive measures of problem difficulty in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 4(1):1–15, 2000.

[124] G. Ochoa, R. Qu, and E. K. Burke. Analyzing the landscape of a graph based hyperheuristic for timetabling problems. In *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO'09)*, pages 341–348. ACM, 2009.

[125] M. Overton. *Numerical Computing with IEEE Floating Point Arithmetic*. Cambridge University Press, 2001.

[126] R. Palmer. Optimization on rugged landscapes. In *Molecular Evolution on Rugged Landscapes*, pages 3–25. Addison-Wesley Publishing Company, 1991.

[127] I. Pavlov. 7-Zip, 2014. URL `http://www.7-zip.org/`.

[128] S. Picek and D. Jakobovic. From fitness landscape to crossover operator choice. In *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO'14)*, pages 815–822. ACM, 2014.

[129] M. Pigliucci. Adaptive landscapes, phenotypic space, and the power of metaphors. *The Quarterly Review of Biology*, 83(3):283–287, 2008.

[130] E. Pitzer and M. Affenzeller. A comprehensive survey on fitness landscape analysis. In *Recent Advances in Intelligent Engineering Systems*, volume 378 of *Studies in Computational Intelligence*, pages 161–191. Springer, 2012.

[131] E. Pitzer and M. Affenzeller. Measurement of anisotropy in fitness landscapes. In *Computer Aided Systems Theory (EUROCAST'13)*, pages 340–347. Springer, 2013.

[132] W. Provine. *Sewall Wright and Evolutionary Biology*. University of Chicago Press, 1989.

[133] T. Rapcsák. Smooth nonlinear nonconvex optimization. In C. Floudas and P. Pardalos, editors, *Encyclopedia of Optimization*, number 1, pages 3622–3625. Springer, 2008.

[134] P. Ratanaworabhan, J. Ke, and M. Burtscher. Fast lossless compression of scientific floating-point data. In *Proceedings of Data Compression Conference (DCC'06)*, pages 133–142, 2006.

[135] V. C. Raykar, R. Duraiswami, and L. H. Zhao. Fast computation of kernel estimators. *Journal of Computational and Graphical Statistics*, 19(1):205–220, 2010.

[136] C. Reeves. Landscapes, operators and heuristic search. *Annals of Operations Research*, 86:473–490, 1999.

[137] C. R. Reeves. Fitness landscapes. In E. K. Burke and G. Kendall, editors, *Search Methodologies*, pages 681–705. Springer, 2014.

[138] C. R. Reeves and J. E. Rowe. *Genetic Algorithms - Principles and Perspectives: A Guide to GA Theory*. Operations Research/Computer Science Interfaces Series. Springer, 2002.

[139] C. Reidys and P. Stadler. Combinatorial landscapes. *SIAM Review*, 44(1):3–54, 2002.

[140] C. M. Reidys and P. F. Stadler. Neutrality in fitness landscapes. *Applied Mathematics and Computation*, 117(2-3):321 – 350, 2001.

[141] E. Ridge and D. Kudenko. An analysis of problem difficulty for a class of optimisation heuristics. In *Proceedings of the European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP'07)*, pages 198–209, Berlin, 2007. Springer.

[142] K. H. Riitters, R. O'Neill, C. Hunsaker, J. D. Wickham, D. Yankee, S. Timmins, K. Jones, and B. Jackson. A factor analysis of landscape pattern and structure metrics. *Landscape ecology*, 10(1):23–39, 1995.

[143] A. H. G. Rinnooy Kan and G. T. Timmer. Chapter IX global optimization. In *Optimization*, volume 1 of *Handbooks in Operations Research and Management Science*, pages 631 – 662. Elsevier, 1989.

[144] S. Rochet. Epistasis in Genetic Algorithms revisited. *Information Sciences*, 102(1-4):133 – 155, 1997.

[145] H. Rosé, W. Ebeling, and T. Asselmeyer. The density of states - a measure of the difficulty of optimisation problems. In *Parallel Problem Solving from Nature (PPSN IV)*, pages 208–217, London, UK, 1996. Springer.

[146] K. H. Rosen. *Handbook of Discrete and Combinatorial Mathematics*. Discrete Mathematics and Its Applications. Taylor & Francis, 1999.

[147] R. W. Russell, G. L. Hunt Jr, K. O. Coyle, and R. T. Cooney. Foraging in a fractal environment: spatial patterns in a marine predator-prey system. *Landscape Ecology*, 7 (3):195–209, 1992.

[148] D. Salomon, G. Motta, and D. Bryant. *Handbook of data compression*. Springer, 2010.

[149] D. W. Scott. *Nonparametric Estimation Criteria*, pages 33–45. John Wiley & Sons, 2008.

[150] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 2009.

[151] D. Seo and B. Moon. An information-theoretic analysis on the interactions of variables in combinatorial optimization problems. *Evolutionary Computation*, 15(2):169–198, 2007.

[152] Y. Sergeyev and D. Kvasov. Lipschitz global optimization. *Wiley Encyclopedia of Operations Research and Management Science*, 4:2812–2828, 2010.

[153] S. Shan and G. G. Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2):219–241, 2010.

[154] S. Sheather and M. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 683–690, 1991.

[155] S. J. Sheather. Density estimation. *Statistical Science*, 19(4):588–597, 2004.

[156] M. F. Shlesinger, B. J. West, and J. Klafter. Lévy dynamics of enhanced diffusion: Application to turbulence. *Physical Review Letters*, 58:1100–1103, 1987.

[157] D. Simon. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

[158] T. W. Simpson, D. K. J. Lin, and W. Chen. Sampling strategies for computer experiments: design and analysis. *International Journal of Reliability and Applications*, 2(3): 209–240, 2001.

[159] J. Slaney and T. Walsh. Backbones in optimization and approximation. In *International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 254–259. Morgan Kaufmann, 2001.

[160] T. Smith, P. Husbands, and M. O'Shea. Not measuring evolvability: Initial investigation of an evolutionary robotics search space. In *IEEE Congress on Evolutionary Computation (CEC'01)*, volume 1, pages 9–16. IEEE, 2001.

[161] T. Smith, P. Husbands, and M. O'Shea. Fitness landscapes and evolvability. *Evolutionary computation*, 10(1):1–34, 2002.

[162] K. Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, 41(1):1 – 25, 2008.

[163] K. Smith-Miles and L. Lopes. Measuring instance difficulty for combinatorial optimization problems. *Computers and Operations Research*, 39(5):875–889, 2011.

[164] K. Smith-Miles and T. T. Tan. Measuring algorithm footprints in instance space. In *IEEE Congress on Evolutionary Computation (CEC'12)*, pages 1–8. IEEE, 2012.

[165] J. Smykla, J. Wołek, and A. Barcikowski. Zonation of vegetation related to penguin rookeries on king george island, maritime antarctic. *Arctic, Antarctic, and Alpine Research*, 39(1):143–151, 2007.

[166] R. K. Som. *Practical sampling techniques*. CRC press, 1995.

[167] G. B. Sorkin. Efficient Simulated Annealing on fractal energy landscapes. *Algorithmica*, 6:367–418, 1991.

[168] E. Specht. Packomania. http://www.packomania.com, 2012.

[169] P. F. Stadler. Towards a theory of landscapes. In *Complex systems and binary networks*, pages 78–163. Springer, 1995.

[170] P. F. Stadler. Landscapes and their correlation functions. *Journal of Mathematical Chemistry*, 20:1–45, 1996.

