UNIVERSITY OF STIRLING

DOCTORAL THESIS

Anatomy of the Local Optima Level in Combinatorial Optimisation

Author:
Sarah L. THOMSON

Supervisor:
Professor Gabriela OCHOA

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Division of Computing Science and Mathematics

August 28th, 2020

Declaration of Authorship

I, Sarah L. THOMSON, declare that this thesis titled, "Anatomy of the Local Optima Level in Combinatorial Optimisation" and the work presented in it are my own. I confirm that:

- This work was done wholly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Sarah Thomson

Date: August 28th, 2020

"Keep your Eyes on the Stars, and your Feet on the Ground."
Theodore Roosevelt

UNIVERSITY OF STIRLING

Abstract

School of Natural Sciences
Division of Computing Science and Mathematics

Doctor of Philosophy

Anatomy of the Local Optima Level in Combinatorial Optimisation

by Sarah L. THOMSON

Many situations in daily life represent complex combinatorial optimisation problems. These include issues such as efficient fuel consumption, nurse scheduling, or distribution of humanitarian aid. There are many algorithms that attempt to solve these problems but the ability to understand their likely performance on a given problem is still lacking.

Fitness landscape analysis identifies some of the reasons why metaheuristic algorithms behave in a particular way. The Local Optima Network (LON) model, proposed in 2008, encodes local optima connectivity in fitness landscapes. In this approach, nodes are local optima and edges encode transitions between these optima. A LON provides a static image of the dynamics of algorithm-problem interplay. Analysing these structures provides insights into the reactions between optimisation problems and metaheuristic search algorithms.

This thesis proposes that analysis of the local optima space of combinatorial fitness landscapes encoded using a LON provides important information concerning potential search algorithm performance. It considers the question as to whether or not features of LONs can contribute to explaining or predicting the outcome of trying to optimise an associated combinatorial problem. Topological landscape features of LONs are proposed, analysed and compared. Benchmark and novel problem instances are studied; both types of problem are sampled and in some cases exhaustively-enumerated such that LONs can be extracted for analysis. Investigations into the nature and biases of LON construction algorithms are conducted and compared.

Contributions include aligning fractal geometry to the study of LONs; proposals for novel ways to compute fractal dimension from these structures; comparing the power of different LON construction algorithms for explaining algorithm performances; and analysing the interplay between algorithmic operations and infeasible regions in the local optima space using LONs as a tool. Throughout the thesis, large scale structural patterns in fitness landscapes are shown to be strongly linked with metaheuristic algorithm performance. This includes arrangements of local optima funnel structures; spatial and geometric complexity in the LON (measured by their fractal dimensionality) and fitness levels in the space of local optima. These features are demonstrated to have explanatory or predictive ability with respect to algorithm performance for the underlying combinatorial problems. The results presented here indicate that large topological patterns in fitness landscapes are important during metaheuristic search algorithm design. In many cases they are incontrovertibly linked to the success of the algorithm. These results indicate that use of the suggested fitness landscape measures would be highly beneficial when considering the design of search algorithms for a given problem domain.

Acknowledgements

I have been incredibly blessed to have such extraordinary people around me throughout my PhD. The following is in no particular order and will not be overly eloquent; all my cognitive strength has already been spent while writing this.

My endless and earnest thanks to my two supervisors: Professor Gabriela Ochoa and Dr David Cairns. Their consistent academic, technical, and pastoral advice and support has been invaluable and I will always be grateful.

To my unofficial second supervisors of the past, Dr Fabio Daolio and Dr Nadarajen Veerapen, who helped me become a more competent and more well-rounded researcher with everything they taught me. Thank you.

My sincere thanks to Dr Sébastien Verel, for his innovative ideas, support, and engaging intellectual conversation.

Ken, Jason, Paul, Saemi, and Mila — the PhD students, present and former, who I was lucky enough to cross paths with: for their incredible guidance, support, and friendship — I can never express my thanks and appreciation through words.

Mum, dad, and brother. For being endlessly compassionate and genuinely caring. Thank you.

My best friends: Anna, Amie, Jenny, Lexy, Tom, Luke, Andrew, Ryan, Rebecca. I could never have done this without their beautiful and timeless friendship.

Brutus and Tank — they always have been, and always will be, the goodest boys.

Contents

List of Figures

List of Tables

List of Abbreviations

LON	**L**ocal **O**ptima **N**etwork
QAP	**Q**uadratic **A**ssignment **P**roblem
ILS	**I**terated **L**ocal **S**earch
TS	**T**abu **S**earch
SA	**S**imuated **A**nnealing
GA	**G**enetic **A**lgorithm
MA	**M**emetic **A**lgorithm
FD	**F**ractal **D**imension
CSOP	**C**hemotherapy **S**chedule **O**ptimisation **P**roblem

List of Symbols

λ	length of distance used in the autocorrelation function
ϵ	local optima fitness distance maximum (box counting fractal analysis)
β	minimum probability of search transition between local optima
δ	fixed effects from predictors on the response variable
α	random effects on the response variable arising from instance class differences
ζ	residuals in mixed-effects regression models
σ, θ	parameters pertaining to tumour growth
η	known harmful effect on an organ from administering drugs

Chapter 1

Introduction

1.1 Overview

Systems which can be formulated as optimisation problems are ubiquitous. Examples include the scheduling of nurses; routing of delivery vehicles; distribution of humanitarian aid; identifying locations for food trucks which can reach as many homeless people as possible. There are microcosmic cases too: personal time management, for example, or video game strategy. The necessity of understanding and conducting optimisation mindfully is self-evident.

A prevalent method aimed at intelligent optimisation is *fitness landscape* analysis [1]. Fitness landscapes model the interplay between optimisation problems and search algorithms; topological features arise during the interactions and these can explain or predict the performance of optimisation algorithms. Indeed, charting fitness landscapes has unveiled intense relationships with algorithm performance in the past [2, 3, 4, 5, 6, 7, 8, 9]. Such findings can serve as a springboard for better-informed algorithm design, which can in turn produce improved optimisation [10].

Local Optima Networks (LONs) [11] are a fitness landscape paradigm which are gaining traction in the research community [12, 13, 14, 15, 16]. LONs encode local optima connectivity based on historical searches through the fitness landscape and they provide a snapshot of algorithm dynamics at the local optima level.

LONs have been used with some success for understanding, explaining, or predicting search algorithm performance [17, 18, 19, 20]; still, the parallel between the anatomy of the local optima layer in the landscape and algorithm performance remains mostly veiled. There is a lack of certainty and insight concerning macroscopic landscape phenomena which can be studied using LONs. This thesis pursues clarity about some of them.

A related element missing from the literature concerns LON construction algorithms; in fact, there are no dedicated works analysing them. Early LON construction algorithms exhaustively enumerated the fitness landscape [8, 11, 21, 22] and built complete LONs. This was initially satisfactory for proof-of-concept scenarios on small problems; that said, research is directed towards larger problems now. Consequently, proficient sampling algorithms for approximate LONs are essential if the explanatory and predictive potency of LONs is to be maximised. Some sampling algorithms have been proposed which are augmented with domain-competitive optimisation algorithms [19, 23, 24, 25] but these have not been critically analysed or compared yet. This thesis addresses that with the aim of enhancing reliability and consistency in LON analysis and — by consequence — in algorithm prediction.

Specific contributions of this work include aligning fractal geometry to the study of LONs; proposals for novel ways to compute fractal dimension from these structures; comparing the power of different LON construction algorithms for explaining

algorithm performances; and analysing the interplay between algorithmic operations and infeasible regions in the local optima space using LONs as a tool. The universal thread binding together these contributions is articulated in the hypothesis below.

Hypothesis. *Valuable information is encoded within Local Optima Networks about reactions between metaheuristic algorithms and combinatorial optimisation problems; this can be used for visualising, explaining, or predicting the proficiency of those algorithms.*

The stipulated hypothesis will be evaluated in Section 9.2 at the conclusion of this thesis. Associated research questions include:

1. Which new knowledge can be gained from applying fractal geometry to LONs?
2. How can we define fractal dimension for LONs?
3. Can we learn about the relationships between LON features and algorithm performance by comparing LON construction methods?
4. How can LONs help better understand algorithm performance on highly-constrained problems?

The rest of the document is structured in the following way: Chapter 2 describes preliminaries which are necessary in understanding the literature review and subsequent contributions; Chapter 3 provides a location for the thesis contributions within the wider field, with descriptions and critical analysis of related literature; contributions are distributed into five subsequent Chapters, 4 - 8; these are followed by Chapter 9, which concludes and directs future work.

1.2 Publications Produced

Publications produced during the course of this PhD are now mentioned. These are in chronological order, starting with the oldest. For each, a statement about the authorship of the work is given. In the interest of clarity, absolutely no writing by other authors, and absolutely no experiments or results obtained by other authors are included in this thesis.

The publications detailed in Sections 1.2.1 and 1.2.2 are not part of this thesis. In the case of the former, this is because there was a collaborative aspect to the work; for the publication in Section 1.2.2, the contribution was comparatively narrow and does not align with the main story and logical sequence of work presented in this thesis.

1.2.1 Comparing Communities of Optima with Funnels in Combinatorial Fitness Landscapes (*GECCO 2017; ECOM track*) — S. L. Thomson, F. Daolio, and G. Ochoa.

As stipulated, this paper is not included in the thesis (due to the collaborative nature). In the publication, I contributed the research described in Sections 5.1 and 5.3 and am the author of approximately 70% of the writing in this paper — specifically, all sections except for 4.2 and 5.2.

1.2.2 The Effect of Landscape Funnels in QAPLIB Instances (*GECCO 2017 Companion; Landscape-aware Heuristic Search Workshop*) — S. L. Thomson, G. Ochoa, F. Daolio, and N. Veerapen.

Although not included in the thesis, I note that all writing, experiments and results were conducted by the author of this thesis, other than proof reads by others.

1.2.3 On the Fractal Nature of Local Optima Networks (*EvoCOP 2018*) — S. L. Thomson, S. Verel, G. Ochoa, N. Veerapen, and P. McMenemy.

All writing, experiments and results were conducted by the author of this thesis, other than proof reads by others. This paper forms the basis of Chapter 4, although the statistical analysis methods are different for this thesis. The network data are available at https://github.com/sarahlouisethomson/fractal-nature-local-optima-networks.

1.2.4 Multifractality and Dimensional Determinism in Local Optima Networks (*GECCO 2018; ECOM track*) — S. L. Thomson, S. Verel, G. Ochoa, N. Veerapen, and D. Cairns.

All writing, experiments and results were conducted by the author of this thesis, other than proof reads by others. Work from this publication is included in Chapter 5. The network data are available at https://github.com/sarahlouisethomson/multifractality-dimensional-determinism-lons.

1.2.5 Clarifying the Differences in Local Optima Network Sampling Algorithms (*EvoCOP 2019*) — S. L. Thomson, G. Ochoa, and S. Verel.

All writing, experiments and results were conducted by the author of this thesis, other than proof reads by others. A section of this work contributes to Chapter 7 alongside the related publication detailed in Section 1.2.7. The network data are available at https://github.com/sarahlouisethomson/clarifying-differences-local-optima-networks.

1.2.6 The Local Optima Level in Chemotherapy Schedule Optimisation (*EvoCOP 2020*) — S. L. Thomson and G. Ochoa

All writing, experiments and results were conducted by the author of this thesis, other than proof reads by others. The work is the foundation for Chapter 9. The network data are available at https://github.com/sarahlouisethomson/chemotherapy-optimisation-local-optima-networks.

1.2.7 Inferring Future Landscapes: Sampling the Local Optima Level (*Evolutionary Computation Journal 2020*) — S. L. Thomson, G. Ochoa, S. Verel, and N. Veerapen

All writing, experiments and results were conducted by the author of this thesis, other than proof reads by others. The publication forms part of Chapter 7 alongside the related publication described in Section 1.2.5. The network data are available at https://github.com/sarahlouisethomson/inferring-future-landscapes-sampling-local-optima-networks.

1.2.8 The Fractal Geometry of Fitness Landscapes at the Local Optima Level (*Natural Computing Journal 2020*) — S. L. Thomson, G. Ochoa, and S. Verel

The writing, experiments, and results of this paper were conducted by the author of this thesis (other than proof reads by others). This work forms the basis for Chapter 6. The network data are available at https://github.com/sarahlouisethomson/fractal-geometry-fitness-landscapes. Fractal analysis algorithms for local optima networks are provided at https://github.com/sarahlouisethomson/compute-fractal-dimension-local-optima-networks.

Chapter 2

Background & Preliminaries

In the incumbent Chapter, notions and axioms which are foundational to the contributions in this work are introduced and described. These lay the groundwork for clarity of understanding and self-containment later on. I begin by detailing the meaning of combinatorial optimisation itself, along with problems which are used in the contribution Chapters. Next, evolutionary computation and affiliated algorithms are elucidated, followed by the definition of fitness landscapes. After that is the Local Optima Network model, which is a fitness landscape analysis tool. Finally, the notions of fractal geometry and fractal dimension are explained.

2.1 Combinatorial Optimisation

A combinatorial optimisation problem [26] concerns the pursuit of an optimal design for an object when considering a finite number of possible designs. Each design — or *solution* — has an associated quality, or *fitness*. Formally speaking, such a problem has four components: $[I, S, f, o]$ with I being an instance set; S the set of solutions which are feasible; f the quality function (fitness function) to grade solutions; and o is the objective, whether that is minimisation or maximisation. The solution *representation* is the choice of how to encode a design for computational optimisation. Typical representations include binary strings and permutations of integers.

In this thesis I exclusively study single-objective optimisation. This stipulates that there is a single objective function involved — in contrast to multi-objective optimisation, where there is more than one objective function [27].

2.2 Benchmark Problems

The following Section illustrates benchmark combinatorial optimisation problems which later serve as test-beds for analysis in the contributions. I report the solution representation, objective, fitness function, and how instances are encoded.

2.2.1 NK Landscapes

NK Landscapes [28] are a family of synthetic fitness functions. They give rise to fitness landscapes which can be tuned from completely smooth to completely rugged. There are two parameters: N and K. Solutions are binary-encoded and of length N. Each bit has a numeric value assigned from a uniform distribution of floating-point numbers which represents the contribution of the bit towards the overall solution fitness. The parameter K dictates the extent of *epistasis*, i.e. how many other bit values are considered when calculating the fitness contribution of a given bit. The overall

fitness f of a given solution s is the average of the fitness contributions of the N bits:

$$f(s) = \sum^{i} f(s_i) \tag{2.1}$$

where s_i is the bit at position i and $f(s_i)$ is its individual fitness contribution. When calculating this, the values of K other bits are also considered:

$$f(s_i) = f(s_i, s_1^i, ..., s_k^i) \tag{2.2}$$

NK Landscapes are often used as a test-bed for new fitness landscape techniques (see for example [4, 11, 29, 30]) because ruggedness can be introduced in a controlled way using the parameter K.

2.2.2 Quadratic Assignment Problem

The Quadratic Assignment Problem (QAP) [31] is often used in fitness landscape analysis [5, 30, 32, 33, 34]. A QAP instance is specified with a distance matrix and a flow matrix. An entry in the distance matrix, D_{ij} is the distance between two locations i and j. In the flow matrix this is the flow between two items: F_{ij}. Solutions are encoded as a permutation of length N, and are the allocation of N items to N locations. Fitness of a solution is the product of distances and flows between the locations and items according to the permutation and the objective is minimisation. The fitness function f for a solution s is then $f(s) = \sum_{i=1}^{N} \sum_{j=1}^{N} D_{ij}F_{ij}$, $\forall s \in SP$, where SP is the solution space. Occurrences of the QAP can largely be placed into one of four categories or instance classes: uniform random distances and flows; random flows on grids; real-world problems; and random "real-world like" problems. Each of these is now introduced.

Uniform random distances and flows

This class of QAP is known to be challenging for metaheuristic algorithms. The entries for both the distance and flow matrices are taken at random from a Gaussian distribution. The naming convention is *tainna*, where *nn* is the problem dimension.

Random flows on grids

For these problems, the locations are each situated in one square on an *m*n* grid, which is rectangular. The flow matrix entries are generated randomly.

Real-world problems

These problems arise from practical applications. Examples include stenographer typing patterns; the planning of a hospital layout; and the testing of sequential circuits.

Random "real-world like" problems

Instances from this class resemble real-world manifestations of the QAP because the distance matrix reflects locations which are not uniformly distributed on the plane, but rather occur in clusters. The naming convention is *tainnb*, where *nn* is the problem dimension.

2.3 Evolutionary Computation & Metaheuristics

An approach to generating solutions to combinatorial optimisation problems is with metaheuristics and evolutionary computation. A metaheuristic is a blueprint for an algorithm which iterates through solutions, changing them and making decisions about how to move to the next solution. Metaheuristics typically combine a localised improvement process (exploitation or intensification) with a more exploratory one (exploration or diversification) [35]. The specific mechanisms and parameter choices are made by the algorithm designer for the problem at hand, but the algorithm conforms to the metaheuristic template. Evolutionary computation [36] concerns metaheuristics which borrow design principles from natural evolution, such as recombination and selection. Metaheuristics which are used in the contribution Chapters of this thesis are now described according to the Handbook of Metaheuristics [35].

Iterated Local Search

An Iterated Local Search (ILS) is a single-point algorithm, meaning that a sole solution is being improved throughout the process. ILS consists of hill-climbing (localised improvement) and perturbation, a strong exploratory change to the solution. Hill-climbing consists of making small changes (mutations) to the solution and then deciding whether to keep the newly-changed variant. This decision is governed by a *pivot rule* which is typically either first-improvement or best-improvement. First improvement dictates that as soon as a solution with superior fitness is found, that replaces the current solution. Best improvement iterates through all one-mutation neighbours of the incumbent solution and picks the one with the most desirable fitness. With every ILS iteration (that is, a perturbation followed by hill-climbing) a local optimum is reached. The algorithm must include an acceptance condition concerning whether to allow this solution to become the new base for the search. In many cases the acceptance rule is the deterministic acceptance of any improving local optimum (with respect to the previous local optimum).

Tabu Search

Tabu Search (TS) marries hill-climbing (local improvement) with a mechanism for escaping local optima: allowing deteriorating fitness moves with guidance from the "tabu" tail. The tail consists of search mutations which are forbidden as choices for the algorithm at that time (they are "tabu"). Tail elements are typically recently-recorded mutations and are intended to prevent looping. The length of the tail is decided by the algorithm designer.

Simulated Annealing

Simulated Annealing (SA) combines the exploitative nature of hill-climbing with an exploratory solution acceptance schedule inspired by the method of cooling metals in a controlled way to minimise imperfections. At the beginning of the simulated annealing search, the probability that a deteriorating solution will be accepted is high. That probability gradually decreases throughout the search process, ending with the acceptance of mostly (or only) improving solutions; the motivation underlying this is that the preliminary stages require exploration to assess where the promising regions are, whereas during the latter stages the algorithm should be converging to a good solution (i.e. exploiting known desirable genes or solution components).

Genetic Algorithm

A Genetic Algorithm (GA) is a population-based metaheuristic and evolutionary algorithm, where a set of individuals evolve with each iteration through a combination of selection, recombination, and mutation. Individuals in the population is replaced by newly-generated offspring. GAs do not have "acceptance conditions" as such, but they can have mechanisms such as *elitism*, where the best individual(s) are forced to remain in the population, and *tournament selection* — selection of parents based on fitness ranking. The performance of the algorithm can be markedly altered through design choices such as the size of the population, recombination and mutation rates, type of recombination, method of tournament selection, and number of generations.

Memetic Algorithm

A Memetic Algorithm (MA) combines a local search or improvement with an additional mechanism; for example, an evolutionary technique (such as a genetic algorithm) with a local improvement procedure on some (or all) individuals in the population of solutions.

2.4 Landscape Systems

Many complex systems which are studied in science fit naturally into the template of a *landscape*. The elegantly-uncomplicated metaphor of the *fitness landscape* was first considered in 1932 by geneticist Sewall Wright [37] for visualising evolutionary trajectories taken by populations; evolutionary biologists have been using the term since [38, 39, 40, 41]. Similarly, physicists observe *energy landscapes*, modelling physical configurations of a molecular structure and their associated energies — these are the 'heights' in the landscape. The model can describe the observable states into which proteins fold, but have broader applicability too [42, 43, 44, 45, 46].

How about combinatorial optimisation and evolutionary computation? The notion of landscapes was brought to combinatorial optimisation in a seminal work by Kauffman [28] which unified the landscape view with well-known optimisation problems such as the Travelling Salesman Problem. In evolutionary computation Weinberger [29] sought to transfigure the abstract *landscape* metaphor into an empirical mathematical object. He viewed it as a graph: $G = (V, E)$, where a vertex $s \in S$ is tagged with its real-valued fitness $f(s)$ and an edge $e \in E$ exists from s_1 to s_2 if s_1 can be transformed into s_2 through application of a search operation. The graphs do not consider the probability of transformation, nor the case where an operation produces more than one configuration (an example would be a recombination operator producing two offspring).

Computer scientist Terry Jones pursued a rigorous definition for fitness landscapes in the context of evolutionary algorithms [47]. He asserted that using elemental distance functions in calculating a landscape (a random one-flip hypercube in binary problems, for example) might not produce topological features which mirror the features generated during optimisation with multi-operator algorithms. His answer to this was captured in a new model: that the landscape is characterised by a graph: $G = (V, E)$ and a vertex $v \in V$ is this time a *multi-set* of configurations. Those are assigned a mapping, $f(v)$, from configuration to fitness. An edge $e \in E$ is traced between multi-sets (v_1, v_2) if the probability of v_1 transforming to v_2 after

using a search operation op is greater than zero, and e is labelled with that probability. This landscape paradigm is congruent with operations which produce more than one solution because multi-sets form the vertices.

While admiring the flexibility of this approach, in this thesis I consider instead the more widely-used assertion by Stadler [1] that a fitness landscape has three elemental parts: $[S, \mathcal{N}, f]$, where S is the set of all possible configurations; $\mathcal{N} : S \longrightarrow 2^{|S|}$ is the notion of adjacency between solutions; and $f : S \longrightarrow \mathbb{R}$ is the mapping from solutions to their fitness.

2.5 Local Optima Networks: Background & Definitions

2.5.1 Complex Networks

A complex network [48] consists of a set of elements, which are called *nodes*, and the connectivity between them, which are the *edges*. Networks can be *directed* — which means that edges have an orientation and relate to a source and destination node — and they can be *weighted*, which means that edges take on a weight typically associated with strength or likelihood of connection.

The definition is sufficiently abstract and generic that systems from markedly different fields can be captured with the model and analysis methods can be shared. Well-known examples include neural networks in the brain; social networks on media sites; the internet; the spread of disease; routes taken by food delivery trucks; and nurses visiting patients at home.

Metrics. There are a wealth of metrics which have been proposed for and computed on complex networks. These can be useful in offering insight into network behaviour and dynamics, or in the classification or categorisation of networks. A comprehensive taxonomy is available [48] and this serves as the fundamental information source for the following descriptions.

The most intuitive features include the number of nodes, the number of edges, and the edge-to-node ratio. The number of nodes indicates the size of the network; the number of edges, how densely connected it is; and the edge-to-node ratio, how many connections nodes typically have.

There are a family of features which relate to the *degree* of nodes. This means the number of connections that they have. In a directed network, the *in-degree* is the number of edges which are directed towards the node. *Out-degree* is the outgoing edges. Where the edges are weighted the weights can be considered, which produces a *weighted in-degree* and *weighted out-degree*. While degree features can give information about individual nodes, often the mean is calculated across the whole network. Other metrics include the *assortativity*, which is the extent to which nodes which are similar are likely to be connected to each other; the *clustering coefficient* (the likelihood that neighbours of a node are also connected to each other); the *diameter*, i.e. the longest path between two nodes (the network width); and measures concerning *self-loops* (where a node is connected to itself).

2.5.2 Local Optima Network Model

Local Optima Networks (LONs) [11] are a fitness landscape tool dedicated exclusively to local optima and their connectivity patterns. In the model, we have a complex network $LON = \{V, E\}$ where V are the local optima. Local optima have surrounding *basins of attraction* [49], which means that members of the basin will be transformed

to the local optimum after local search. The E encode connectivity potential between the V. LONs have gained research traction and attention and have been extracted for NK Landscapes [11, 50, 51, 8]; Quadratic Assignment [52, 5, 53, 15]; Permutation Flowshop Scheduling [54, 12]; Travelling Salesman [55, 19]; Number Partitioning [56]; feature selection [14]; Genetic Improvement problems [57, 58]; computational protein design [59]; in multi-objective optimisation including *pmnk*-landscapes [60, 61]; resource-constrained project scheduling [22]; in continuous optimisation [62, 63, 64]; for MAX-SAT [65]; for inventory routing [20]; and for parameter configurations [66].

Initially, LON edges captured basin of attraction adjacency [11, 50, 52]: there is an edge traced between the local optima for basins b_i and b_j if there is at least one solution s_1 in basin b_i which is a single search operation apart from a solution s_2 in basin b_j. Edge weights are basin transition probabilities; that is, the average probability prb for s_1 in b_i to transform to s_2 in b_j:

$$prb(b_i \rightarrow b_j) = \frac{1}{\sharp b_i} \sum_{s_1 \in b_i} prb(s_1 \rightarrow b_j) \tag{2.3}$$

where $\sharp b_i$ is the size of basin b_i. The edges in this design are called basin-transition edges. The effect of pivot rule during the construction of basin-transition LONs has been studied [67]: first-improvement for enumeration of basin-transition edges results in densely connected LONs, although the probabilities encoded in the LON are often diminutive. That means that those search trajectories are not probable. Using best-improvement produces sparse LONs containing shorter paths to the global optimum. In the study, these had comparatively more heavily-weighted self-loops, which means that search algorithms are more likely to become trapped within the sphere of a local optimum. Extracting basin-transition edges is computationally expensive and as I mentioned the process leads to dense, almost complete LONs and these are challenging to study for any meaningful structure [67, 52]. This led to the development of an alternative sort of edge: escape edges [4]. Escape edges are calculated with respect to two parameters; the first is a distance function (a search operation) defining connectivity between solutions, d, and D is the maximum applications of d separating two solutions which form a LON edge.

An escape edge $e \in E$ is traced between LO_i and LO_j if there is a solution s satisfying the conditions $d(s, LO_i) \leq D$ and $h(s) = LO_j$, where $h(s)$ is the function mapping s to a local optimum with best-improvement hill-climbing. The weight of e is the number of solutions satisfying this condition, i.e. $\sharp\{s \in S \mid d(s, LO_i) \leq D$ and $h(s) = LO_j\}$, taken as a proportion of reachable solutions at this distance i.e. the probability of "escape".

2.6 Fractal Geometry & Fractal Dimension

The notion of a Fractal Dimension (FD) for patterns was conceived by Mandelbrot [68] and is defined as a complexity index which captures how the detail in a pattern changes with the resolution used to measure it. The fractal dimension can be computed as the ratio between the natural logarithm of the extent of detail observed and the natural logarithm of the scale used to measure it:

$$fractal\ dimension = \frac{ln(detail)}{ln(scale)} \tag{2.4}$$

To appreciate fractal dimensions we can begin by contemplating the recognisable shapes associated with topological dimension and with classic geometry: a one-dimensional line; a two-dimensional square; a three-dimensional cube.

These shapes are smooth and simple. In Figure 2.1 we can observe the scaling complexity for a square. In Figure 2.1a the scale of measurement, m, is the length of one side. This scale detects the whole square as a single unit of detail. Moving onto Figure 2.1b we observe that when the resolution is twice as fine four smaller squares are measured. These are four units of geometric "detail". Similarly, when m is one-quarter of the length of a side of the square (scaling factor of four; see Figure 2.1c) this results in the detection of sixteen squares (units of "detail"). In the case of all three Figures the fractal dimension, which is obtained by inserting the numeric values for scaling factor and units of detail into Equation 2.4, is two. When rearranged the Equation stipulates that for a given scaling factor m the extent of detail observed will be m^2. For a square, the fractal dimension matches the topological dimension: this is a smooth and straightforward shape from classic geometry. Different topological and fractal dimensions for a pattern indicate the presence of fractal geometry.

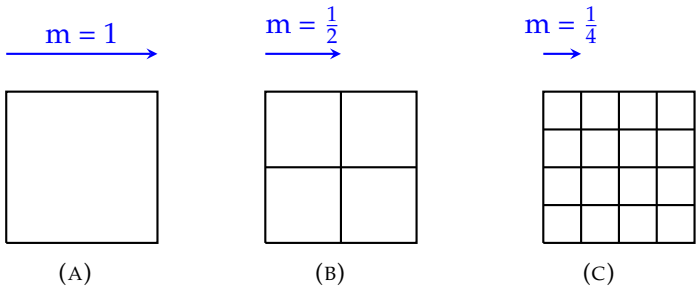
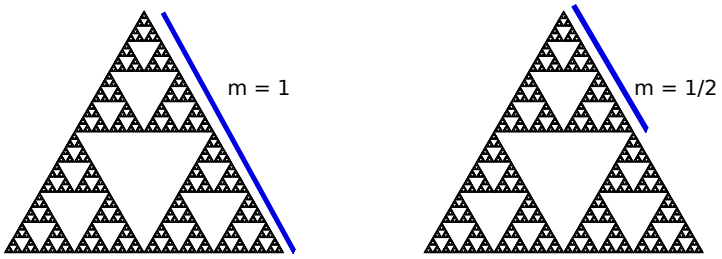


FIGURE 2.1: Units of detail detected under different scales of measurement for a square; m is the measurement scale

In that case, the way detail scales with increased resolution cannot be captured with topological dimension and the pattern is more spatially complex or less smooth. An illustrative example can be seen in Figure 2.2 with the Sierpinski Triangle.

Figure 2.2a shows that when m=1 (the length of one side of the triangle) the complete triangle is measured as the only unit of detail. If we increase the resolution twofold as in Figure 2.2b three smaller copies of the large triangle are now detected. Fractal dimension can be obtained by solving for the equation $m^{fractal\ dimension} = detail$ (this is Equation 2.4 rearranged) and in this case the exponent is not an integer. The equation becomes $2^{fractal\ dimension} = 3$ which results in a fractal dimension = $\sim$1.585.

Fractal dimension analysis can produce and characterise spatial and geometric information about real-world complex systems. It has been used, for example, in engineering for detecting cracks in plate structures [69]; in biology for characterising the tortuosity of animal trails [70]; and also in medicine for characterising mammographic patterns [71] and detecting colon cancer [72]. A pattern which has fractal dimension higher than its topological Euclidean dimension contains fractal geometry [73]. A *multifractal* system [74] requires more than one fractal dimension due to spatial heterogeneity (together the dimensions form a multifractal spectrum), which means that the scaling of the pattern detail follows different rules in different parts of the pattern.



(A) (B)

FIGURE 2.2: Units of detail detected under different scales of measurement for a Sierpinski triangle; m is the measurement scale. The pattern has topological dimension two and fractal dimension ∼1.585

2.7 Statistics

I now detail some of the statistical metrics or methods which are used in the experimentation conducted for this thesis.

2.7.1 Linear Regression

Linear regression is used to obtain information about the relationship between one or more predictor variables and a dependent (response) variable [75]. Where there is a single predictor, this is *single* linear regression; the use of more than one predictor is *multiple* linear regression.

Linear regression requires the assumption that relationships are linear; this is not always the case with LON features and algorithm performance [60]. Therefore, although multiple linear regression is used in the thesis it is always used in conjunction with random forest regression, which models non-linearities well [76] and is described in the next Section. A caveat of linear regression in the context of the thesis contributions is that LON features tend to be correlated with one another (multiple collinearity), which complicates this type of regression. The analysis can still be useful [17] — although the variation of coefficient estimates may be large [77].

2.7.2 Random Forest Regression

Random forest regression [76] is an ensemble method that computes the average prediction from multiple individual decision trees, and incorporates both the random sampling of observations and also the use of random sub-sets of predictors for node splits in the trees.

This type of regression is robust to the effect of outliers and also to predictors which have non-linear relationships [78]. I expect LON feature vectors to contain outliers and non-linearity [17, 60]; therefore, I use random forest regression in this thesis for building models for metaheuristic performance prediction and explanation using LON features.