[171] P. F. Stadler. Fitness landscapes. In *Biological Evolution and Statistical Physics*, volume 585 of *Lecture Notes in Physics*, pages 183–204. Springer, 2002.

[172] P. F. Stadler and W. Grüner. Anisotropy in fitness landscapes. *Journal of Theoretical Biology*, 165(3):373, 1993.

[173] P. F. Stadler and W. Schnabl. The landscape of the traveling salesman problem. *Physics Letters A*, 161(4):337 – 344, 1992.

[174] P. F. Stadler, W. Hordijk, and J. F. Fontanari. Phase transition and landscape statistics of the number partitioning problem. *Physical Review E*, 67(5):056701, 2003.

[175] K. C. B. Steer, A. Wirth, and S. K. Halgamuge. Information theoretic classification of problems for metaheuristics. In *Simulated Evolution and Learning*, volume 5361 of *Lecture Notes in Computer Science*, pages 319–328. Springer, 2008.

[176] J. Stewart. *Calculus*. Cengage Learning, 2007.

[177] M. J. Streeter. Two broad classes of functions for which a no free lunch result does not hold. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'03)*, volume 2724 of *Lecture Notes in Computer Science*, pages 1418–1430. Springer, 2003.

[178] R. Strongin. On the convergence of an algorithm for finding a global extremum. *Engineering Cybernetics*, 11:549–555, 1973.

[179] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, KanGAL Report, 2005.

[180] P. G. Szabó, M. C. Markót, and T. Csendes. Global optimization in geometry - circle packing into the square. In P. Audet, P. Hansen, and P. Savard, editors, *Essays and Surveys in Global Optimization*. Kluwer, 2005.

[181] E. Talbi. *Metaheuristics: from design to implementation*. Wiley Series on Parallel and Distributed Computing. John Wiley & Sons, 2009.

[182] J. Theiler. Estimating fractal dimension. *Journal of the Optical Society of America A*, 7(6): 1055–1073, 1990.

[183] L. Tian. Fitness landscape analysis for capacitated vehicle routing problem. In *Proceedings of the International Conference on Cybernetics and Informatics*, volume 163 of *Lecture Notes in Electrical Engineering*, pages 119–125. Springer, 2014.

[184] M. Tomassini, L. Vanneschi, P. Collard, and M. Clergue. A study of fitness distance correlation as a difficulty measure in genetic programming. *Evolutionary Computation*, 13(2):213–239, 2005.

[185] M. G. Turner. *Landscape ecology in theory and practice: pattern and process*. Springer, 2001.

[186] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

[187] J. van Hemert. Property analysis of symmetric travelling salesman problem instances acquired through evolution. *Evolutionary Computation in Combinatorial Optimization*, pages 122–131, 2005.

[188] L. Vanneschi, M. Tomassini, P. Collard, and S. Vérel. Negative slope coefficient: A measure to characterize genetic programming fitness landscapes. In *Genetic Programming*, volume 3905 of *Lecture Notes in Computer Science*, pages 178–189. Springer, 2006.

[189] L. Vanneschi, D. Codecasa, and G. Mauri. An empirical comparison of parallel and distributed particle swarm optimization methods. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'10)*, pages 15–22, New York, USA, 2010. ACM.

[190] V. K. Vassilev, T. C. Fogarty, and J. F. Miller. Information characteristics and the structure of landscapes. *Evolutionary Computation*, 8:31–60, 2000.

[191] G. Venter and J. Sobieszczanski-Sobieski. Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization. *Structural and Multidisciplinary Optimization*, 26(1-2):121–131, 2004.

[192] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 9999:2837–2854, 2010.

[193] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. E. Da Luz, E. P. Raposo, and H. E. Stanley. Optimizing the success of random searches. *Nature*, 401(6756):911–914, 1999.

[194] P. M. Vitányi, F. J. Balbach, R. L. Cilibrasi, and M. Li. Normalized information distance. In *Information theory and statistical learning*, pages 45–82. Springer, 2009.

[195] M. Wagner, J. Day, and F. Neumann. A fast and effective local search algorithm for optimizing the placement of wind turbines. *Renewable Energy*, 51:64–70, 2013.

[196] S. Wagner and D. Wagner. *Comparing clusterings: an overview*. Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007.

[197] D. Wales. *Energy Landscapes: Applications to Clusters, Biomolecules and Glasses*. Cambridge Molecular Science. Cambridge University Press, 2003.

[198] Y. Wang and B. Li. Understand behavior and performance of real coded optimization algorithms via NK-linkage model. In *IEEE Congress on Evolutionary Computation (CEC'08)*, pages 801 –808, 2008.

[199] E. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63:325–336, 1990.

[200] E. Weinberger and P. Stadler. Why some fitness landscapes are fractal. *Journal of Theoretical Biology*, 163(2):255–275, 1993.

[201] D. Whitley and J. P. Watson. Complexity theory and the no free lunch theorem. In *Search Methodologies*, pages 317–339. Springer, 2005.

[202] D. Whitley, M. Lunacek, and A. Sokolov. Comparing the niches of CMA-ES, CHC and pattern search using diverse benchmarks. *Parallel Problem Solving from Nature (PPSN IX)*, pages 988–997, 2006.

[203] D. Whitley, A. M. Sutton, and A. E. Howe. Understanding elementary landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'08)*, pages 585–592. ACM, 2008.

[204] G. R. Wood and B. P. Zhang. Estimation of the Lipschitz constant of a function. *Journal of Global Optimization*, 8:91–103, 1996.

[205] S. Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution. In *Proceedings of the International Congress of Genetics*, volume 1, pages 356–366, 1932.

[206] W. Zhang. Phase transitions and backbones of the asymmetric traveling salesman problem. *Journal of Artificial Intelligence Research*, 21(1):471–497, 2004.

[207] W. Zhang and R. E. Korf. A study of complexity transitions on the asymmetric traveling salesman problem. *Artificial Intelligence*, 81(1-2):223 – 239, 1996.

Appendices

APPENDIX A

Test Problems

This appendix describes the optimization problems analysed in this thesis. The problems include various continuous artificial problems, problems from the Black-Box Optimization Benchmarking (BBOB) set, Circle in a Square (CiaS) packing problems, Travelling Salesman Problems (TSP) and Number Partitioning Problems (NPP).

A.1 Continuous Problems

A.1.1 Artificial Problems

There are many problems with artificial and contrived structures that have been defined in the optimization literature [181]. Often, such problems are purposefully constructed with a particular structural feature in mind, and then subsequently used to experimentally investigate algorithm behaviour. The problems defined in Table A.1 were carefully selected from the optimization literature, and contain a variety of structural features that have been shown to affect the behaviour of many algorithms (see Simon [157] for a survey of popular benchmark problems).

Sphere

The Sphere function is essentially a quadratic, separable, multi-dimensional bowl. Hence it is symmetric and convex. The gradient changes smoothly as solutions approach the global optimum, which is located at $\mathbf{x}^* = [0]^D$. Any algorithm that can efficiently solve a convex problem (e.g. Broyden-Fletcher-Goldfarb-Shanno quasi-Newton) will likely perform well on the Sphere function.