2.7.3 Correlation Coefficient

A correlation coefficient quantifies the extent to which there is an association between the values of two variables [79]. Pearson's correlation coefficient is a commonly used method [80] which captures a linear relationship. Spearman's rank coefficient is another, and is based on the *rank* of observations. Some authors have found

Pearson's correlation coefficient does not perform well with highly non-normal distributions [81, 82]. De Winter *et al.* stipulate that Pearson correlation can be susceptible to outliers, while Spearman is more reliable to this end [82]. I use Spearman correlation instead of Pearson to show pairwise associations between variables in this thesis. The reasons for this can be summarised:

1. Real-world distributions often contain noise [83];
2. The distributions for LON features are frequently non-normal — this can be observed in plots within the contribution Chapters of this thesis — and Spearman correlation is *non-parametric*, meaning it does not make assumptions about the distribution (such as the normality assumption).

2.7.4 R^2, Marginal R^2, & Conditional R^2

The R^2 is the proportion of variation in a response variable which is explainable using predictors in a model [84] and for multiple regressors is computed as the correlation coefficient between the response values and the predicted values, squared. R^2 is used as a summary statistic for models in the thesis contributions.

A mixed-effects model is one where random effects, which are derived from observations belonging to particular categories, are controlled for. The R^2 in this case is in the form of the *Marginal* R^2, which is the amount of variation explained after the random effects have been accounted for, and the *Conditional* R^2 — the amount explained if including the random effects. These metrics are involved in this thesis as summary statistics when I use mixed-effects modelling.

2.7.5 *P*-value

A p-value [85] describes the probability of a relationship occurring by random chance and not because of a genuine trend (i.e., the probability that the null hypothesis is true). It follows that lower values are desirable and the generally-accepted maximum threshold for indication of statistical significance seems to be 0.05 [86]. Nevertheless, a p-value of less than 0.05 does not mean that the null hypothesis is definitely false; as an example, $p = 0.01$ means that there is a 1% chance of getting the observed value if it is indeed true (that is, there is no statistical significance). In this thesis, p-values are computed with respect to the Spearman correlation between pairs of variables and for predictor coefficient estimates in models.

2.7.6 MSE: Mean Squared Error & RMSE: Root Mean Square Error

The mean squared error (MSE) [87] is a quality metric for models and is calculated as the mean difference, squared, between the predicted values and the true values. Similarly, the root mean square error (RMSE) [88] is the square root of the MSE and comes in the same units as the response variable. MSE and RMSE are used at appropriate points in the incumbent analysis as indicators of regression model quality.

2.7.7 Bootstrapping

Bootstrapping [89] is a technique used to estimate the true value for a statistic through random sampling with replacement. Subsets of observations are taken from the sample and the value for the chosen statistic is computed for that subset. Bootstrapping

is useful to infer properties about the population while employing only the information present in the sample at hand; it is used to estimate linear and random forest regression model statistics presented in this thesis.

2.8 Summary

The concepts which have elementary roles within contributions of this thesis are now properly defined. In the next Chapter I review and critically analyse relevant and related literature, with the aim of carefully contextualising the location and significance of the work in the contribution Chapters that follow.

Chapter 3

Literature Review

With fitness landscapes described and retaining the lucidity of the metaphor in our minds, I will begin to document the topological features which have been studied in computational landscapes. Section 3.1 describes a collection of prevalent features. Section 3.2 is for *Local Optima Networks* (LONs) in particular, followed by Section 3.3 which concerns LON construction algorithms. Works relating to *funnel*-like organisation of local optima, which is involved in the thesis contributions, follow in Section 3.4. This begins with a history of the definition for funnel, followed by critical analysis of studies which employ the definition used in experimentation within this thesis (Section 3.4.1). Similarly, literature on *fractal geometry* in the context of fitness landscapes is described in Section 3.5 with directly-relevant works critically analysed. The final Section concerns general landscape analyses for either explanation or prediction of algorithm performance (Section 3.6), and then LON analysis for those same purposes (Section 3.7). There is a conclusive passage (Section 3.9) which clarifies the placing of the thesis in the relevant literature before moving to the contribution Chapters.

3.1 History of Landscape Features

3.1.1 Ruggedness

The demarcation of landscapes [29] brought two associated spatial features which are still used today: *auto-correlation* and *correlation length*. Auto-correlation captures, for solution pairs, correlation between fitness and distance of the solutions. An approximation for the distribution is obtained through random walks on the landscape. In a correlated landscape, similar solutions (close together with respect to the distance function) have similar fitness. Although auto-correlation is a scalar — the mean correlation at a defined distance λ — the coefficient can be plotted against increasing values for λ. The correlation length is the maximum value for λ where there is fitness correlation for solutions separated by λ steps of the distance function. Autocorrelation and correlation length both require the assumption of statistical isotropy, in that the statistics are independent of the starting point of the random walks. The two metrics capture how *rugged* the landscape is at a low level. A subsequent study [90] stipulated that a single value (such as *auto-correlation* or *correlation length*) is insufficient in estimating ruggedness and proposed *amplitude spectra* to this end, an approach which produces a whole spectrum of landscape information. They found that amplitude spectra characterised ruggedness well and noted that correlation length can also be computed from the spectra results. Amplitude spectrum representation of landscapes has also been used to propose five associated features [91]. A *spatial-domain* fitness landscape framework has been proposed for characterising ruggedness too [92]. In the model, multi-dimensional configurations

are mapped to integer values; those are displayed using a technique for visualising 3D terrain called Digital Elevation Modelling. The slope of the dimension-reduced object can be used as a measure of ruggedness and the neutrality ratio to quantify landscape neutrality.

A fresh perspective on ruggedness was proposed and instrumented [93] using an information-theoretic lens to view fitness landscapes. Associated metrics are based on the minimum amount of information needed to describe a random walk. Slopes encountered during a walk are represented by a time-series of symbols. A symbol represents a particular gradient. If fitness stays constant for every step of the walk, a single symbol can encode the entire walk. A sequence of symbols can be studied for the number of distinct symbols and how many switches in symbols were necessary.

Four ruggedness metrics were introduced which use the symbol sequence method, all of which capture the amount of variance in the walks. The features are the *information content*; *partial information content*; *information stability*; and *density-basin information*. Information content is the number of distinct symbols needed to describe the walk; partial information content is the number of adjacent symbol changes (as a proportion of the walk length); information stability is the most dramatic gradient change on the walk; and density-basin information is the probability of adjacent symbols in the sequence being the same, and therefore captures the degree of smoothness and neutrality. Landscape information was also considered in a separate study [94] which argued that fitness landscapes do not sufficiently capture quantity and quality of information which can help steer optimisation. They proposed *information landscapes* as an alternative paradigm to fitness landscapes. These have three ingredients $[S, \mathcal{N}, t]$, with S the set of configurations, $\mathcal{N}$ the neighbourhood relation, and t is a function producing the probability that one configuration is more favourable than another. The function used to calculate solution ranking involves the probability of the solution being transformed to the global optimum with a search operation. Information landscapes were used to predict the performance of a genetic algorithm before actually running it in the study; however, they have not gained traction in the research community.

3.1.2 Local Optima and Basins of Attraction

Basins of attraction are important structures found in fitness landscapes [95]. These are sections of the solution set from which hill-climbing can result in transformation to a particular local optimum [96] (local optima are also sometimes called *attractors*). Basins of attraction can be classified as strong or weak [97]. Strong basins are unconditional: hill-climbing from a basin member will always terminate at the local optimum regardless of the operator applied. Applying hill-climbing within a weak basin, contrarily, may or may not result in convergence to the local optimum. Whether the local optimum is reached depends on the algorithmic process and operators.

Cataloguing local optima and their basins of attraction is a notoriously expensive task [98]. One study managed to demonstrate that their method for local optima estimation is computationally efficient [99]; they used random sampling and steepest-ascent searches. An alternative was provided in a later study [100]: their approach used confidence intervals, concerning the probability of all local optima being found and also for estimating the number of local optima. Hernando et al. [98] implemented two techniques for estimating the basin of attraction sizes. The first technique samples solutions randomly and calculates how many are assigned to a particular local optimum; the other samples solutions at a set distance from a

particular local optimum to see how many of those are assigned to it. Along a similar vein, an empirical comparison has been conducted between two local optima estimation algorithms [101], with results suggesting that simple random sampling is best where there is no prior information about the landscape. A metric called the *escape-rate measure* was proposed [30] for gaining a snapshot of the basins of a fitness landscape. The principle is straightforward: several attempts are made at escaping the attraction towards a local optimum, thereby estimating the number and distribution of local optima.

3.1.3 Neutrality

Neutrality is a property of landscape anatomy which is often encountered by heuristic search algorithms. Two solutions are neutral if they have the same fitness evaluation. If they are adjacent then they are called *neutral neighbours* [102]. The *neutral degree* is the proportion of directly adjacent neighbours of a solution with the same fitness. Sampling the neutral degree of solutions can be a good indicator of the extent of neutrality present. An easy way to do this is with *neutral walks*; the use of these as a landscape analysis tool crystallised in [103]. Neutral walks allow exclusively steps with no fitness gradient. The result of a neutral walk is a scalar: the length of the entirely neutral path through the landscape. A *neutral landscape* must have a significant proportion of neutrality between neighbouring configurations [103].

A *plateau* is a sequence of neutral neighbours; that is, a path of adjacent solutions with the same fitness [102]. A plateau is also sometimes called a *neutral network* [104]. It has been noted that *Local Optima Plateaus* can occur [102], and that to escape a suboptimal plateau a neutral walk must be conducted in search of a *portal* solution, whose neighbour has superior fitness (and thus is not a member of the plateau).

One study built neutral networks (plateaus) to tackle problems which give rise to uninformative, highly uncorrelated fitness landscapes [105]. Another proposed a collection of metrics calculated from neutral networks (plateaus) [106]. This set of features taken together can reflect the overall neutral nature of the fitness landscape. Among the features are the mean fitness improvement upon escape from a plateau, the mean neutrality ratio, the non-improvable solutions ratio and the profitable and unprofitable mutations ratio.

3.1.4 Evolvability

Evolvability is concerned with the capability of a population to evolve towards better fitness. There are several methods proposed for quantifying this. The term *searchability* has also been used in evolutionary algorithm research to allow inclusion of single-point based methods. An intuitive method for calculating evolvability has been proposed [107]; it produces objects called *fitness clouds*. To build a fitness cloud, a sample of solutions have the fitness of a one-mutant neighbour calculated. The fitness gradients indicate the evolvability of the landscape under the chosen distance operator. Fitness clouds themselves are plots of the pairwise fitness of the evolutionary pairs (a pair being the original solution and the transformed solution). A universal view of the landscape evolvability is obtained. Similarly, *evolvability portraits* [108] take a snapshot of landscape evolvability. Solutions are sampled and a set of evolvability metrics are calculated. These include the probability of improving mutation and the expected fitness of the mutated solution. The features are plotted as a function of fitness; this is an evolvability portrait. Along the same vein, the *negative slope coefficient* has been proposed [109] as a refined *fitness cloud* method. The

cloud is partitioned into fitness levels; from there, the slope between the centres of adjacent levels is calculated. The *negative slope coefficient* is the sum of all negative slopes and the amount of deleterious mutation is captured. *Fitness-probability clouds* [110] have also been used for evolvability analysis. Fitness-probability clouds are similar to fitness clouds; however, instead of fitness-fitness pairs, the plot comprises fitness-probability pairs. The probability is the likelihood that a mutation will result in fitness benefit. A related metric is the *accumulated escape probability*, which is derived from a fitness-probability cloud: the mean probability of beneficial mutation. A value close to one implies a highly searchable landscape under the chosen operator.

3.1.5 The Global Structure

Fitness Distance Analysis (FDA) [111] aims to detect the overarching shape in a fitness landscape. During FDA random solutions are sampled and the distance to the global optimum (or best known solution) is calculated alongside their fitness. The Fitness Distance Correlation (FDC) is the correlation coefficient of these two vectors. FDC has been used and cited extensively in investigating whether fitness landscapes are globally convex (in the case of a minimisation problem) or globally concave (for maximisation) [112, 113, 114, 115, 116, 117, 19, 59, 118]. A high FDC indicates an organisation of solutions which has been termed the *big valley* [119], where the fitness of a solution is correlated with its distance to the global optimum. The *big valley* phenomenon has alternatively been called a *bowl*, a *massif central* (for maximisation problems), or a *funnel*; the notion of funnel is properly elucidated in Section 3.4 shortly.

Although FDA has been consistently popular as a landscape analysis tool, it is not without shortcomings. One study showed that a problem which is easy for GAs had a low FDC; a low FDC is used to infer problem difficulty for a GA [120]. They found instead that evolvability could predict GA performance, and notably that FDC calculated using *crossover* as the distance function could, too.

Other pursuits into global landscape analysis have centred exclusively on local optima and their connectivity. This paradigm abstracts away from elemental distances and fitness gradients. Concentrating on this local optima level or space with the aim of reducing unnecessary information has been implemented in a study which extracted properties such as distribution of local optima, the FDC, and the attractiveness of each local optimum [121].

A separate work, more theoretical in nature, formulated fitness landscapes as multi-layered objects and *level two* was the local optima level [122]. The paper asserted that acknowledgement of the local optima level as a landscape in its own right required a definition for neighbourhood or adjacency at that level.

3.2 Local Optima Networks

Having defined LONs already in Chapter 2, I now detail relevant and important contributions which use the analysis.

Although several LON works have implemented escape edges for mutation operations [4, 54, 19, 123], this is not always the arrangement. Efforts have also analysed evolutionary recombination edges for LONs [55, 23]. In one study, edges are

defined according to partition crossover, a specialised operator from the Travelling Salesman Problem literature [55]. Pairs of local optima are subject to partition crossover and any fitness-improved offspring they produce are retained. Local search is applied to the offspring to ensure that they are local optima. An edge is traced from both parents and is oriented towards the fitness-improved (and now locally-optimal) offspring. A later work observed a LON setting where edges are defined according to any of three operations: crossover, mutation, or perturbation [23]. In that study, an edge is delineated between two local optima, LO_i and LO_j, if LO_i can be transformed to LO_j using one of the three search operations followed by k-opt local search.

Most contributions in the LON literature use benchmark combinatorial optimisation domains (see for example [11, 52, 8, 19]). LON and landscape analysis in this thesis is principally conducted on two of them: NK Landscapes and the QAP, which have been detailed in Section 2.2. NK Landscapes were the original testbed for LONs [11] and have been proficient for several proof-of-concepts, including the effect of pivot rule on LONs already mentioned [67], LONs with neutrality [124], the escape edge model [4], the first venture into using LONs for performance prediction [17], as the chosen domain for a LON *sampling* construction algorithm [51] (there is a segment concerning this type of algorithm later in Section 3.3), and for bi-layer LONs [16]. NK Landscape LONs have been subject to centrality and cluster analysis too [125, 8, 126]; the features were used with success to predict Iterated Local Search algorithm performance on the landscapes (see Section 3.7 for further discussion).

LONs for the QAP were first modelled with basin-transition edges [52]. Some metrics from the LONs, such as weighted degree, provided differentiation between problem classes; this being said, it was asserted that the LONs were confusingly dense, and no statistical link to metaheuristic performance was established.

Later, the LONs were subject to community detection [5] and this revealed more differences between problem classes: LONs extracted from "clustered" instances show modularity, while LONs for "uniform-random" instances had no such structure. A much larger set of QAP LONs were generated the following year [21]. The topological features were used to predict algorithm performance. That study is highly relevant to the contributions presented later in this thesis, and is therefore discussed further in a dedicated Section concerning LON performance prediction (Section 3.7).

The QAP has also been a testing ground for LON construction algorithms [7, 24, 25]. These consider the escape edge LON model and define the edges according to perturbation operations. The algorithms are described in detail next in Section 3.3.

3.3 LON Construction Algorithms

3.3.1 Exhaustive Enumeration

The initial LON construction algorithm fully enumerated the basins of attraction to calculate *basin-transition* edges, and that schema has been used many times [11, 124, 52, 21, 8, 22]. The algorithm runs, from every possible solution, a best-improvement local search to obtain the mapping to a local optimum. The local search operations used are elemental; a single bit-flip for NK Landscapes, which are binary-encoded [11] and a pairwise exchange in the permutation strings representing QAP or resource scheduling solutions [52, 22]. It follows that the constructed LON consists of best-improvement local optima for nodes, and basin of attraction adjacency for edges. Edges are weighted with *probability* of basin transition.