Name	**Definition**	**Bounds**
2-D Ellipse	$f(\mathbf{x}) = x_1^2 + a{x_2}^2$	$\mathbf{x} \in [-1, 1]^2, a \in \mathbb{R}$
Griewank	$f_D(\mathbf{x}) = 1 + \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} cos\left(\frac{x_i}{\sqrt{i}}\right)$	$\mathbf{x} \in [-600, 600]^D$
Michalewicz	$f_D(\mathbf{x}) = -\sum_{i=1}^{D} \sin x_i \sin^m \frac{ix_i^2}{\pi}$	$\mathbf{x} \in [0, \pi]^D$
Rastrigin	$f_D(\mathbf{x}) = AD + \sum_{i=1}^{D}\left(x_i^2 - A\cos\left(2\pi x_i\right)\right)$	$\mathbf{x} \in [-5.12, 5.12]^D$
Rosenbrock	$f_D(\mathbf{x}) = \sum_{i=1}^{D-1}(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2$	$\mathbf{x} \in [-5, 10]^D$
Sphere	$f_D(\mathbf{x}) = \sum_{i=1}^{D} x_i^2$	$\mathbf{x} \in [-1, 1]^D$

Table A.1: Artificial Objective Functions

2-D Ellipse

The 2-D elliptical function is very similar to the 2-D Sphere function, however it has elliptical contours (as opposed to circular). The 2-D elliptical function is convex, unimodal and has a single global optimum located at $\mathbf{x}^* = [0]^D$. The parameter a controls the eccentricity of the elliptical function; at $a = 1$, the ellipse is perfectly circular, and as a increases, the contours of the function narrow and the eccentricity becomes more pronounced. Elliptical functions are frequently used to test the sensitivity of algorithms to elliptical contours (ranking-based algorithms are typically invariant).

Rastrigin

The Rastrigin function (see Table A.1) is a non-convex, highly multimodal function. It is parameterised by a perturbation term, A, that controls the magnitudes of oscillations in f, and so A can be used to adjust the degree of ruggedness in problem instances. At $A = 0$, the problem is a smooth, convex function that is equivalent to the Sphere function, and as A increases, the oscillations (and hence modes) in the function become more pronounced. Ignoring the perturbations caused by the oscillations, the Rastrigin function is globally convex. The global optimum is located at $\mathbf{x}^* = [0]^D$.

Rosenbrock

The Rosenbrock function (see Table A.1) is a non-convex, unimodal, smooth optimization problem with a global minimum at $\mathbf{x} = [1]^D$. The global optimum is located in a narrow, largely neutral parabolic-shaped valley. Rosenbrock is often deemed difficult, due to the neutrality of the valley that algorithms must navigate in order to find the global optimum.

Griewank

The Griewank function can be broken down into two major components; a convex structure defined by $1 + \frac{1}{4000}\sum_{i=1}^{D} x_i^2$, and an oscillating, non-convex structure defined by $\prod_{i=1}^{D} cos\left(\frac{x_i}{\sqrt{i}}\right)$. While the number of local optima increases exponentially with dimensionality, the non-convex component shrinks in volume, and hence its contribution to the overall structure of the function becomes increasingly less relevant [100]. Consequently, an increasing majority of the Griewank function is convex as dimensionality increases, which in turn can make the function "easier"; any algorithm that can efficiently solve a convex problem (e.g. Broyden-Fletcher-Goldfarb-Shanno quasi-Newton) will likely perform well. The global optimum is located at $\mathbf{x}^* = [0]^D$.

Michalewicz

The Michalewicz function contains $D!$ axis-aligned neutral valleys that span across $\mathcal{S}$. The global optimum occurs at the intersection of the valleys, and its location is dependant on the dimensionality. The steepness of the valleys are defined by m, and increases as m increases.

A.1.2 Black-Box Optimization Benchmarking Set

The Black-Box Optimization Benchmarking (BBOB) problem set [72] consists of a variety of continuous artificial benchmark functions. The problems are scalable with dimension and defined over $\mathbb{R}^D$, although the specific search space used in algorithm performance competitions is $\mathcal{S} = [-5, 5]^D$. The problems are randomly translated in both the search space and f, meaning different *instances* can be produced by supplying a different seed to the benchmark generator. The BBOB problems are classified into one of five classes, based on the expert intuition of the developers [72]. The five classes are:

1. Separable
2. Low or moderate conditioning
3. Unimodal and high conditioning
4. Multimodal with adequate global structure
5. Multimodal with weak global structure

Table A.2 describes known properties of the BBOB problems, including whether problems are unimodal or multimodal, and the degree of conditioning. Further details can be found in [72].

A.1.3 Circle Packing in a Square

Circle in a Square (CiaS) packing problems are a class of well-studied geometric packing problems. Given the unit square defined in a 2D Euclidean space and a pre-specified number of circles, n_c, constrained to be of equal size, the problem is to find an optimal packing; i.e. to position the circles and compute the radius length of the circles such that the circles occupy the maximum possible area within the square. All circles must remain fully enclosed within the square, and cannot overlap.

Mathematically, the problem can be stated as follows [2]. Let $C(\mathbf{z}^i, v)$ be the circle with radius v and center $\mathbf{z}^i = (y_1^i, y_2^i) \in \mathbb{R}^2$. Then the optimization problem is:

$$\begin{aligned} v_n &= \max v && \text{(A.1.1)} \\ C(\mathbf{z}^i, v) &\subseteq [0,1]^2, i = 1, \ldots, n_c && \text{(A.1.2)} \\ C^{int}(\mathbf{z}^i, v) \cap C^{int}(\mathbf{z}^j, v) &= \emptyset \; \forall \; i \neq j && \text{(A.1.3)} \end{aligned}$$

where C^{int} is the interior of a circle.

Alternatively, the problem can be reformulated as finding the positions of n_c points inside the unit square such that their minimum pairwise distance is maximized. In this case the problem (and constraint) can be restated as:

$$\begin{aligned} d_n &= \max \min_{i \neq j} \parallel \mathbf{w}^i - \mathbf{w}^j \parallel_2 && \text{(A.1.4)} \\ \mathbf{w}^i &\in [0,1]^2, i = 1, \ldots, n_c && \text{(A.1.5)} \end{aligned}$$

It is known that a solution to (A.1.4) can be transformed into a solution to (A.1.1) using the following relation:

$$v_n = \frac{d_n}{2(d_n + 1)}.$$

From the point of view of evaluating metaheuristic optimization algorithms, the prob-

ID	Name	Class	Modality	Ill-Conditioning	Other
1	Sphere	1	Unimodal	Low	Symmetric, convex
2	Ellipsoidal	1	Unimodal	High	Separable
3	Rastrigin	1	Multimodal	Low	Regular
4	Büche-Rastrigin	1	Multimodal	Low	Asymmetric, deceptive
5	Linear Slope	1	Unimodal	Low	Purely linear
6	Attractive Sector	2	Unimodal	Low	Highly asymmetric
7	Step Ellipsoidal	2	Unimodal	Low	Highly neutral
8	Rosenbrock (original)	2	Multimodal	Low	Narrow valley, deceptive
9	Rosenbrock (rotated)	2	Multimodal	Low	Narrow valley, deceptive
10	Ellipsoidal	3	Unimodal	High	Non-separable
11	Discus	3	Unimodal	High	Globally convex
12	Bent Cigar	3	Unimodal	High	Narrow valley
13	Sharp Ridge	3	Unimodal	High	Narrow valley
14	Divergent Powers	3	Unimodal	High	Small global basin
15	Rastrigin	4	Multimodal	Low	Irregular
16	Weierstrass	4	Multimodal	High	Regular, multiple $\mathbf{x}^*$
17	Schaffers F7	4	Multimodal	Low	Asymmetric
18	Schaffers F7 (ill-conditioned)	4	Multimodal	Moderate	Asymmetric
19	Griewank-Rosenbrock F8F2	4	Multimodal	Low	Narrow valley, deceptive
20	Schwefel	5	Multimodal	Low	Partially separable
21	Gallagher's 101-me	5	Multimodal	Low	Irregular
22	Gallagher's 21-hi	5	Multimodal	Moderate	Irregular
23	Katsuura	5	Multimodal	Low	Regular
24	Lunacek bi-Rastrigin	5	Multimodal	Low	Two funnels, deceptive

Table A.2: Black-Box Optimization Benchmarking Problem Set

lem given by Equation A.1.4 is convenient because generating a feasible candidate solution simply requires placing a set of n_c points within the unit square. Note that the optimization problem is over $2n_c$ continuous variables (the coordinates of each point $\mathbf{w}^i$ in the unit square). A candidate solution is then a vector of the circle coordinates, i.e. $\mathbf{x} = \left[w_1^1, w_2^1, \ldots, w_1^n, w_2^n\right]$. Hence in this thesis, the objective function for CiaS problems is:

$$f_n(\mathbf{x}) = -d_n \tag{A.1.6}$$

using d_n from Equation A.1.4.

CiaS packing problems can be considered as a simplified version of a number of different real-world problems and have received a large amount of attention in the mathematical, optimization and operations research literature (see Castillo et al. [31] for an overview). For most values of n_c below 60 and for certain other values, provably optimal packings have been found using either theoretical or computational approaches (see [180] and the references therein). For larger values of n_c, finding provably optimal packings in general becomes increasingly difficult and time-consuming. The Packomania website [168] maintains a large list of the optimal (or best known) packings for many values of n_c from 2 up to 10000, along with references and other related resources.

A.2 Combinatorial Problems

A.2.1 Travelling Salesman Problem

The Travelling Salesman Problem is a well-studied NP-hard combinatorial optimization problem where the objective is to find a tour through n cities, such that each city is visited exactly once and the total distance of the tour is minimised. The TSP can be defined using a directed graph, $G = (V, D)$, where the vertex set $V = \{1, \ldots, n\}$ represents the cities, and the distance matrix $D = (d_{i,j})$ specifies the distance from city i to j. Let $\mathbf{x} = (x_{1,1}, \ldots, x_{n,1}, \ldots x_{n,n})$ be a solution vector where

$$x_{i,j} = \begin{cases} 1 & \text{if } i \text{ is connected to } j \\ 0 & \text{otherwise} \end{cases} \tag{A.2.1}$$

Then, the TSP can be formulated as:

$$\min \sum_{i \neq j} D_{i,j} x_{i,j} \tag{A.2.2}$$

where

$$\sum_{i=1, i \neq j}^{n} x_{i,j} = 1 \qquad (j \in V, i \neq j) \tag{A.2.3}$$

$$\sum_{j=1, j \neq i}^{n} x_{i,j} = 1 \qquad (i \in V, j \neq i) \tag{A.2.4}$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \qquad (S \subset V, 2 \leq |S| \leq n-2) \tag{A.2.5}$$

A related problem is the NP-complete *decision* TSP, where the objective is to decide if, given a distance k, there exists a tour through the n cities (visiting each city exactly once) that is shorter than k [48, 76].

A notion of distance between TSP solutions is based on the number of common edges between the solutions, which can be calculated via the Hamming distance between the solution/tour matrices.

$$\text{shared}(\mathbf{x}^a, \mathbf{x}^b) = \sum_{i \neq j} x^a_{i,j} \wedge x^b_{i,j} \tag{A.2.6}$$

If city i is connected to city j in a *symmetric* TSP solution, then $x_{i,j} = 1$ and $x_{j,i} = 1$. Consequently, the number of shared edges calculated above in Equation A.2.6 will be twice what is actually shared.

Hence given two *symmetric* TSP solutions, $\mathbf{x}^a$ and $\mathbf{x}^b$, the distance between solutions is defined as:

$$\text{dist}_{\text{TSP}}(\mathbf{x}^a, \mathbf{x}^b) = 1 - \frac{shared(\mathbf{x}^a, \mathbf{x}^b)}{2n} \tag{A.2.7}$$

While the distance between two *asymmetric* TSP solutions is:

$$\text{dist}_{\text{ATSP}}(\mathbf{x}^a, \mathbf{x}^b) = 1 - \frac{shared(\mathbf{x}^a, \mathbf{x}^b)}{n} \tag{A.2.8}$$

TSPLib

TSPLib is a collection of hundreds of TSP (and related problem) instances assembled from a variety of real-world and artificial sources. TSPLib has been widely used in the TSP community for benchmarking algorithm performance and assessing the discriminatory power of problem features [164]. The library includes instances with a wide variety of cities and distance metrics, including L_1, L_2 and L_∞ norms in 2D and 3D as well as geographical distance (i.e. distance measured along the surface of the earth). The subset of instances used in this dissertation are summarised in Table A.3.

Phase Transitions in Asymmetric Travelling Salesman Problems

Both the decision and optimization versions of the TSP have been shown to exhibit phase transition behaviour [34, 206]. For the optimization TSP, Zhang [206] showed that the tour distance and size of backbones, as well as the performance of a well-known branch and bound algorithm, have two characteristically different values, and that the transition between these values is very abrupt. Problems were constructed by generating distances uniform randomly from integers in the range $[0, \ldots, \lfloor 10^b \rfloor]$, where $b > 0$. The parameter b essentially controls the diversity of distances in D, and is normalised by $\log_{10}(n)$ to allow comparisons between problems of varying n. Phase transitions are typically exhibited as $\frac{b}{\log_{10}(n)}$ is varied at minor increments from 0 to 6.5.

A.2.2 Number Partitioning Problem

The number partitioning problem is a classic NP-hard combinatorial problem [58]. Given a multiset of positive integers, S, the objective is to partition or separate S into two disjoint subsets, S_1, S_2, such that the sums of each set are as close as possible. This can be formulated as an optimization problem in the following way. Let $S = \{s_1, s_2, \ldots, s_n\}$, where s_i are drawn randomly (according to some distribution, e.g. Uniform) from $\{1, 2, \ldots, m\}$. Let S_1, S_2 be two disjoint subsets of S, i.e. $S_1, S_2 \subset S$ such that $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \emptyset$. Let $\mathbf{x} = 0, 1^n$ represent a candidate solution where if position i is set, then $s_i \in S_1$ (and hence if i is not set, then $s_i \in S_2$). Then, the objective is to minimise the *discrepancy* of the set:

$$f(\mathbf{x}) = \max \left\{ \sum_i^n s_i x_i, \sum_i^n s_i (1 - x_i) \right\} \tag{A.2.9}$$

The optimal discrepancy is 0 for even n, and 1 for odd. Because $\mathbf{x} \in \{0, 1\}^n$, and there are

Name	**Type**	**Number of Cities (n)**	**Distance Type**
kroA100	Symmetric	100	2-D L_2
kroB100	Symmetric	100	2-D L_2
kroC100	Symmetric	100	2-D L_2
kroD100	Symmetric	100	2-D L_2
kroE100	Symmetric	100	2-D L_2
kro124p	Asymmetric	100	2-D L_2
bayg29	Symmetric	29	Geographical
bays29	Symmetric	29	Geographical
gr17	Symmetric	17	Explicit
gr21	Symmetric	21	Explicit
gr24	Symmetric	24	Explicit
gr48	Symmetric	48	Explicit
ulysses16	Symmetric	16	Geographical
ulysses22	Symmetric	22	Geographical
br17	Asymmetric	17	Explicit
ft53	Asymmetric	53	Explicit
ft70	Asymmetric	70	Explicit
ftv33	Asymmetric	33	Explicit
ftv35	Asymmetric	35	Explicit
ftv38	Asymmetric	38	Explicit
ftv44	Asymmetric	44	Explicit
ftv47	Asymmetric	47	Explicit
ftv55	Asymmetric	55	Explicit
ftv64	Asymmetric	64	Explicit
p43	Asymmetric	43	Explicit
ry48p	Asymmetric	48	Explicit

Table A.3: TSPLib Instances

a total of 2^n possible candidate solutions. However, the NPP contains a natural symmetry between candidate solutions and their compliments. That is, given a candidate solution $\mathbf{x}$, $f(\mathbf{x}) = f(\bar{\mathbf{x}})$. Hence while there are 2^n possible solutions, there are only 2^{n-1} *unique* solutions due to the symmetry.