As mentioned in Section 2.5, *escape edges* are an alternative edge model for LONs. The study that designed them [4] also algorithmically extracted them starting with best-improvement local search to find local optima. An escape edge e is delineated when there is a solution s which is at a distance of maximum D applications of d away from a local optimum LO_i, and where s is within the basin of attraction of another local optimum LO_j. Although the proposal paper conducted full enumeration for *complete* LONs [4], I note here that escape edges have a particular pertinence with respect to sampling for *approximate* LONs: edge calculation can be augmented with an existing optimisation algorithm as long as the algorithm contains local search. The distance function d can be an arbitrary operator (or sequence of operators) and when its application is followed by local search, the requirements for an escape edge can be met: the solution s which resulted from maximum D applications of d to the local optimum LO_i is in the basin of attraction of local optimum LO_j and will transform to it following local search.

The suitability of escape edges for sampling, especially using existing domain-appropriate algorithms, seems to have established their place as the more popular of the two edge models [7, 19, 23, 123, 25, 24]. Escape edges have also been used in exhaustive enumeration, however [56, 127, 14, 15].

Exhaustive enumeration of LONs in the beginning was essential to establish them as a useful tool and for testing analysis techniques on the networks. Unfortunately, due to computational expense, exhaustive enumeration limits analysis to problems which are diminutive in size compared to most "real-life" situations. For NK Landscapes, this is approximately N=18 (binary strings of length 18) and for the QAP, N=11 (11 facilities assigned to 11 locations). To properly prepare LON analysis for prospective "real-world" problems, it is critical we develop intelligent construction algorithms for sampling.

3.3.2 Sampling

In pursuit of modelling large combinatorial fitness landscapes as LONs, sampling algorithms which produce approximate LONs have been proposed [51, 7, 19, 24, 25, 127]. The first such algorithm was instrumented on NK Landscapes [51], where a previous work provided the set of local optima for nodes in the LON. In the algorithm, first-improvement local search begins from random solutions 10,000 times. For each candidate solution during this process the basin of attraction membership is ascertained. If the mapped local optimum is within the predefined local optima set, empirical basin-transition probabilities are updated inside an adjacency matrix. In this way, a LON is constructed which reflects the empirical dynamics of hill-climbing with restarts.

A follow-up study introducing a sampling algorithm for QAP LON extraction appeared the same year [7]. This method identifies the nodes (that is, the local optima) by applying first-improvement hill-climbing to $2000n$ random starting solutions (n being the problem size). The connectivity between the resultant local optima is estimated by perturbing the local optima and then hill-climbing again for each of them $20n^2$ times; if the obtained local optimum is also in the node set, an edge is added to the LON. Edges are weighted with the probability of transformation, that is, the frequency that source local optimum LO_i was transformed to destination local optimum LO_j. This construction algorithm framework was also used for the Travelling Salesman Problem later [13].

Another algorithm for the Travelling Salesman Problem augmented a search algorithm with LON construction, such that they are concurrent [19]. A competitive

ILS heuristic for the domain, Chained Lin-Kernighan (CLK) [128], is the base algorithm. During optimisation, nodes and edges are registered as they are encountered by CLK. This construction strategy has appeared in implementations for the QAP [24] and MAX-SAT [65] as well; in each case LON building is augmented with an ILS algorithm for the problem domain. ILS-based LON algorithm systems begin r independent ILS runs from random solutions. Runs are terminated after t iterations with no improvement. Local optima which are reached form the node set. Trajectories between local optima using perturbation then local search form the edge set. The nodes and edges from r runs are amalgamated to construct a single LON for the problem instance. ILS is not the only heuristic this is possible with. In fact, a LON algorithm which uses *Generalized Partition Crossover* (a highly effective recombination operator) was proposed [23] for the Travelling Salesman Problem and then used again for NK Landscapes [129].

A comparatively more divergent procedure is called Snowball Sampling (SS) [25]. Instead of recording the movements of an optimisation algorithm, SS conducts a random walk on the local optima space. Local optima are identified by a best-improvement local search. From each local optimum on the walk a recursive local optima neighbourhood exploration occurs, i.e. local optima neighbours of local optima are identified. The probing is done with random mutation followed by local search. A separate and distinct algorithm for approximating LONs [127] has been applied to NK Landscapes. The approach reduces the amount of computation during construction by retaining previous mappings from solutions to local optima in memory. The resulting LONs are samples because local search does not run from every possible configuration. Aside from this and the efficiency modification the algorithm resembles a reduced version of the exhaustive algorithms seen in earlier studies [11, 4]. The framework has also been used to compute feature selection problem LONs [14].

3.4 Funnels

Funnel-like organisation of local optima has been observed in the study and visualisation of physical energy landscapes [130, 131]. Funnels have alternatively been called *super-basins* [132] or *barrier trees* [133].

Organisation of local optima has also been considered for computational fitness landscapes. One study theoretically mapped basins of attraction at different "levels" [122]. The local optima *of the local optima* were called "local minima at level two", and it was stipulated that a suitable neighbourhood structure must be defined for each level within this multi-level fitness landscape paradigm. "Level two" in the model resembles later and more explicit proposals for the characterisation of funnels [134, 56] (see Section 3.4.1). A more empirical definition for fitness landscapes [135] considers a funnel to be an overarching structure leading down to the global optimum. In the paper, it is argued that funnels merit more consideration during the design of optimisation algorithms. Physicists introduced and solidified the concept of funnel the same year [131], in terms of local minima and saddle points in energy landscapes. Studying the phenomenon of *communities* of local optima was later proposed as a procedure for identifying local optima organisation [5]. A later work suggested mapping funnels directly using communities [8], arguing that they are equivalent.

A separate empirical study detected multiple funnels in high-quality landscape regions of Travelling Salesman Problem instances [119]. The bottom of the funnels

(termed *funnel floors*) are judged to be the final solutions obtained from runs of CLK. From the funnel floors, neighbouring local optima are searched for. These comprise the funnel that surrounds the funnel floor. This schema has also been used in a later study on multiple funnels [136].

A further interpretation of funnel organisation [134] stipulates that a funnel is 'the set of configurations that reach the global minimum by iterating exits from gradient basins over the lowest gradient saddle.' That paper asserts that funnel members are local optima from which there is an ever-improving path directed towards the global optimum. The authors also considered the *funnel-partitioning* of a landscape, thereby acknowledging the existence of sub-optimal funnels.

A novel set of features which can classify a landscape as either *funnel* or *random* was proposed for real-valued optimisation problems [137]. The features are taken from a $500n$ (where n is the problem dimension) function evaluation sample. The feature analysis has the underlying assumption that funnel landscapes will contain a multitude of paths directed towards the highest-quality local optima. The definition for *funnel* in the study does not seem to consider bi-funnel or multi-funnel landscapes as being within the funnel landscape class.

3.4.1 Recent Advances

I will now directly contextualise the funnel feature analysis which is involved in this thesis by addressing closely-related research papers. *Coarse-Grained Barrier Trees* were proposed to represent fitness landscapes at the level of local optima clusters [138]. The technique identifies barriers between clusters. The barrier tree is analogous to funnel-like organisation, where local optima are connected within their own funnel but have less or no reachability to other funnels. Indeed, a funnel was considered to be equivalent to a connected component of local optima in a related study [19]. Funnels have been visualised and conceptualised in three dimensions to assess how local optima fitness relates to their organisation in space [139]. Soon after that paper, a more rigorous definition for funnels in combinatorial optimisation was proposed [56]. *Monotonic sequences* of local optima are extracted from LONs. These are paths of local optima with only fitness improving gradients (W.R.T the search operation used to define the edges in the LON) between them. The termination points of monotonic sequences are *sinks*. These are also the nodes in a LON with no outgoing and improving edges. Collections of sequences which terminate at the sinks are extracted using breadth-first search on the LON, beginning from the sink nodes. The blueprint for calculating and defining a funnel has been used several times since its proposal [140, 123, 24, 141, 14, 65] and I use this paradigm for the funnel analysis conducted in this thesis.

A related study uses the monotonic sequencing model in order to crystallise our understanding of funnels in the landscapes of Travelling Salesman Problem instances — differentiating between *attractors* and *funnel floors* in doing so [142]. Attractors, it was stipulated, are the final solutions obtained from runs of a metaheuristic, which were termed "funnel floors" in previous literature [119]. Recently, a study identified sub-optimal funnels or clusters (they use the terms interchangeably) of local optima near the global optimum in computational protein design [59]. Although there was no formal definition for funnels or clusters provided, these were observed through visual inspection of LONs and from statistical features of the networks. The authors confirmed that search transitions between clusters were unlikely and, as such, the existence of funnel structure was implicitly identified.

A space in the literature exists concerning precisely how funnels are related to metaheuristic algorithm performance. Some preliminary steps forward have been made [123, 24, 14]. One paper found links between search difficulty and both the *number* of funnels and the *size* of the optimal funnel using correlation analysis [123]. Interactions between variables were not considered. Similarly, another work argued that there was a correlation between search and the attractiveness and also size of the optimal funnel through visual inspection of distribution plots [24]. A paper the same year noted that a problem instance whose landscape exhibited several sub-optimal funnels also had a low metaheuristic success rate [141] but did not compute statistics to this end. In this thesis, I use techniques which consider interactions between variables — and also non-linear relationships — to capture the contributions of funnel features to the prediction and explanation of metaheuristic performance. I argue that this approach is necessary for illuminating our understanding of the role funnels play in fitness landscapes and optimisation.

3.5 Fractal Geometry in Landscapes

Recalling and retaining the preliminaries concerning fractal geometry from Chapter 2, I note that the idea of fractal geometry in fitness landscapes of optimisation systems dates back to 1991 [143] when simulated annealing was proposed for use on fractal *energy* landscapes. Soon after, in evolutionary computation literature, a relationship was drawn between the *correlation length* (see Section 3.1.1) of Travelling Salesman Problem landscapes and metaheuristic algorithm success, although this was done only in conjecture. The authors noted that the *autocorrelation function* is related to fractal geometry, and also that Sorkin's definition for 'fractalness' in landscape systems [143] is related to fractal dimension.

A later study stipulated that there is fractal geometry in a fitness landscape topology if the *correlation length* (see Section 3.1.1) is proportional to the landscape diameter (i.e. the maximum distance between solutions) [144]. That study considered the elemental neighbourhood function for the problem and therefore analysed its most basic fitness landscape. As such, the deductions concerning structure were not derived from a metaheuristic-induced fitness landscape. Mentions of fractals in fitness landscape analysis literature were sparse for a time, although one study contained a remark that considering fractal structure in landscape analysis and subsequent design of algorithms could be important [113]. The idea was not implemented as part of their study. Several years later fractal analysis was included as part of a dynamic fitness landscape study on the targeting of chaotic systems [145]. The fractal analysis method was visual inspection of hierarchical zooms conducted on a 2D depiction of the landscape, before calculating the ratio between detail and scale, i.e. fractal dimension (refer to Section 2.6 for the full definition).

Fitness landscapes of a targeting problem have also been viewed with a fractal-geometric lens [146]. The analysis considered how features of landscape ruggedness (*correlation length* and *information content*, see Section 3.1.1) transform under different scales of measurement. The results showed that the fitness landscapes involved were indeed fractal and the landscape features were *scale-invariant*. The findings were not linked to metaheuristic performance and were limited to a single problem domain.

3.5.1 Fractal Geometry at the Local Optima Level

The fractal analysis contributions presented in this thesis are contextualised within the literature in this Section. Works which are directly related to the contributions are critically analysed.

A study on fitness landscapes of the graph bi-partitioning problem included fractal analysis [121]. Notably, the focus was exclusively on the local optima level — as is the case in this thesis. An investigation into the *big valley* structure of local optima was presented. FDA of local optima sub-spaces was used in experiments to show that the overall local optima level conforms to the same pattern as sub-spaces of it (i.e. *funnel*-like organisation). The analysis carried the cardinal assumption that local optima form a single big valley or funnel and tested only seventeen problem instances. They considered the elemental hamming distance function for defining space between solutions. Later on in this thesis, in the contributions Chapters, an alternative paradigm for fractal analysis of the local optima level is introduced, where the spatial layout is according to metaheuristic search paths. In the previously-mentioned study [121] sub-sets of sampled local optima were tested for funnel-like organisation (i.e. whether the central local optimum also had the best fitness). Indeed, most of them did; this implied the existence of a higher-level global funnel structure. Positive correlations were shown for fitness and distance within local optima sub-sets although those only had an average correlation of around 0.56 compared to a *global* average FDC of 0.84. The authors do not address the correlations being less pronounced in smaller sub-spaces of local optima. The discrepancy between the correlation strengths raises the question of whether sub-sets of local optima really mimic the universal spatial pattern. The pattern may not actually be a single funnel replicating, and treating it as such might miss a multiple funnel pattern. The study does identify clusters of local optima which are distant from the global optimum and speculates on the existence of sub-optimal funnels, possibly contradicting the assertion of fractal replication in the form of single funnel structure. The landscape analysis was not backed up with direct empirical comparison to algorithm performance. Fractal dimension was not part of the fractal analysis. The work in this thesis includes fractal analysis on the local optima level captured as a complex network and is conducted on both single-funnel and multiple-funnel landscapes. The method of fractal analysis is new for the local optima level (*box-counting* for a fractal dimension, i.e. index of spatial complexity) and local optima layout reflects sequences of metaheuristic search operations rather than fitness landscapes according to elemental distance functions. The relationship of fractal complexity of the local optima level to algorithm performance is, in addition, directly and empirically established using predictive modelling.

Another paper which is essential for properly contextualising this thesis took the position of viewing fitness landscapes with a hierarchical lens and separated them into levels [122]. For each level there is a single local optimum at the bottom (assuming minimisation) and a level-specific distance function for defining solution connectivity. All levels therefore exhibit the same topological structure, which implies fractal geometry. In dividing the landscape this way the local optima level is considered and referred to as *level two* by the author. He remarks that if there is more than one local optimum on level two (in funnel terminology these are called *funnel floors* or *sinks*, see Section 3.4) this will present problems for metaheuristic algorithm performance, although this stipulation was based on speculation.

A few years later, artificial fractal landscapes were created [147] and it was demonstrated that a locust swarm algorithm is proficient for optimisation on this type of

landscape. The authors propose this algorithm where there are multiple funnels, asserting its success as being related to the mechanisms not focussing solely on historical fit solutions (convergent search behaviours). A limitation to this study is that the generated landscapes are *precisely* self-similar (i.e. parts of the landscape exactly resemble the whole), which is a situation unlikely to arise naturally in real-world problems or even in other synthetic problems where the fractal geometry has not been specifically enforced.

3.6 Landscape Geometry & Metaheuristic Performance

An essential component of the evidence provided in the contribution Chapters of this thesis involves the relationships between fitness landscape features and algorithm performance on the underlying combinatorial problem. In particular, experiments contain regression models to help explain or predict metaheuristic performance. The following Section describes works where fitness landscapes have been used in this way. There are a great many papers fitting this description; as such, only a representative sample is discussed.

Fitness landscape features which can explain GA performance were identified as far back as 1992 [2]. Similarly, a study used two well-known search operators for the Travelling Salesman Problem to build fitness landscapes and compared the FDC for them [148]. The FDC helped to explain performance discrepancies which had been observed using the two search operators for optimisation. FDA has also been conducted on different classes of graph bi-partitioning problems [149]. The FDC was found to be dependent on problem class, and the authors proposed memetic algorithm variants for tackling each of those in a landscape-aware manner. TSP landscapes have also been subject to adaptive walks [150] which provide fitness and distance information for local optima. In that study, the *big valley* layout was subsequently observed and basins of attraction were found to be highly overlapping. The findings lead the authors to suggest a particular search algorithm to exploit the observed structure: a hybrid GA, which begins with best-improvement local search to place itself in the big valley and then proceeds with crossover of local optima from inside the big valley. Landscape geometry has also been deployed in performance prediction for the algorithms on the QAP using autocorrelation and FDA [32]. The authors do note, however, that FDA conducted on a completely uncorrelated landscape can lead to misleading results. A couple of years later, different landscape features were chosen to analyse multi-objective QAP instances [151]: the entropy, the diameter, and number of transformations from a starting solution to a local optimum. They found that the correlation between fitnesses of local optima can be used to suggest appropriate search strategies. Another means of landscape analysis was proposed in 2004 [30] where a random walk begins from a local optimum and proceeds towards another local optimum. Metrics taken from this walk were shown to explain the superior speed of crossover against mutation on the problems studied.