The NPP is known to undergo a phase transition in a number of landscape properties as well as difficulty for heuristic algorithms [17, 22, 61, 174]. The stage of the phase transition can be controlled using the parameter $k = \frac{1}{n}\log_2 m$. There is a single critical stage at which the problem structure and difficulty changes drastically. For $k < k_c$, there are numerous global optima, few plateaus and problems are generally known as "easy". At $k > k_c$, problems have few global optima, many plateaus and are generally seen as difficult. Borgs et al. [22] derive k_c by the following:

$$k_c = 1 - \frac{\ln\left(\frac{\pi}{6}n\right)}{2n\ln(2)} \tag{A.2.10}$$

APPENDIX B

Definitions and Formulae for Combinatorial Problems

Length Scale Distribution

Definition B.1. *Let r be a discrete random variable taking values from the set $R \subset \mathbb{R}$ (i.e. $r \in R$). The **length scale distribution** is defined as the probability mass function $p(r)$.*

Note $\sum_{r \in R} p(r) = 1$.

Entropy

$$H(r) = -\sum_{r \in R} p(r) \log_2 p(r) \tag{B.0.1}$$

where the convention $0 \log_2 0 = 0$ is used.

KL Divergence

$$D_{KL}(p||q) = \sum_{r \in R} p(r) \, \log_2 \frac{p(r)}{q(r)} \tag{B.0.2}$$

where $0 \log_2 0 = 0$ and $q(i) = 0$ implies $p(i) = 0$.

APPENDIX C

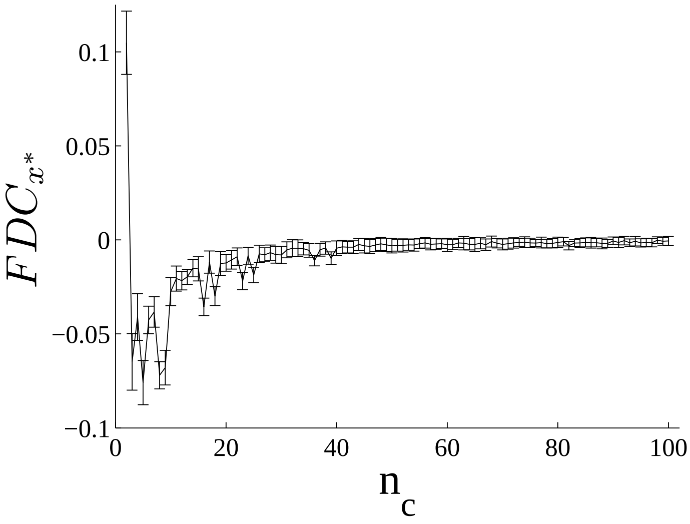
Length Scale Analysis of Circle in a Square Problems

The following results are based on the experiments in Section 7.3 of Chapter 6, where a Lévy random walk is the basis for estimating several landscape analysis features in Circle in a Square (CiaS) packing problems. The results in this appendix are for a *uniform random sample*, as opposed to the Lévy random walk. The motivation behind using a uniform random walk is twofold; firstly, to examine how the features perform with such a sample, and secondly, to relate the results to intuition regarding uniform packings. In particular, the objective function value assigned to a solution is the minimum distance between any two circle centres (multiplied by -1, as this analysis assumes minimisation). As n_c increases, the radius of the circles decreases, and so for a random solution, the minimum distance between any two circle centres is expected to also decrease. Hence, as n_c increases, the objective function value of uniform random packings should decrease.

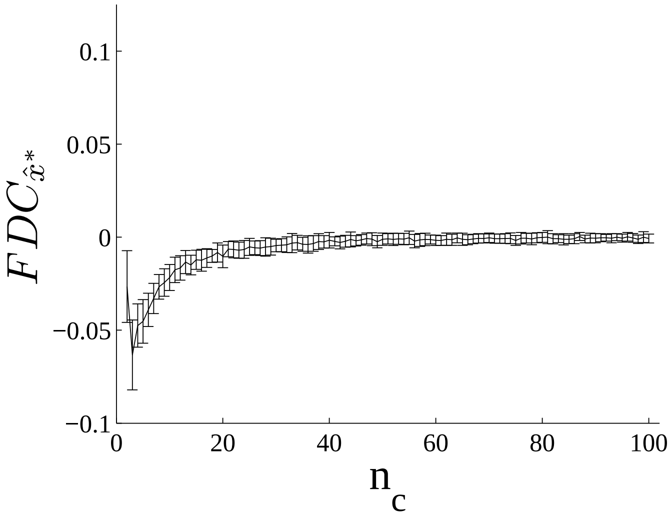
C.1 Results Comparing Length Scale to Existing Features

C.1.1 Fitness Distance Correlation

In general, both estimators of FDC (shown in Figure C.1a and C.1b) have relatively small standard deviation (errorbars) over trials, which decreases as the number of circles increases. FDC values are typically small and negative for small n_c, and as n_c increases, values increases towards 0. An exception of this trend occurs at the transition from $n_c = 2$ to $n_c = 3$, where the FDC values decrease noticeably; $FDC_{\mathbf{x}^*}$ transitions from 0.1048 to -0.0649, while $FDC_{\hat{\mathbf{x}}^*}$ transitions from -0.0265 to -0.0633. In the case of $FDC_{\hat{\mathbf{x}}^*}$, values then steadily increase towards 0 as n_c increases. On the other hand, $FDC_{\mathbf{x}^*}$ shows fluctuations in FDC as n_c ini-



(a) $FDC_{\mathbf{x}^*}$



(b) $FDC_{\hat{\mathbf{x}}^*}$

Figure C.1: FDC for CiaS problems. Lines show the mean of the 30 trials, while error bars indicate one standard deviation.

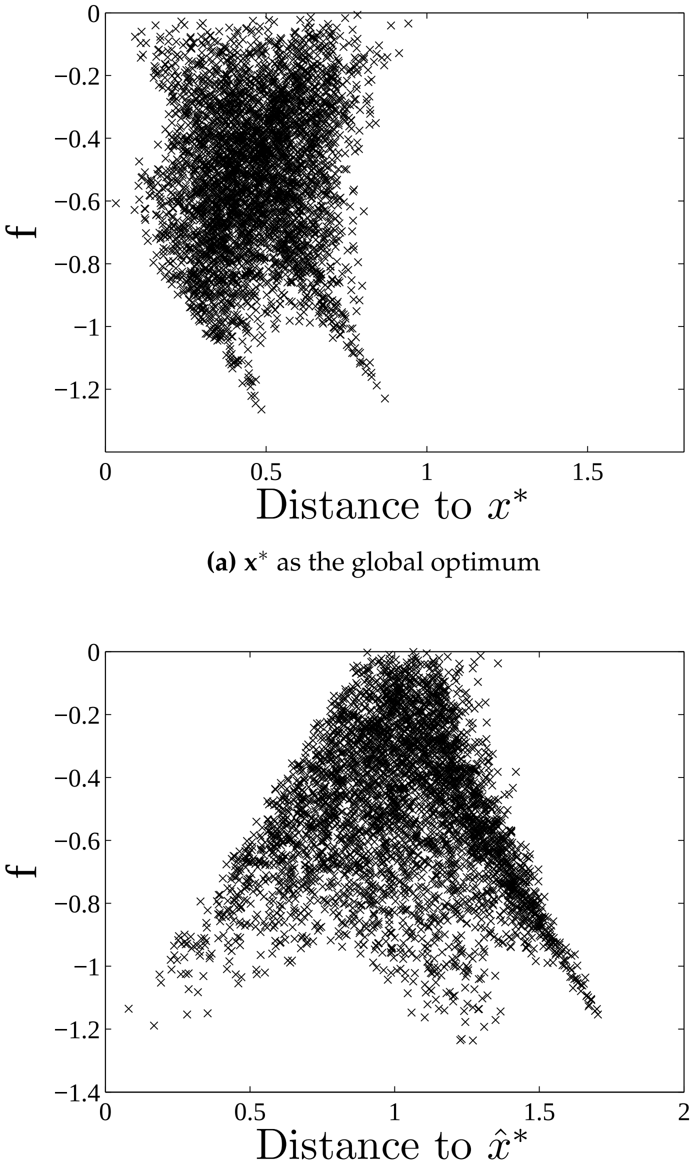
tially increases, and from approximately $n_c = 40$ the values steadily increase towards 0 as n_c increases. The fluctuations generally correlate with problems where symmetrical global solutions exist for at least one of the problems (e.g. the transition from $n_c = 9$ to $n_c = 10$). Furthermore, for $n_c \geq 40$ packings (where the FDC values are rather stable) the majority of problems (i.e. approximately 85%) have assymetrical global solutions.

Figure C.1a and C.1b generally indicate that for low numbers of circles (i.e. $n_c < 20$), the f-values of random solutions is slightly negatively correlated with their distance to the global optimum, however, as the number of circles increases, the f-values of random solutions has essentially no correlation with their distance to the global optimum. A negative value of FDC in the context of a minimization problem indicates that in general, the objective function values of the sampled solutions gets smaller as the distance from their closest global optimum increases. Such a circumstance can be caused by many factors (and their interactions), including the presence of many local optima and multiple global optima, which CiaS problems are known to have (as discussed in Section 7.3 of Chapter 6). The FDC values alone give no further insight into such factors, nor do they adequately differentiate between problems of varying n_c (particularly for $n_c > 40$).

The landscape at $n_c = 2$ has a positive $FDC_{\mathbf{x}^*}$ value and yet a negative $FDC_{\hat{\mathbf{x}}^*}$ value. Compressing the complex interaction between fitness and distance (to the global optimum) to a correlation coefficient may obviously lose important structural information, and so fitness-distance scatter plots can be used to visualise and better understand FDC and the landscape structure. The fitness-distance scatter plots for $FDC_{\mathbf{x}^*}$ and $FDC_{\hat{\mathbf{x}}^*}$ at $n_c = 2$ are shown in Figure C.2a and C.2b respectively.

Figure C.2a shows a general lack of correlation between f-values and distance to $\mathbf{x}^*$, however there are a few subsets of solutions that show correlation in their distribution. In particular, there is one area of strong positive correlation (i.e. solutions in Figure C.2a where $f(\mathbf{x}) > -0.8$), indicating that the objective function values increase as the distance from $\mathbf{x}^*$ increases. There are also two areas of weak negative correlations (i.e. solutions where $f(\mathbf{x}) < -1$), where the objective function values decrease as the distance from $\mathbf{x}^*$ increases. The positive correlation is much stronger than the two weaker correlations, and thus overall there is a positive $FDC_{\mathbf{x}^*}$ value (albeit a small one). Figure C.2b shows quite different structure compared to Figure C.2a, in particular, there are much larger distances between solutions and $\hat{\mathbf{x}}^*$. The overall shape and trends in the data are also substantially different. Figure C.2b shows little evidence of the weak negative correlations that are present in Figure C.2a. While there is perhaps a small subset of solutions with positive correlation (i.e. solutions in Figure C.2b where the distance to $\hat{\mathbf{x}}^*$ is less than 1), there is a prominent subset of solutions with a negative correlation (i.e. solutions where the distance to $\hat{\mathbf{x}}^*$ is greater than 1), thus explaining why $FDC_{\hat{\mathbf{x}}^*}$ is negative (albeit small).

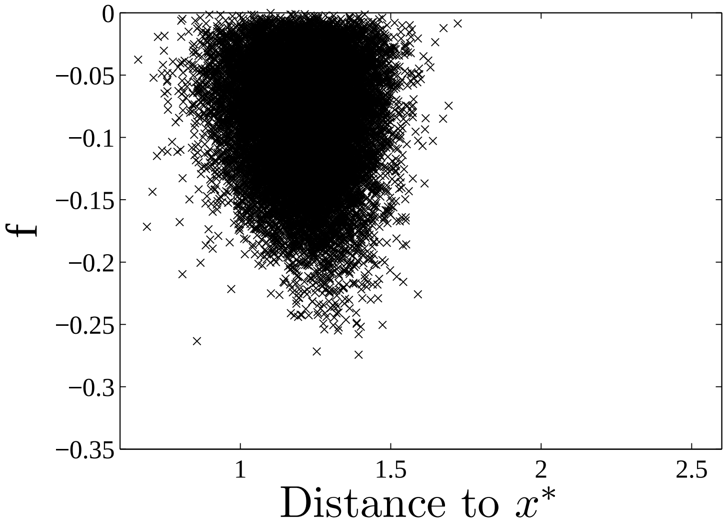
In general, the interesting structure shown in Figure C.2a and C.2b was not evident when looking at scatterplots for $n_c \geq 5$. For example, the fitness-distance scatter plots for $n_c = 9$ and $n_c = 10$ are examined and shown in Figure C.3a and C.3b respectively. While there are no obvious trends in either Figure C.3a or C.3b, the overall shape of the data distribution is different. Solutions in Figure C.3a have a smaller distance to $\mathbf{x}^*$ as well as a smaller range in the distances to $\mathbf{x}^*$ compared to solutions in Figure C.3b. This is not surprising given that



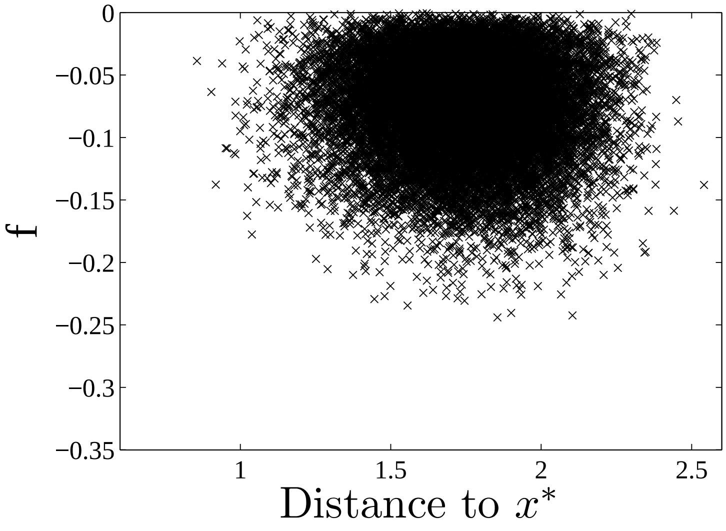
(a) $\mathbf{x}^*$ as the global optimum

(b) $\hat{\mathbf{x}}^*$ as the global optimum

Figure C.2: Typical fitness-distance scatter plots for $n_c = 2$ circles.