A more recent study examined the landscape geometry activated through the use of two search operators and related it to the performance of algorithms which use those operators [152]. They extracted the 'step length' (transformations from a start solution to a local optimum) and the autocorrelation as features. Under one of the two search operators the landscape had deeper valleys and indeed the results showed that superior local optima were discovered using this operator.

A sequence of papers aim to explain and predict Particle Swarm Optimisation

performance using fitness landscape features [6, 153, 154]. They did so with success using features such as funnels, gradients between adjacent solutions, and neutrality metrics. Landscape analysis has been used to tune a search algorithm in a study which predicted the optimal termination condition for Ant Colony Optimisation [155]. Landscape features served as predictors in regression models. Doing so significantly helped computational costs used by the ACO to reach good solutions. The features included *speed of optimisation* (fitness gradient from initial solution to the best known solution) and the *acceleration* (how quickly fitness improves during runs).

A contribution to multi-objective optimisation includes fitness landscape features in correlation analysis and regression for explaining metaheuristic algorithm proficiency [156]. Amongst the features are the connectivity of Pareto optimal solutions and the *autocorrelation* (recall Section 3.1.1 for definition) of the hyper-volume. In particular, *autocorrelation* of the solution hyper-volume and local hyper-volume were found to rank amongst the highest effect-size predictors. An extended version of that paper similarly used landscape features as part of a larger set [157] in a correlation study and mixed-effects regression models for predicting algorithm performance. They found and emphasised that ruggedness and multimodality appear (from the analysis) to have an important effect on evolutionary multi-objective algorithms. A recent contribution concurs [158], implementing random forest trees for algorithm performance prediction, and additionally emphasises the salient need for a vector of features for describing a landscape instead of one or two.

There is a sequence of works which concern the study of Exploratory Landscape Analysis (ELA) on continuous functions [159, 160, 137, 161, 162]; while the earlier works classified problem instances using observed features, later ones moved to algorithm selection. Feature sets typically include both fitness landscape features and those pertaining to the behaviour of the fitness function. ELA has been used with success in predicting algorithmic proficiency [161, 162].

There are surveys in the field of algorithm selection and prediction [163, 161, 164]. Smith-Miles [163] details relevant works in machine learning, operational research, and artificial intelligence; she stipulates that much of the fitness landscape research from the metaheuristic community does not delineate relationships with algorithm performance directly. Algorithm prediction methods are surveyed and proposed in another work [164]. Using mostly random forest techniques they accurately predict algorithm performances. The feature set consists mostly of features derived from the problem specification and fitness function themselves (i.e. not fitness landscape features), although one landscape measure — autocorrelation — is included. A recent paper summarises the current state-of-the-art in algorithm selection techniques [161] and stipulates that local optima work related to large features such as funnels could be important in the future of the field.

3.7 LON Geometry & Metaheuristic Performance

Contributions outlined throughout Chapters 4-7 of this thesis use LON features to explain or predict algorithm performance. Literature which is directly relevant to that is critically discussed now to reveal the path leading to the contributions made in this thesis.

A few years of purely-descriptive analysis of LONs [11, 52, 4] were succeeded by studies linking LON features to algorithm proficiency on the problems to which

the LONs were associated. The first study to use statistics to establish these relationships focused on small NK Landscape problems and computed features on the fully-enumerated LONs [17]. These were contrasted with ILS performance on the NK Landscapes. Using multiple linear regression they showed that four features could build a strong model for predicting ILS runtime: path length to the global optimum, number of outgoing edges from nodes, weight uniformity of the outgoing edges, and the nearest-neighbours degree correlation.

A study followed for the QAP, again fully enumerating the LONs and extracting features to correlate with SA and GA performance on the QAP instances [21]. None of the considered features had a statistical correlation to GA performance, but a couple (the number of local optima and the path length to the optimum) correlated well to SA search difficulty. The variables were considered in a pairwise fashion and univariate regression analysis was carried out; no combined regression model was built. Multiple linear regression models have been built in a statistical study which used features of LONs derived from Permutation Flowshop Scheduling problems [54] for explanation of iterated local search runtime. Some moderately strong models were generated, with the average path length to the global optimum dominating the contributions of predictors.

Then a sequence of studies proposed and analysed *PageRank Centrality* of the global optimum in a LON as a valuable predictor to explain ILS performance [18, 165, 126]. *PageRank Centrality* [166] ranks the importance of a node in a network, doing so by considering how well-connected the node is and also the quality of the connected notes (in terms of their own *PageRank* value). Both *basin-transition* LON edges and *escape* LON edges were included in the studies. *Basin-transition* LONs were better-suited to predicting SA performance and *escape edge* LONs were better for predicting ILS performance. Predictions were made using univariate linear models and they found that the global optimum *PageRank* within a LON is a good predictor of algorithm proficiency. There were limitations in these works though: potentially confounding variables such as the landscape ruggedness were not considered; nor were variable interactions, and the problem sizes were small.

Correlation analysis has revealed an association between communities in LONs and metaheuristic performance for NK Landscapes [8]. The number of LON communities and also the size of the community containing the global optimum showed correlations with ILS performance on the instances. This being said, the study did not control for the fact that different settings of K (which is the extent of epistasis in the problem) were used in the instance set. Larger K is known to induce higher ruggedness and therefore reduced algorithm performance, but this was not controlled for in the analysis. As a result, the apparent binding between LON communities and metaheuristic performance may not be as it appears. Instead, the instances with low ILS performance might be difficult because of the higher K and therefore landscape ruggedness, instead of LON community structure.

Using LONs, one study calculated funnels present within Number Partitioning Problem fitness landscapes [56] and found that the amount of funnels as well as the attracting strength of sub-optimal funnels were linked to the phase transition to a 'hard' instance. Instances with many funnels were also those which were in the 'hard' problem category. This was deduced through correlation analysis between funnels and instance hardness.

Another contribution extracted the LONs of Permutation Flowshop Scheduling problems [12]. Features were tested for correlation with ILS performance on the underlying combinatorial problem; of particular note in the results were the size of the optimal funnel and the *PageRank* of the global optimum.

Recently, various perturbation strengths were used to morph funnel structures in landscapes of Travelling Salesman Problem instances [167]. Funnel features of the LON such as the number of funnels and the size of the optimal funnel) had stronger correlations to metaheuristic performance than the number of local optima.

As discussed in this Section, the bulk of LON prediction or explanation works use exclusively correlation analysis (for example [21, 8, 12, 167]) to do so which carries the inherent risk of disregarding other confounding variables and perhaps interacting effects. Multiple linear models have been built (see for example [17, 54]) to address this although the authors included a disclaimer that they suspected linear regression may not properly capture non-linearities in the variables. A recent paper used random forest regression in multi-objective optimisation with the non-linearity of LON features in mind [60] and showed the contribution of features such as the *degree of connectedness* towards explaining metaheuristic performance variation.

3.8 Genetic Algorithm Theory

Although they are not direct predecessors to the work in this thesis, there are some wider concepts in the field of genetic algorithms which are related to the study of fitness landscapes and therefore merit mentioning in brief. These include the proximate optimality principle, which holds that "good solutions possess some similar structure" [168]. There is also the separation of difficult fitness functions into either "wide-gap" or "long-path" problems [169] - in "wide-gap" problems, algorithms can stall at a particular fitness level because ascending through levels is not trivial, while on "long-path" problems, algorithms traverse a lengthy route to obtain the global optimum. The "no free lunch" theorem stipulates that there is no algorithm which is best for all problems [170]. Also related is the notion of *deceptive* problems for evolutionary algorithms — these "deceive" searches towards locally optimal points but not towards the global optimum as intended [171].

3.9 Conclusions & Literature Gaps

The present Chapter has positioned contributions of this thesis within the relevant literature, and I stipulate that the issues and open questions being addressed are as follows.

3.9.1 Fractal Analysis on the Local Optima Level

Literature considering fitness landscapes using a fractal lens is scattered, which we saw in Section 2.6. LONs have not been subject to fractal analysis yet. The Local Optima Level (LOL) has been analysed with the comparison of FDC for sub-spaces of local optima to estimate fractal geometry (Section 3.5.1). In that analysis, however, there were indications that the fractal geometry was not particularly apparent; in addition, distance calculations were made on the elementary fitness landscape. In Chapters 4, 5, and 6 distance between solutions is considered with respect to metaheuristic search paths and *fractal dimension* is used to capture the *extent* of fractal complexity in a fine-grained manner.

The standard algorithm for fractal analysis of a complex network is box-counting but this method is agnostic of the *semantics* of the network, such as the fitness information encoded in a LON. This thought precedes motivation to find new ways for calculating dimension on LONs while respecting the semantics of the network. In

Chapters 4, 5, and 6 LONs are studied for their fractal complexity and new ways to define their fractal dimension are presented. LON fractal dimension metrics are used in regression models to explain and predict metaheuristic algorithm performance on the underlying problems.

3.9.2 LON Construction Algorithms

For LON analysis to be applied to real problems of large size and complexity, the development of intelligent LON construction algorithms for sampling the landscape is essential. As was evident during Section 3.3 the majority of LON papers fully enumerate the fitness landscape to construct the LONs. This limits the analysis to small problems; further, the applicability of associated analysis and observations to larger problems is not obvious. A direction for future LON research, therefore, is comparing and testing LON construction algorithms for large combinatorial problems. A few LON construction algorithms for sampling have been proposed (recall Section 3.3) although these have not tested sufficiently or compared. The nature of their sampling biases is not immediately apparent. Chapters 7 provides contributions which help to illuminate the mechanisms and effectiveness of the construction algorithms.

3.9.3 LON Features & Metaheuristic Algorithm Performance

Most studies connecting LON features with algorithm performance do so using correlations only, which can miss other confounding variables and interactions. Predictive models using LON features have been presented previously and these were acknowledged in Section 3.7; however, the matter of finding a sufficient set of features to use as predictors remains, as well as the most appropriate model set-up to use. Authors have acknowledged there is a need for modelling the complex non-linearities of LON features. This has been done in multi-objective optimisation (although not single-objective optimisation) with some success [60]; nevertheless, further studies with other experimental settings, search operations, and feature sets would fortify and extend those findings.

Contribution Chapters 4, 6, and 7 in this thesis include linear and random forest regression. These consider a variety of different predictor collections. Particularly effective sets of features for use in such models are proposed in Chapter 7 and new discoveries emerge concerning modelling LON-based algorithm performance predictions with random forest trees.

3.9.4 LONs of a Constrained Healthcare Problem

Although certain benchmark instances which have been subject to previous LON analysis are derived from real-world problems, the LON literature for distinctly 'human' problems is lacking. The large majority of studies have involved benchmark domains instead of specific real-world problems. These works were certainly needed to provide initial statistical results for LONs: large quantities of problem instances are required to show their use, and benchmark domains facilitate that. In pursuit of returning from the abstract, scenarios which are close to humanity should also be considered. To this end, Chapter 8 extracts and studies LONs for a chemotherapy schedule optimisation problem. This is a highly-constrained problem, which presents a new consideration for LON analysis: infeasible regions in the fitness landscape.

Chapter 4

Exploring the Fractal Nature of Local Optima Networks

As discussed in Chapter 3, there is little literature concerning fractal geometry in landscapes. This is important because the emergent structure of the landscape can limit or stall attempts at optimisation by search algorithms. This Chapter presents a preliminary and exploratory study of fractal analysis on local optima networks. Empirical experiments reveal relationships between LON fractal dimensions and metaheuristic algorithm performance on the underlying associated problem.

4.1 Abstract

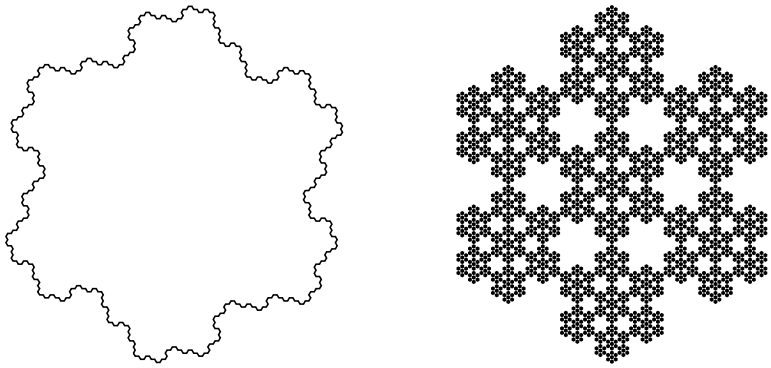
Recent literature has applied fractal analysis to complex networks. According to the definitions specified in Chapter 2, a pattern is said to be *fractal* if a part of it resembles the whole and the *fractal dimension* is a spatial complexity index which captures how the detail in a pattern changes with the scale of measure. A high fractal dimension is associated with spatial convolution. In the upcoming experiments I study the fractal nature of Local Optima Networks (LONs) for a benchmark combinatorial optimisation problem (NK Landscapes). This Chapter contributes results which draw correlations between fractal characteristics of LONs and performance on the underlying combinatorial problem pertaining to three prominent metaheuristics: Iterated Local Search, Simulated Annealing, and Tabu Search.

4.2 Introduction

Weinberger and Stadler [144] noticed that certain fitness landscapes contain fractal geometry. By increasing the landscape diameter, they asserted that landscape ruggedness scales in a way which indicates a multilevel *self-similar* structure, i.e. a fractal. They conducted random walks at the solution level for their analysis. Considering the solution level as the elemental or underlying layer of the fitness landscape we can also then consider fractal complexity at higher levels of abstraction, such as the local optima level.

A Local Optima Network (LON) [11] models the local optima level of a fitness landscape. A network is formed which encodes search path connectivity between local optima. An edge traced between two nodes means that the destination local optimum can be reached from the source local optimum by carrying out a chosen search operation (or sequence of operations). This definition for adjacency represents the neighbourhood function at the local optima level.

Fractal analysis of a LON should produce information about the spatial layout and complexity of metaheuristic search trajectories. Of particular interest is the Fractal Dimension (FD) [172], a complexity index which can assign a non-integer dimension to a pattern. The index is a ratio between the amount of detail observed and the scale of measurement used. Figure 4.1 provides examples: both patterns have a topological dimension of two, and both have a fractal dimension somewhere between one and two, yet they have markedly different complexities. It can be seen that Figure 4.1b displays a significantly more detailed pattern composition and I remark that it fills more of the two-dimensional space it is embedded within. The pattern in Figure 4.1a leaves a lot of empty space because the pattern is not "space-filling" in nature. It therefore has a low fractal dimension. Another perspective is to consider how they compare to a one-dimensional straight line. The pattern in Figure 4.1a has a *fractal* dimension which is just above one, at 1.1292; indeed, we can see that the shape is effectively a line with a small amount of extra ruggedness or detail. Figure 4.1b, contrarily, has a fractal dimension just under two (1.7712). This pattern appears to have complicated shapes cut out of it and it fills much of the space with its convolution and detail.



(A) Gosper Island: FD 1.1292 (B) Hexaflake: FD 1.7712

FIGURE 4.1: Two patterns with the same topological dimension but different *fractal* dimensions

The advent of complex networks as a stand-alone field has brought a wealth of innovation, including algorithms for calculating the fractal dimension of a complex network. One of these is called a box counting algorithm [173] and the aim of the process is to describe a network using as few "boxes" as possible, with each "box" containing nodes which are within m network edges of each other. The parameter m corresponds to the length of measure used in the equation to obtain fractal dimension and the extent of detail in the pattern is taken as the number of "boxes" required to cover the whole network when using the measuring scale m.

Because "box counting" is agnostic of the semantics of the network which is supplied, it can be used to calculate the fractal dimension of a local optima network. However, in a LON, distance between nodes is not the only encoded information. Node fitness, as well as edge distance, is of great significance in a LON. Accordingly, a modification of the "box counting" algorithm would be desirable. I implement this as a constraint for the "boxing" of nodes: the fitness distance between two nodes should not be more than a specified threshold ϵ. In this way, only nodes which satisfy both distance and fitness "similarity" criteria can be "boxed". This will, in turn, affect the eventual computed fractal dimension for the network.

This Chapter is a proof-of-concept concerning the use of fractal analysis on the local optima level in a fitness landscape. I compute the fractal dimension and associated metrics for the LONs of an NK Landscape instance set. The obtained results suggest a parallel between the fractal geometry in the LONs and metaheuristic algorithm performance on the associated NK Landscapes.