(a) $n_c = 9$



(b) $n_c = 10$

Figure C.3: Typical fitness-distance scatter plots using $\mathbf{x}^*$ as the global optimum.

global solutions for $n_c = 9$ have 8 symmetries, compared with 1 for $n_c = 10$, and so it is expected that solutions with a greater number of global optimum to compare to will have both smaller distances and a smaller range of distances. Thus, while there are no obvious differences in the trends in Figure C.3a and C.3b, the difference between the number of global optima affects the value and range of distances obtained, which in turn likely causes fluctuations between $FDC_{\mathbf{x}^*}$ values.

C.1.2 Dispersion

The bound-normalised dispersion values for the CiaS problems are shown in Figure C.4 and reveals a relatively large decrease in dispersion from 2 circles to 10 circles. A decrease

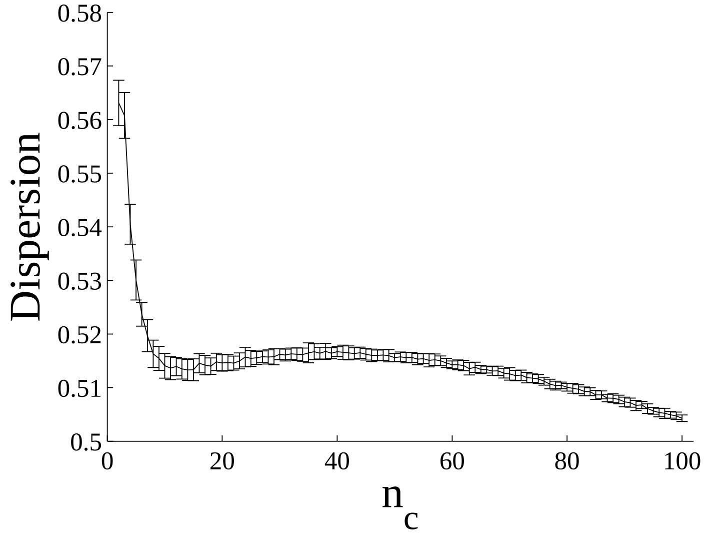
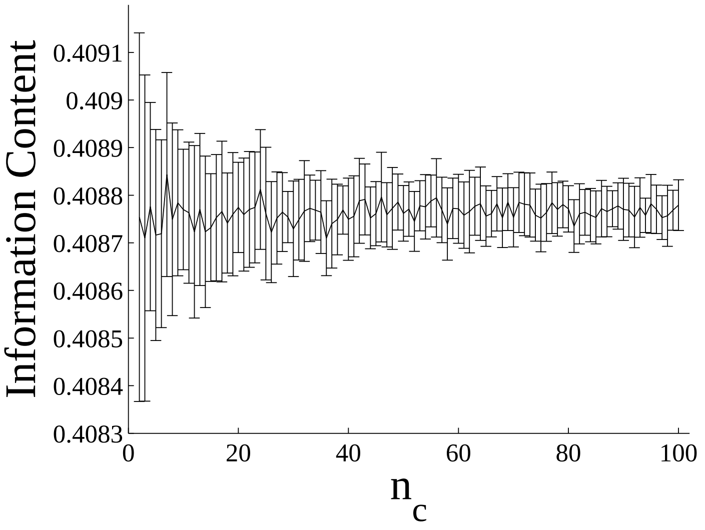


Figure C.4: Bound-Normalised dispersion for CiaS problems as n_c increases.

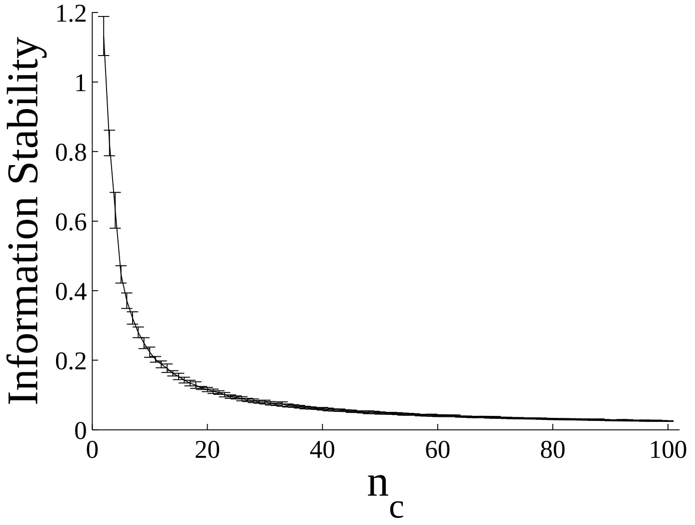
in dispersion indicates that high quality solutions are increasingly closer together. Hence, Figure C.4 indicates that high quality solutions for $n_c = 10$ are closer together in $\mathcal{S}$ than high quality solutions for $n_c = 2$. Following the initial decrease in dispersion values, from $n_c = 10$ to $n_c = 40$, the dispersion increases slightly. The slight increase of dispersion is very interesting, however analysis of the distance distributions of the fittest 5% of solutions for $n_c = 20, 40$ and 80 did not give an obvious explanation into the increase in dispersion. Figure C.4 also shows that for $n_c > 40$, dispersion is slightly decreasing, meaning that for $n_c > 40$, high quality solutions are slightly closer together as n_c increases. Overall, the bound-normalised dispersion has small variability between samples, however because the dispersion values are very similar and non-unique across problems, it would seem to be of limited use differentiating and characterising the CiaS packing problems.

C.1.3 Information Content, Partial Information Content and Information Stability

The information content and partial information content features were found to be highly correlated for the CiaS problems (the sample correlation coefficient is 0.999). Consequently, only the information content is shown in Figure C.5a. The value of information content is roughly constant over all of the CiaS problems, with small fluctuations as indicated by the scale on the information content axis in Figure C.5a. Comparisons with the information content and partial information content of highly rugged landscapes in [190] suggest that the values (and fluctuations) obtained are reasonable. Similar to other features, the standard



(a) Information Content.



(b) Information Stability.

Figure C.5: Information-Theoretic Analysis for CiaS problems as n_c increases.

deviation decreases as n_c increases. The information content indicates that the problems do not significantly change in ruggedness. Most importantly, however, the technique is clearly unable to differentiate and characterise CiaS problems.