4.3 Background

4.3.1 Fractal Dimension

The experiments in this Chapter aims to delineate a relationship between fractal detail in LONs and metaheuristic performance on the underlying problem instances. As stipulated in Section 2.6, fractal dimension is a spatial complexity index capturing the ratio between detail observed in a pattern and the scale of measurement [172]. To compute the fractal dimension we have the equation

$$scale^{fractal\ dimension} = detail \tag{4.1}$$

which can be rearranged to obtain the ratio between *detail* in the pattern and *scale* of measurement used, i.e. the fractal dimension

$$fractal\ dimension = \frac{ln(detail)}{ln(scale)} \tag{4.2}$$

4.3.2 Fractal Complexity in Fitness Landscapes

I mentioned previously that Weinberger and Stadler [144] discovered fractal geometry within certain fitness landscapes. They used the well-known *autocorrelation* metric [29] in their fractal analysis and investigated the way it scaled alongside increasing landscape diameter. Several years later Locatelli formulated a fitness landscape paradigm [122] concerning multiple layers of abstraction. They termed this the "multi-level" structure of optimisation problems and noted that it could be exploited. Viewing landscapes in this way puts them under a fractal lens. Zelinka *et al.* demonstrated the potential of using fractal analysis for learning more about the nature of fitness landscapes [174], focusing on low-dimensional and continuous spaces. Fractal patterns within LONs, however, have not previously been investigated. A consideration for pursuit of this path is how precisely to define the fractal dimension: a complex network such as a LON is quite different to topologically two-dimensional images which are frequently used in fractal analysis. Methods have been proposed for calculating fractal dimension in the specific case of a network. In this Chapter a "box-counting" algorithm is used and extended to cater for the specific semantics of a LON.

4.3.3 Fractal Complexity in Complex Networks

In this Chapter I use a box counting algorithm [173] to compute and define the fractal dimension. This "boxes" together nodes which are within m network edges of each other, aiming to describe the network using as few "boxes" as possible. The parameter m is the scale of measurement used and is therefore involved in the calculation to obtain fractal dimension alongside the number of "boxes" required to cover the network, which is the amount of detail observed.

In the specific case of a LON, edge distance need not be the sole consideration when "boxing" together nodes as merged units of detail. Crucial information about landscape structure is encoded in the network as local optima fitnesses. "Box counting" while blind to this would ignore fitness differences and may "box" together two nodes which are connected in the LON but are not sufficiently "similar" to be considered as a merged unit of detail. I therefore propose an extension to the algorithm, in that nodes should be "boxed" together if they satisfy both a distance *and* a fitness condition. More specifically, two nodes LO_i and LO_j can be "boxed" together if they are separated by less than m edges and if their absolute fitness difference $|f(LO_i) - f(LO_j)| < \epsilon$, where ϵ is the maximum allowed fitness distance and $f(LO_i)$ is the fitness of local optimum LO_i. Intuitively, the effect of ϵ should be that it reduces the amount of boxing (because the criteria are stricter) and therefore raise the fractal dimension which is calculated. Stricter (lower) values of ϵ only allow boxing of nodes which are similar in fitness, and would presumably result in high dimensions. The impact of the parameter m will be investigated later.

Pseudocode for box counting a LON is shown in Algorithm 1. The notation `MASS`(n) represents the quantity of nodes which can be "boxed" using the vertex (local optimum) n as a reference point; `DISTANCE`(LO_i, LO_j) calculates the distance in number of LON edges between nodes LO_i and LO_j; and `DIFFERENCE`($f(LO_i)$, $f(LO_j)$) produces the absolute fitness difference between LO_i and LO_j.

In stage one of the procedure, "centre" nodes are initially identified as those which are the best connected in the network; see lines 7-8. Nodes which are at a distance of no more than m edges and whose fitness difference with respect to the centre node is less than ϵ are then marked as "covered" and are added to the "box" associated with the centre, as is seen at the *if* statement execution block beginning at line 12. Notice from the termination condition for this stage at line 14 that the process continues until all nodes are either "covered" (i.e. belong to a "box") or they are centre nodes. That means wherever a node cannot be "boxed" with any of the centres, it becomes a centre itself and may be the solitary member of a "box" at this stage. In stage two observe at lines 18-21 that for all nodes the central distances are calculated; this is — for each node — the closest distance to a centre. Following that, at lines 25-26, the identity of each non-centre node is switched to that of a neighbour which is closer to a centre node. The original node is removed.

When the algorithm completes the fractal dimension is taken as the ratio between detail and the scale of measurement i.e. the number of "boxes" required and the distance threshold m, respectively. The number of boxes needed is taken as proportion of the network: bp. I insert bp and m as numerator and denominator into Equation 4.2 to compute fractal dimension for the LON.

4.4 Experimental Setting

4.4.1 Test Problem

I use a benchmark combinatorial optimisation problem in this Chapter: NK Landscapes [175]. The instances are from previous literature [11] and are deliberately small in size, such that a full enumeration of the local optima is possible. This is particularly necessary due to the introductory and exploratory nature of this study into fractal analysis of LONs. Here I use $N = 18$ and $K \in [2, 4, 6, 8]$ with 30 instances for each K, giving a total of 120. The values for K are selected so that the emergent landscapes are not excessively rugged and therefore too unstructured for analysis (a higher K produces a more rugged topology).

Algorithm 1 LON Fractal Analysis

Input: LON, ϵ, m
Output: number of boxes required

Initialisation:
$centre.nodes \leftarrow \emptyset$, $noncentre.nodes \leftarrow all.nodes$
$covered.nodes \leftarrow \emptyset$, $uncovered.nodes \leftarrow all.nodes$
Stage one:
repeat
  for n in $noncentre.nodes$ **do**
    $\texttt{MASS}(n) \leftarrow \texttt{COUNT}(n' \in uncovered.nodes$ where $\texttt{DISTANCE}(n, n') < m$ and $\texttt{DIFFERENCE}(f(n), f(n')) < \epsilon)$
  $next.centre \leftarrow n$ where $\texttt{MASS}(n) == \texttt{MAXIMUM}(\texttt{MASS}(n \in noncentre.nodes))$
  for n in $uncovered.nodes$ **do**
    $distance.to.centre \leftarrow \texttt{DISTANCE}(next.centre, n)$
    $e = \texttt{DIFFERENCE}(f(next.centre), f(n))$
    if $distance.to.centre < m$ and $e < \epsilon$ **then**:
      $uncovered.nodes \leftarrow uncovered.nodes - n$
until $\forall n \in all.nodes : n \in covered.nodes$ or $n \in centre.nodes$
Stage two:
for n in $all.nodes$ **do**
  for c in $centre.nodes$ **do**
    $pairwise.distance \leftarrow \texttt{DISTANCE}(n, c)$, $e \leftarrow \texttt{DIFFERENCE}(f(n), f(c))$
    if $pairwise.distance < lowest$ and $pairwise.distance < m$ and $e < \epsilon$ **then**:
      $closest.centre \leftarrow c$
      $lowest \leftarrow pairwise.distance$
  $closest.centres[n] \leftarrow lowest$
$uncovered.nodes \leftarrow uncovered.nodes$ according to $closest.centres$ (ascending)
for n in $noncentre.nodes$ **do**
  $n' \leftarrow$ neighbour of n with lower value in $closest.centres$
  $id_n \leftarrow id_{n'}$
  remove n from $noncentre.nodes$

4.4.2 Metaheuristic Algorithms

Examining a possible relationship between fractal dimension in a LON and search difficulty on the underlying problem is essential in this study. I deploy trajectory-based metaheuristic algorithms on the NK instances to obtain difficulty information; these are Iterated Local Search (ILS), Simulated Annealing (SA), and Tabu Search (TS). For the local search in the three algorithms, a single bit-flip is the search operation. The local search is best-improvement for ILS and TS; for SA, this is first-improvement. The ILS has a perturbation operator of two bit-flips. The SA considers a standard exponential cooling scheme; parameters are those suggested in a previous study [176] where they were tuned based on preliminary runs observation. The start and end temperatures are 1.4 and 0.0, respectively; α is set at 0.8; and the maximum iterations at the same temperature are 262. The length of the tail in the tabu search is set at N, i.e. 18, which is the length of the solutions. All three algorithms were implemented using *Paradiseo* [177], which is an open-source package in C++, and they were executed 1000 times on each NK instance.

4.4.3 Fractal Analysis

To calculate fractal dimension for the LONs I employ Algorithm 1, a box counting algorithm from the literature [173] which I specialised for the semantics of a LON. An implementation for generic "box counting" of a complex network in C was obtained from the original work in Song *et al.* [173] and then extended.

As stipulated in Section 4.3.3 two important parameters affect the computed fractal dimension for a LON. The parameter m, used in the original algorithm, specifies maximum edge distance between "boxed" nodes. This is initially set at two and is increased in step sizes of one if nodes remain uncovered during stage two of the algorithm; this design choice is made by the authors of the original box counting procedure [173]. There is also the maximum fitness difference between "boxed" nodes, ϵ. The value for ϵ should be suitable for all LONs regardless of the fitness value range. I therefore standardise the fitness values, which are provided as part of the LONs used and are computed over the whole solution space (exhaustive enumeration), as $\hat{f\prime} = \frac{f\prime - E(f\prime)}{sd(f\prime)}$, where $E(f\prime)$ is the expected fitness value and $sd(f\prime)$ is the standard deviation. In this way the mean becomes zero while the standard deviation is one. For the experimentation I use values for ϵ which are in the range $[0.0, 1.0]$ in step sizes of 0.05. I selected this range for its breadth, in order to account for the unknown nature of using fitness distances in LONs for computing fractal dimension.

4.4.4 Features

Metaheuristic performance metrics. Metaheuristic performance can be measured in a number of ways. I aim to concisely capture speed and effectiveness of the algorithms described in Section 4.4.2 on the NK problems under consideration. To this end I divide the number of runs which successfully reach the global optimum after 26214 fitness function evaluations (this is 10% of the search space) by the total number of runs (1000 in this study). In subsequent text I refer to this performance metric as $algorithm.s$, which is the success rate where $algorithm$ is the associated metaheuristic. To measure speed of convergence I consider the number of evaluations used in the duration of any runs which obtained the optimal fitness within 26214 fitness function evaluations — this performance metric is hereafter referred to as $algorithm.t$.

Fractal dimension features. Although ϵ in the range $[0.0, 1.0]$ in step sizes of 0.05 is involved, for the purposes of relating fractal dimensions to performance I focus on four representative fractal dimension features: $\epsilon \in [0.25, 0.50, 0.75, 1.00]$, with respect to standardised fitness ranges. A lower value for ϵ infers stricter conditions during box counting: remember that, to consider nodes as a single unit of detail, they must satisfy $\texttt{DISTANCE}(LO_i, LO_j) < m$ and $|f(LO_i) - f(LO_j)| < \epsilon$. If ϵ is nearer to 1.0 then the constraint for boxing nodes is more lenient and a lower number of boxes will be required to cover the network. That will affect the computed fractal dimension because the number of "boxes" serve as the extent of detail measured in the pattern (see Equation 4.2).

Other landscape features. I additionally include two fitness landscape features which do not relate to the fractal dimension of LONs in the analysis. These are the number of local optima and the number of *funnels*, that is, basins of attraction at the level of local optima [56]. Both of these features have been linked to metaheuristic performance in previous literature [21, 167].

4.4.5 Regression Model Setup

I build algorithm performance models using LON features for predictors and the performance of metaheuristic algorithms as the response variables. The aim is the elucidation of how LON features can contribute to explaining or predicting algorithm proficiency, paying particular attention to the fractal nature of the LON. In pursuit of that I conduct linear and random forest regressions. I use *bootstrapping* (random repeated sub-sampling cross-validation) to estimate the sampling distribution for model statistics. The models are bootstrapped for 100 iterations. Pattengale *et al.* found that between 100 - 500 bootstrapping replicates provide the necessary information [178]; I use 100 for computational efficiency, after noticing that separate bootstrapping runs produce similar estimates and that increased iterations have a diminutive effect on the calculated estimates. A training-test split of 80 - 20 is applied. The random forest regression uses 500 trees. Predictors are standardised (due to different value ranges) as follows: $\hat{pre} = \frac{(pre - E(pre))}{sd(pre)}$, with pre being the predictor in question and where $E(pre)$ is the expected predictor value and $sd(pre)$ is the standard deviation. The model statistics I focus on are R^2, which captures the amount of variation in the response variable which can be explained using the predictor set, and *mean squared error*, which expresses the mean squared difference between the model-estimated values and the actual values.

4.5 Results

I compute the fractal dimension of the LONs, which themselves were extracted from 120 NK Landscapes. This is done using the modified "box counting" algorithm outlined in Section 4.4.3 and represented in Algorithm 1. For each LON, 20 fractal dimensions are extracted — differentiated by the setting of ϵ.

The three optimisation algorithms described in Section 4.4.2 are executed on each NK Landscape. With the fractal complexity information and the performance data I can proceed to examine potential relationships between them. The essential aim of experiments in this Chapter is to ascertain whether search performance on the underlying optimisation problems can be explained using fractal complexity in the LONs.

4.5.1 Fractal Complexity and Epistasis

To start with I investigate the distribution of fractal dimensions. The problem instances can be split by their *epistasis* value, K. Figure 4.2 shows the distributions for instances from each of the four K settings.

We can see from the Figure that the fractal dimensions increase with the epistatic parameter K, which is known to increase ruggedness and randomness in NK Landscapes. Consider, for example, the interquartile range of category $K2$ against that of category $K4$. Fractal dimension increasing with ruggedness suggests that more ordered and predictable landscapes contain local optima connectivity which is of lower fractal dimension (i.e. lower spatial complexity).

The interquartile range for the $K2$ LONs spans $\sim$1.008 to $\sim$1.379. The median $K2$ LON fractal dimension is 1.260, meaning that these are patterns which fill space in a way that is different to one-dimensional or two-dimensional patterns. The fractal dimension is just above one, implying a structure which fills space somewhat similarly to a line except with some added detail or convolution. In our context the patterns under study represent search connectivity between local optima. A median fractal

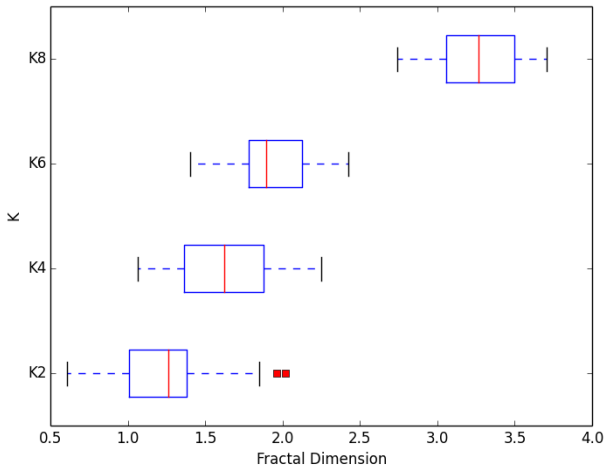


FIGURE 4.2: Distributions for fractal dimension (ϵ = 1.0) of the local optima networks, grouped by epistasis level; the LONs associated with 30 instances from each group are considered for $K \in [2, 4, 6, 8]$, all with $N = 18$

dimension of 1.260 indicates that the *K2* LONs contain somewhat linear sequences of local optima with some deviation or "spokes" branching away from the main path.

The *K4* LONs generally have a higher fractal dimension, with the interquartile range spanning ∼1.363 to ∼1.878. This fact asserts that the LON structural detail fills space in a manner somewhere between the space-filling behaviour of one-dimensional and two-dimensional objects. The dimensions are closer to two than the *K2* counterparts; the median is 1.623. The pattern in Figure 4.3 has fractal dimension ∼1.585. The topology of a LON with this dimension is unlikely to be equivalent to this pattern; however, the scaling of spatial complexity is comparable between the two. A fractal dimension of around 1.585 stipulates that using a scale of measurement equating to one-fourth of the size of the pattern (i.e. a *scaling factor* of four: m=4) produces nine units of detail. Indeed, in Figure 4.3 using a length scale one-quarter of a side of the triangle measures nine smaller triangles. Inserting the values into Equation 4.2 and solving $4^{fractal\ dimension} = 9$ does not result in an integer exponent but instead *fractal dimension*=∼1.585.

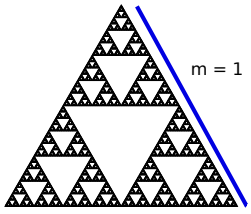


FIGURE 4.3: Sierpinski triangle; a pattern with fractal dimension ∼1.585

Let us consider again Figure 4.2 and concentrate on the *K6* values. The interquartile range comprises ∼1.779 to ∼2.126 with a median value of ∼1.893. That postulates the situation where local optima connectivity patterns resemble either a highly spatially-convoluted pathway (for values such as 1.779) or a few interconnected

pathways with quite a simple layout (for values such as 2.126).

Notice that the increase in dimension is stark between the *K6* and *K8* groups. While the majority of fractal dimensions in the lower-epistasis LONs (*K2* and *K4*) were between one and two, these highly rugged *NK* Landscapes appear to give rise to LONs with fractal dimension mostly between three and four. This situation is difficult to visualise or conceptualise.

4.5.2 Fractal Complexity and Metaheuristic Performance

I now investigate the parallel between fractal complexity in LONs and metaheuristic performance on the associated *NK* problems.

Correlation Analysis

An intuitive foundation for contrasting features with metaheuristic performance is correlation analysis. I compute the Spearman correlation coefficient and the corresponding p-value — these were detailed in Section 2.7 — for pairwise combinations of the observed fractal dimension features and the performance measures. Figure 4.4 shows the correlation matrix for a set of features of the 120 considered problems and LONs. I show the correlation coefficient between variables (upper triangle of plot); density plots (middle diagonal); and scatter-plots (lower triangle). In the case of the density and scatter-plots, difference in colour indicates value for K as outlined in the caption. The size of the text is proportional to the strength of the absolute value of the correlation. An indication of p-value level is given by the asterisk, as described in the caption. All measures of metaheuristic performance are included in the variable set. As asserted in Section 4.4.4 there are two metrics each for ILS, SA and TS, giving six performance variables. Information about each variable is given in the abbreviated name: for example, *ILS.s* is the success rate of ILS, while *ILS.t* is the speed. The other variables included are fitness landscape features. The principal focus lies in those concerning fractal complexity in the LONs. In the Figure, the fractal dimensions involved are defined with ϵ set at 0.25, 0.50, 0.75, and 1.00. In addition to correlations between fractal dimensions and performance metrics, I additionally show correlations for other fitness landscape features: the number of local optima (labelled as *optima* in the Figure) and the number of landscape funnels (*funnels*).

Surveying the correlation matrix, the effect of the parameter ϵ – the maximum fitness difference allowable for boxing nodes together — on the resultant FD can be seen when comparing the correlations between *FD.25*, *FD.100* and the six performance metrics. The general trend (although *ILS.t* is an exception) is that the correlations are slightly stronger in the case of *FD.25*; with variables *SA.s* and *TS.s* the *FD.25* dimensions also have a lower p-value level than the *FD.100* dimensions. All four FD metrics display at least four correlations with the six considered performance metrics which have associated $^*p < 0.05$ or lower. These provide pieces of evidence that the correlations are statistically valid: such p-values indicate that the probability of the set of observations arising by chance (i.e. if there is no relationship between the variables) is less than 5%, which is the generally-accepted threshold for significance [179]. Looking at the algorithm speed rows (*ILS.t*, *SA.t* and *TS.t*), we can see that the LON fractal dimensions show a positive correlation. The correlations are weak in the case of *ILS.s*; moderate in the case of *TS.t*; and strong for *SA.t*. All have $^*p < 0.05$ or lower. For SA and TS, there are weak positive correlations between their success and fractal dimension metrics — although these are noticeably weaker than with

respect to runtime. In the case of TS, all four fractal dimension features have correlations with success with $^{*}p < 0.05$; for SA, only the dimensions associated with the strictest ϵ (*FD.25*) have this level of p-value.

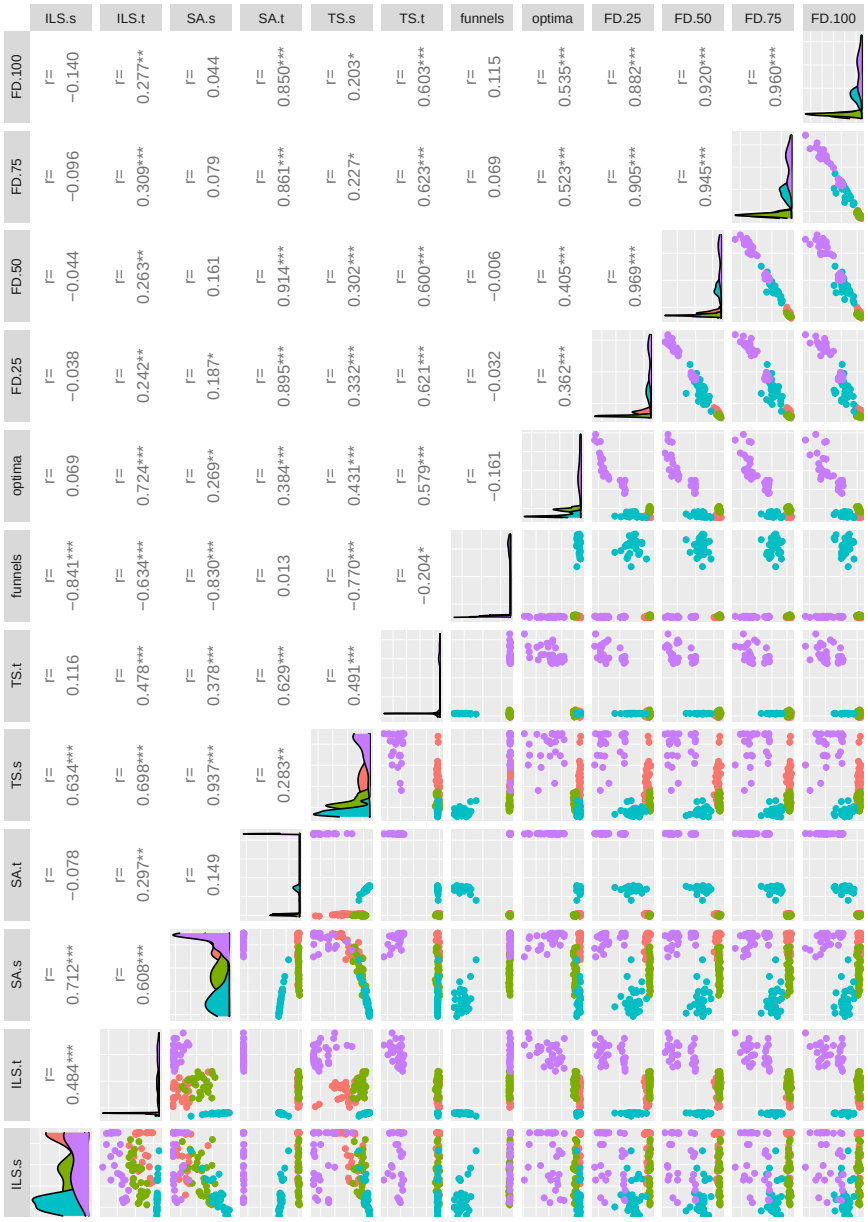


FIGURE 4.4: Correlation matrix for performance metrics and landscape features (see facet titles). Lower triangle: pairwise scatter plots. Diagonal: density plots. Upper triangle: pairwise Spearman's rank correlation, $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$. Colour represents instances split into different levels of epistasis ($K \in [2, 4, 6, 8]$)

Regression Models

In this Section I complement the correlation analysis with multiple linear regression models which include LON fractal dimensions as predictors and metaheuristic performance metrics as response variables. The predictors used are introduced in Table 4.1; Tables 4.1 and 4.2 summarise six models each: two types of regression $\times$ three

metaheuristics. Provided are the R^2 and mean squared error (recall Section 2.7) value for the model.

TABLE 4.1: Predictor variables used in the regression models

feature	description
optima	number of local optima
funnels	Number of funnels
fractal dimension^1	FD with fitness difference threshold ϵ set at 0.25
fractal dimension^2	FD with fitness difference threshold ϵ set at 0.50
fractal dimension^3	FD with fitness difference threshold ϵ set at 0.75
fractal dimension^4	FD with fitness difference threshold ϵ set at 1.00

TABLE 4.2: Summary statistics estimated with bootstrapping for explaining the hit-rate performance of ILS, SA, and TS. Predictors include deterministic fractal dimension LON statistics, as well as other landscape features

type of regression	response variable	R^2	mean squared error
linear	*ILS.s*	0.377	0.066
random forest	*ILS.s*	0.654	0.035
linear	*SA.s*	0.712	0.035
random forest	*SA.s*	0.805	0.021
linear	*TS.s*	0.724	0.025
random forest	*TS.s*	0.823	0.016

In Table 4.2, notice that the random forest models are associated with a higher R^2 and lower MSE than their linear equivalents. This implies they are higher quality models and that the features can explain more of the variance under the random forest setting.

The random forest models with *SA.s* and *TS.s* are a particularly good fit, with approximately 80% and 82% of variance in the dependent variables explained. Linear regression does not capture the relationships well where *ILS.s* is the response — this set-up yields the weakest R^2 value at 0.377. This is additionally associated with the highest MSE, indicating a comparatively low model quality.

Table 4.3 summarises regression models in the same format as Table 4.2. In these, metaheuristic speed of convergence serves as the dependent variable instead of success rate.

An immediately apparent observation about the model statistics in this Table is that every R^2 value is high — over 86% in all cases — indicating that a large proportion of the variance in the response variables is explainable using the input features. This is encouraging with respect to the utility of using LON fractal dimension features in models to explain and predict metaheuristic performance on the underlying problem. The values are higher than those associated with the *algorithm.s* variables in Table 4.2, which implies that this set of features is more proficient in explanation of variation in speed of convergence of the metaheuristics, when compared to success rates.

In terms of the MSE values, the lowest is associated with the random forest model with *TS.t* as the dependent variable; the highest is attached to the linear model which uses *SA.t* as the response. The two *TS.t* models have the lowest MSE estimates out of the three algorithms. This perhaps hints that the set of features is slightly more

TABLE 4.3: Summary statistics estimated with bootstrapping for explaining the runtime performance of ILS, SA, and TS. The response variables are log-transformed for ease of interpretability for the MSE values. Predictors include deterministic fractal dimension LON statistics, as well as other landscape features

type of regression	response variable	R^2	mean squared error
linear	*ILS.t*	0.861	0.109
random forest	*ILS.t*	0.878	0.071
linear	*SA.t*	0.870	0.223
random forest	*SA.t*	0.983	0.015
linear	*TS.t*	0.903	0.043
random forest	*TS.t*	0.962	0.012

consistent in explaining runtime variance for the TS than ILS and SA under these experimental conditions.

TABLE 4.4: Variable importance rankings, bootstrapped, for each of the six random forest models. Columns are labelled with the model response variable

feature	*ILS.s*	*SA.s*	*TS.s*	*ILS.t*	*SA.t*	*TS.t*
fractal dimension^1	4	3	**2**	2	2	2
fractal dimension^2	**2**	6	4	**5**	**3**	3
fractal dimension^3	3	5	5	3	4	**4**
fractal dimension^4	5	4	6	6	5	5
optima	6	2	1	1	1	1
funnels	1	1	3	4	6	6

Table 4.4 indicates the variable importance rankings for the considered random forest regression models. To compute the importance of a variable, the reduction in decision tree node impurities when splitting on the variable is noted and these are averaged over all 500 trees used in the regression. Node impurities are measured with the residual sum of squares. In the Table, each column represents a model, and is labelled with the response variable.

Surveying the Table, notice that *funnels* usually ranks first where success rate is concerned; its key role as a predictor of hit-rate is evident. The number of local optima ranks highest for *TS.s* and for all three runtime response variables (*ILS.t*, *SA.t*, and *TS.t*). Fractal dimension features rank higher than the number of local optima for *ILS.s*, and rank higher than the number of funnels for *TS.s*, *ILS.t*, *SA.t*, and *TS.t*; especially important is *fractal dimension*^{1}, which is the most strict ϵ setting. For *SA.t* and *TS.t*, as ϵ is increased, the importance of the computed fractal dimensions decreases.

Fractal dimension^4 never places higher than fourth and is usually last or second-last. This is the fractal dimension which is produced using the ordinary box counting from the literature (that is, the variant which does not involve fitness distances). In contract, *fractal dimension*^{1}, which has the most strict fitness threshold setting, places second on four occasions.

4.6 Discussion

4.6.1 Fractal Complexity in Local Optima Networks

In Section 4.5.1 we saw that the instances under study lend to LONs with varying fractal dimension. Recall Section 2.6 for how to interpret fractal dimensions. The LONs which were extracted from instances with the lowest level of epistasis (and therefore landscape ruggedness) exhibited a fractal dimension interquartile range of $\sim$1.008 to $\sim$1.379, meaning that most of the LONs fill space in a manner in-between the behaviour of one-dimensional and two-dimensional shapes. A fractal dimension just above one implies a linear structure with some additional detail or convolution, in that the scaling of detail in the pattern is not equivalent to that of a one-dimensional line. My interpretation of this for LONs is that they (that is, the local optima) comprise a somewhat linear sequence or main pathway of local optima with a small amount of deviation or detail aside from the main path.

LONs derived from $K \in [4, 6]$ instances generally had higher fractal dimension. Some of the fractal dimensions indicated that the LONs are spatially complex, for example where FD = 1.89. I argue that a value such as this stipulates a LON composed of winding and convoluted sequences with many complicated spokes leading off the main path. Other local optima networks had a calculated fractal dimension lying just above two; this infers a low-complexity pattern of geometry. In the context of LONs, I stipulate that this means a low number of paths of local optima with an insignificant amount of non-regular detail.

LONs extracted from the most rugged NK instances ($K = 8$) cannot be properly conceptualised in terms of their fractal dimension because the majority of them had FD ≥ 3. Most were just above three, indicating the detail within the LONs scales in a form comparable to that of a three-dimensional object. It is possible that the three dimensions represent connecting or intersecting paths of local optima.

4.7 Conclusions and Future Work

In this Chapter, an empirical and introductory study on fractal analysis of local optima networks — and how the fractal complexity relates to metaheuristic search — has been conducted. A benchmark combinatorial problem, NK Landscapes, was used as a case study. Various ruggedness levels were considered in the instance set.

Linear and random forest regression analysis revealed that combining fractal dimension features with the number of local optima and the number of *funnels* to serve as predictors results in extremely strong models — in particular for the explanation of metaheuristic runtime, although also when random forest is used in conjunction with SA or TS success rate as the dependent variable. In the case of the success rate models, random forest produced stronger and higher-quality models than linear regression and I hypothesise that this is due to attendant non-linear relationships.

Variable importance rankings for the random forest regression showed that, in many cases, fractal dimension features rank higher than the number of funnels — communicating the fact that they contributed more information than lower-ranked variables. This was especially true when fractal dimensions were computed in a manner specific to LONs, that is, with the involvement of fitness distances in the calculations. The most strict fitness threshold produces LON fractal dimensions which ranked as second most-important predictor in four out of six model setups. Fractal dimensions which were obtained through ordinary complex network box counting

ranked lower — never higher than fourth place out of six predictors. This discrepancy between the two shows that the proposed approach of considering fitness has value. The number of funnels present in the LON was salient as a predictor of success rates; the number of local optima was more important for algorithm runtimes.

While the focus was on small problems here there is no reason that fractal analysis could not be deployed on sampled LONs. With this new insight into the "middle" layer in the fitness landscape (i.e. the space of local optima) I can proceed further down this avenue of possibility. The traditional (agnostic of the semantics of the network) fractal analysis captures a unique element of a LON; a phenomenon which is linked to slower speed but raised effectiveness in search algorithms, and which I argue merits further investigation. Furthermore, the addition of the fitness element for "box counting" to cater for the special case of a LON gives valuable insight into the importance of local optima connectivity and their fitness distribution.