In contrast to information content, the information stability feature, shown in Figure C.5b, exhibits a strong, smooth trend as n_c increases and has very small standard deviation between trials. In particular, as n_c increases, the information stability is a monotonically decreasing function approaching 0. This is determined by the nature of objective function values in CiaS problems and the evaluation of solutions. Information stability is simply the largest change in objective function value between two steps in the walk. Because the magnitude of objective function values (for random solutions) are generally decreasing as n_c in-

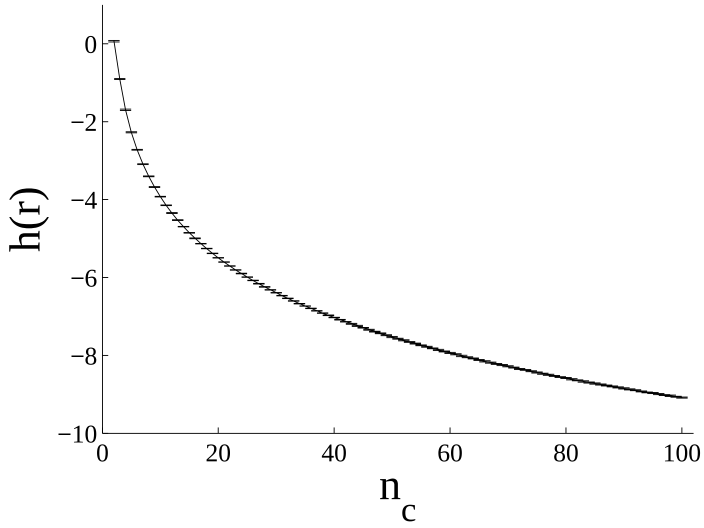


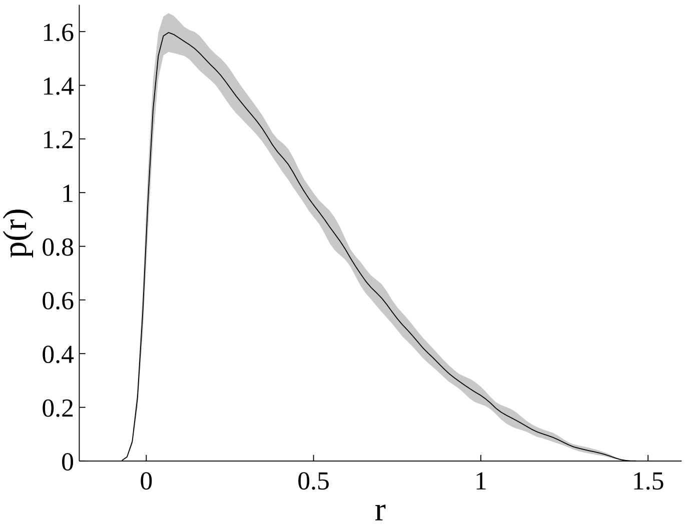
Figure C.6: $h(r)$ for CiaS problems as n_c increases.

creases, it is no surprise that the information stability is also decreasing as n_c increases. Thus, while information stability is a robust and unique characteristic the problem for changing n_c, it is likely due to the decreasing nature of f, rather than changes in landscape structure. Furthermore, analysis of individual information stability values does not give much insight into landscape structure. For example, at $n_c = 2$, the average information stability over the 30 trials is approximately 1.13. This merely indicates that the largest change in f-values (between a step in the walk) is 1.13; no information regarding other changes in f-values, the distribution of objective function values or the interaction of solutions and objective function values is captured.

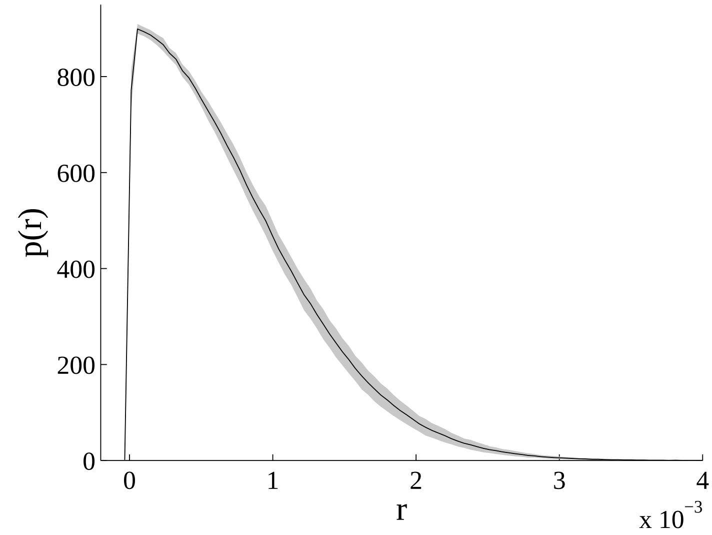
C.2 Interpreting Length Scale Distributions of the Circle in a Square Problems

The entropy of the length scale distribution, $h(r)$, is shown in Figure C.6. It clearly characterises and discriminates problems of different n_c. In addition, $h(r)$ has very low standard deviation across the repeated samples, which suggests that for these problems, it is a highly robust landscape feature.

The decrease in $h(r)$ (as n_c increases) indicates that the diversity of the changes in objective function between two random solutions is decreasing. The length scale distributions in Figures C.7a and C.7b confirm this; Figure C.7a has a heavier tail than C.7b. In particular, for $n_c = 100$, $p(r)$ favours "small" length scales compared to $p(r)$ for $n_c = 2$. Further insight



(a) $n_c = 2$



(b) $n_c = 100$

Figure C.7: The change in shape of $p(r)$ as n_c increases.

can be gained by comparing these distributions. Examining this more closely, the ratio of mode and 99th percentile shown in Figure C.8. As n_c increases, both the mode and 99th percentile of $p(r)$ are decreasing, with the 99th percentile decreasing at a faster rate.

The range of length scales is also very different between $n_c = 2$ and $n_c = 100$, which is evidence of the decrease of the magnitude of objective function values for random solutions as n_c increases (as already discussed in the context of information stability). However, the analysis of $p(r)$ and $h(r)$ provides compelling evidence that the decrease is complex and non-uniform across solutions. If it were a uniform decrease across solutions, the length scale values would merely be scaled by a factor and there would be no change in the shape of $p(r)$. Hence, length scale analysis has uncovered two valuable insights into the nature

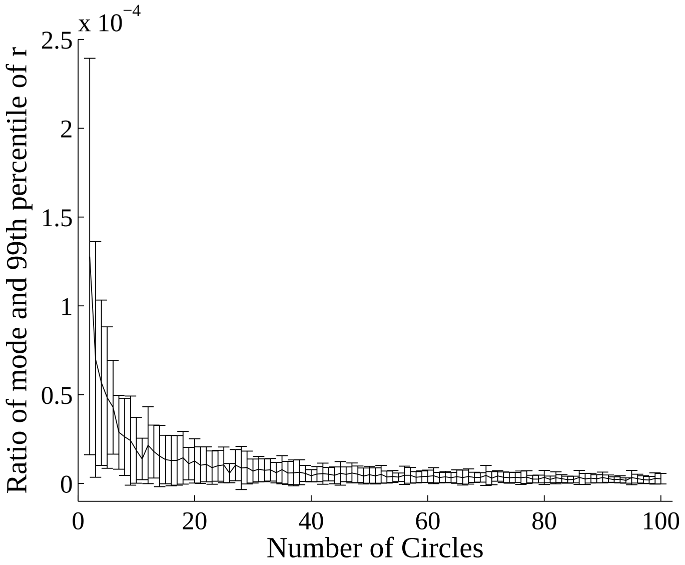


Figure C.8: Ratio of the mode and 99th percentile of r, confirming the significant change in the shape of $p(r)$ as n_c increases.

of CiaS packing problems; as the number of circles packed increases, it is expected that 1) the packing of a random configuration gets better and that 2) moving from one random configuration to another will produce increasingly less significant changes in quality. While both these insights are known in the circle packing literature, the length scale analysis is able to uncover such insights using purely black-box information. Furthermore, the second insight is extremely useful and not identifiable from other existing landscape analysis techniques. This knowledge could potentially be used to help algorithm practitioners design more effective restart strategies for this problem, as well as aid in the judgement of algorithm convergence.