4.8 Summary

This Chapter provided a first fractal analysis of LONs, but questions remain. One consideration is how to intelligently calculate and define fractal dimension for a LON. The analysis so far used a generic "box counting" from complex network literature as well as proposing a LON-specific extension which includes fitness distance into the "boxing" calculations. There is no reason to assume, however, that LONs are "mono-fractal" systems, as they have been assumed to be in this Chapter. Some complex networks have presented as "multi-fractal" systems [180] and multi-fractal analysis is necessary to investigate this. In addition, LON edge weights encode crucial search dynamic information. They capture probability of connectivity between local optima and as such can reveal heavily-traversed paths and lightly-traversed paths through the landscape. Edge weights could be considered when during calculation of fractal dimension for a LON. The upcoming Chapter pursues this and applies multi-fractal analysis to LON systems as well. Additionally, the upcoming analysis considers sampled LONs together with exhaustively-enumerated ones.

Chapter 5

Multifractality and Probability in Local Optima Networks

The previous Chapter detailed a preliminary study on fractal analysis for local optima networks. What follows now is an investigation concentrating on how to define fractal dimension for these structures.

5.1 Abstract

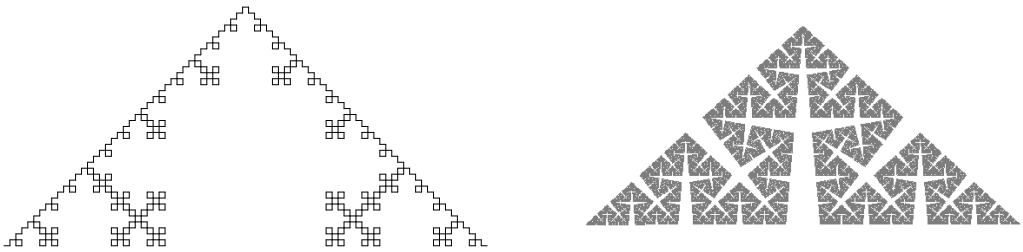
This instalment of work views fitness landscapes through a fractal geometry lens by calculating the fractal dimensions of LONs. The fractal dimension is a spatial complexity index which can assign a non-integer dimension to an object; I propose a fine-grained approach to obtaining the FD of LONs using metaheuristic path probability information which is encoded in LON edge weights.

For complex systems such as LONs, the fractal dimension may be different between sub-systems and multifractal analysis is needed. In this Chapter multifractal calculations are applied to LONs for the first time and a comparison with monofractal dimensions is conducted. I focus on the QAP, bringing fractal analysis to *sampled* LONs of moderate size. Fully enumerated LONs of smaller size are additionally included. The results show that local optima spaces can be multifractal and that valuable information regarding probabilistic spatial complexity is encoded in the edge weights of local optima networks. Pairwise relationships are delineated between these phenomena and the performance of competitive metaheuristic algorithms on the associated QAP instances.

5.2 Introduction

Interest is growing in the fractal complexity of networks; indeed, some networks have recently been shown to be *fractal* [181, 182] which means that similar patterns replicate at different scales within the object. The *fractal dimension* can assign a non-integer dimension to a pattern as an index of spatial complexity. It captures the space-filling nature of the pattern and is the ratio between the extent of detail and the scale of measurement.

Figure 5.1 shows two patterns with different fractal dimensions. While Figure 5.1a has dimension 1.465, for Figure 5.1b this is 1.785. We can see the latter is convoluted and complex. The former is more straightforward in the manner that it fills space. If these two structures were abstracted representations of fitness landscapes then Figure 5.1b would intuitively provide more complications during metaheuristic search.



(A) Fractal dimension = 1.465 (B) Fractal dimension = 1.785

FIGURE 5.1: Patterns with fractal dimension between one and two

Assigning a single fractal dimension may not be appropriate for a complex system. Doing so has the underlying assumption that the fractal geometry is approximately uniform (*isotropic*). Isotropy cannot always be assumed, as is stipulated in some literature which proposes *multifractal* analysis for complex networks [182, 180]. Multifractal analysis produces a whole spectrum of fractal dimensions for a single object. Benoit Mandelbrot, the pioneer of fractal geometry, himself argued that a continuous spectrum of dimensions is necessary to properly capture the complicated dynamics of a real-world system [183].

The contributions of this Chapter can be summarised as follows:

1. A first application of fractal analysis to *sampled* LONs
2. New methodology for probabilistic fractal dimensions when studying LONs
3. First multifractal analysis of LONs
4. A comparison of monofractal, multifractal, and probabilistic fractal dimension characterisation for LONs
5. Exploration of the relationship between fractal dimensions and the performance of competitive metaheuristics known in the QAP domain

5.3 Methodology

5.3.1 Box Counting

Many patterns found in nature exhibit a fractal complexity which can be characterised with fractal dimension [183]. As I described in Chapter 4 a proposal for computing fractal dimension for networks is "box counting" [181]. To recap; the algorithm "boxes" together nodes which can be considered as a single unit of detail. They are merged if they are separated by less than m network edges. The parameter m is the scale of measurement and this is used in the fractal dimension calculation. The extent of detail observed is simply the number of "boxes" necessary to fully cover the network using the scale m. Box counting is agnostic of the semantics of the network and does not take into account node attributes. In the previous Chapter, however, I specialised box counting for local optima networks such that fitness difference is considered alongside edge distance: two nodes can be boxed together if the distance between them in the LON $\texttt{DISTANCE}(LO_i, LO_j) < m$ and their absolute fitness difference $|f(LO_i) - f(LO_j)| < \epsilon$. I extend this approach for the fractal analysis algorithms used in this Chapter.

5.3.2 The Multifractal Spectrum

Real-world systems often do not exhibit the spatial homogeneity that traditional monofractal analysis assumes is present [184]. In this case, the extent of fractal geometry cannot be characterised by a single fractal dimension metric, and instead requires a spectrum of numbers. Indeed, pioneering authors in the fractal community have stated that the multifractal approach is necessary for many real-world patterns [183, 185]. In the past few years, studies have surfaced where the concept of *multifractality* has been considered for complex networks [182, 180, 186]. Some of the networks which the authors analysed required a full multifractal dimension spectrum to properly characterise the spatial complexity properties.

An approach to calculating a set of generalised fractal dimensions for a complex network is called the *Sandbox Algorithm* [184] which is related to the "box counting" process which I extended in Chapter 4 and recapped in Section 5.3.1 just above.

I implement the algorithm from the literature [184] and then modify the process to allow for the special case of a local optima network: because LONs are a compression of the fitness landscape, node fitness is taken into account (as was the case in Chapter 4). Pseudocode for the sandbox algorithm, which I have refashioned for the local optima network case, is provided in Algorithm 2. The `DISTANCE`(c, v) function produces the edge distance in the LON between nodes c and v; the `DIFFERENCE`$(f(c), f(v))$ function returns the fitness difference between local optima c and v.

Algorithm 2 LON Multifractal Analysis

Input: *LON*, *q.values*, *radius.values*, *fitness.thresholds*, *number.centres*
Output: mean sandbox size

1: Initialisation:
2: *centre.nodes* $\leftarrow \emptyset$, *noncentre.nodes* $\leftarrow$ *all.nodes*
3: *mean.sandbox.sizes* $\leftarrow \emptyset$
4: **for** *q* in *q.values* **do**
5: **for** *rd* in *radius.values* **do**
6: **for** ϵ in *fitness.thresholds* **do**
7: *centre.nodes* $\leftarrow$ `RANDOM.SELECTION`(*all.nodes*, *number.centres*)
8: *sandbox.sizes* $\leftarrow \emptyset$
9: **for** *c* in *centre.nodes* **do**
10: *number.boxed* $\leftarrow 0$
11: **for** *v* in *all.nodes* **do**
12: $d \leftarrow$ `DISTANCE`(c, v)
13: $j \leftarrow$ `DIFFERENCE`$(f(c), f(v))$
14: **if** ($d == 1$) **OR** ($d == rd - 1$ **and** $j < \epsilon$) **then**:
15: *number.boxed* $\leftarrow$ *number.boxed* + 1
16: *sandbox.sizes* $\leftarrow$ *sandbox.sizes* $\cup$ {[*number.boxed*]}
17: *bs* $\leftarrow$ `MEAN`(*sandbox.sizes*)
18: *mean.sandbox.sizes*[*q*][*rd*][ϵ] $\leftarrow$ *bs*

The process begins at line 7 with *number.centres* nodes being randomly selected as sandbox centres. These are allocated a radius, *rd*. In the algorithm from the literature, nodes which are less than *rd* edges from a centre are counted as inside its sandbox. I change the conditions of inclusion, to account for the semantics of the local optima network. For each sandbox centre, nodes which are a single edge away are always added to the box: this is to allow a certain level of guaranteed boxing movement, irrespective of fitness differences between the local optima. In addition,

nodes which are at a distance of $rd - 1$ from the centre, *and* whose fitness difference with the centre is less than ϵ are added to the sandbox. This is a stricter design than the literature sandbox procedure, and I argue that implementing it emphasises the inherent dimension: less nodes are boxed — but those that are, are genuinely "similar" in either location or fitness. Additionally, this process exploits and respects the information which is encoded in a LON. The decision mechanism for boxing the nodes is seen at lines 12-15 of the Algorithm.

The average sandbox size, *bs*, for *number.centres* central nodes is calculated (see lines 17-18). The value for *bs* is later used in the equation to obtain a fractal dimension. The whole algorithmic process is repeated for various values of ϵ (the fitness difference maximum; line 6), r (the sandbox radius; line 5), and also q, which is from an arbitrary set of numbers: $q \in [q.values]$, at line 4. The parameter q is also used in the computation of fractal dimension — there is a dimension value for each q, rd, ϵ combination — and in this way facilitates the production of a spectrum of dimensions. Both q and rd are part of the algorithm from the literature [184]; ϵ is a novel addition. The output from an algorithm iteration is involved alongside q to generate a fractal dimension:

$$fractal\ dimension = \frac{ln(detail^{q-1})}{(q-1) * ln(scale)} \tag{5.1}$$

where *detail* is the mean sandbox size using radius *rd*: *bs* (as a proportion of the total network size) and *scale* is the radius *rd* as a proportion of the network diameter.

Upon completion of the sandbox algorithm, the product will be a multifractal spectrum of dimensions, each of them calculated according to Equation 5.1.

5.3.3 Probability-based Fractal Dimension

In the preceding Chapter, fractal dimensions were calculated on LONs derived from a set of NK Landscape problems. The box counting algorithm which was used considered edge distance between nodes as the scaling factor for computing fractal dimension. A consideration when using this methodology is that when the network diameter is small, the units of detail observed may be of low resolution; for example, if most nodes are within one or two edges of each other. In pursuit of a higher-resolution view on the scaling behaviour in LONs I propose to use the metaheuristic transition probabilities which are encoded as edge weights as a replacement for edge distance. These are the total number of times the transition is followed as a proportion of the number of times the local optimum at hand is visited. Edge weights in LONs essentially inscribe the probability that a particular search transition between two local optima will be traversed. I propose to modify the ordinary box counting algorithm [181] (within which nodes are boxed if they are separated by less than m network edges) to consider this information alongside fitness details in the following way: two nodes can be boxed together if it is deemed *likely* that the local optima they represent will be connected with search. There must be a single edge between them which has a certain minimum weight. To normalise an edge weight, W, in the LON I simply subtract it from one i.e. the standardised weight $W\prime$ is defined as 1 - W which is the opposite probability. Mathematically, nodes LO_i and LO_j can be "boxed" as a single unit if $W\prime(E_{LO_i,LO_j}) < \beta$, where E_{LO_i,LO_j} is the edge between LO_i and LO_j in the LON. In addition, their fitness values must be sufficiently similar. Pseudocode for this box counting policy is provided in Algorithm 3. The function `PROBABILITY`(n, n') returns the standardised weight of the edge (that is, the opposite probability) between local optima n and n'; `DIFFERENCE`$(f(n), f(n'))$

produces the fitness difference between the two. Contrary to the design in Chapter 4, I compute the fitness differences in this Algorithm as the *logarithmic return* of the two fitnesses — this statistical tool was outlined in Section 2.7.

Algorithm 3 Probability-based LON Fractal Analysis

Input: LON, ϵ, β
Output: number of boxes required

1: Initialisation:
2: $centre.nodes \leftarrow \emptyset$, $noncentre.nodes \leftarrow all.nodes$
3: $covered.nodes \leftarrow \emptyset$, $uncovered.nodes \leftarrow all.nodes$
4: Stage one:
5: **repeat**
6: **for** n in $noncentre.nodes$ **do**
7: $\texttt{MASS}(n) \leftarrow \texttt{COUNT}(n' \in uncovered.nodes$ where $\texttt{PROBABILITY}(n, n') < \beta$ and $\texttt{DIFFERENCE}(f(n), f(n')) < \epsilon)$
8: $next.centre \leftarrow n$ where $\texttt{MASS}(n) == \texttt{MAXIMUM}(\texttt{MASS}(n \in noncentre.nodes))$
9: **for** n in $uncovered.nodes$ **do**
10: $probability.to.centre \leftarrow \texttt{PROBABILITY}(next.centre, n)$
11: $e = \texttt{DIFFERENCE}(f(next.centre), f(n))$
12: **if** $probability.to.centre < \beta$ and $e < \epsilon$ **then**:
13: $uncovered.nodes \leftarrow uncovered.nodes - n$
14: **until** $\forall n \in all.nodes : n \in covered.nodes$ or $n \in centre.nodes$
15: Stage two:
16: **for** n in $uncovered.nodes$ **do**
17: $n' \leftarrow$ neighbour of n: $\texttt{PROBABILITY}(n, n') < \beta$ and $\texttt{DIFFERENCE}(f(n), f(n'))$ and $n' \in covered.nodes$
18: $id_n \leftarrow id_{n'}$
19: remove n from $uncovered.nodes$

Much of the procedure in Algorithm 3 matches the one detailed in Section 4.3.3 previously. The computation of MASS for a node n, however, here involves identifying nodes which are connected to n with an edge of standardised weight $< \beta$ and with a fitness distance to n which is no greater than ϵ — notice this at line 7. Instead of central distances being calculated in Stage 2, uncovered nodes are dealt with as follows: if they have a direct neighbour (where the probability between the two is less than β and their fitness difference is less than ϵ) which has been boxed already, then the node is given the same box membership as that neighbour. This is at lines 15-19. I iterate the Algorithm for a range of β values. At the end of each iteration, the output can be used to compute a fractal dimension for the LON. As with ordinary box counting, the proportional number of boxes required to cover the network — nb — serves as the amount of detail in the equation which produces fractal dimension (Equation 4.2). The nature of the value which serves as the scale of measurement is different, however: the selected value for β (the maximum *opposite* probability or standardised weight of an edge) fulfils this role:

$$fractal\ dimension = \frac{ln(nb)}{ln(\beta)} \tag{5.2}$$

By implementing this box counting variant, I argue that fractal dimensions can be calculated in such a way that the stochastic origins of the information in a LON is respected. In subsequent text, I refer to fractal dimensions obtained using this

method as probabilistic fractal dimensions.

5.4 Experimental Setup

5.4.1 Benchmark Problem

I focus on a benchmark combinatorial domain here: the Quadratic Assignment Problem (QAP). A thorough elucidation of the QAP can be found in Chapter 2. The experiments in this Chapter include benchmark and synthetic instances. Some are from the well-studied QAP Library (QAPLIB) [187]. One of the contributions of this Chapter is the fractal analysis of *sampled* local optima networks. The previous Chapter exclusively considered fully enumerated LONs which is only plausible with diminutive problem sizes. The instances from the QAPLIB used in the present study are larger (between 12 and 28 locations, detailed in Table 5.1). For these a full enumeration is not feasible. I use samples of the LONs instead, provided by authors of previous literature [7]. To obtain the local optima, the following is executed $2000 \times N$ times: during the sampling they begin from random solutions and conduct local search using best-improvement single pairwise exchange; they then attempt to improve the obtained local optimum with best-improvement local search considering either one or two pairwise exchanges.

Then for defining the edges they repeat these steps $20N^2$ times: for each local optimum obtained in the previous procedure, they perturb the solution using a random kick move of three swaps then perform local search again. If the obtained local optimum is also a member of the pre-defined set of optima, the edge weight (adjacency) between the two is incremented. After all runs the edge weight is this count divided by $20N^2$. While the sampling introduces an inevitable bias, the bias is towards the landscape regions which are likely to be encountered by search algorithms.

In addition to the QAPLIB problems I use "structured" instances produced using a generator [188]. The generator produces flow entries that are non-uniform random values. Clusters of points are placed in compact circular areas on the plane; they themselves are enclosed in a large circle. These instances have been called "real-like" because they simulate the structure found in practical manifestations of the QAP.

The synthetic problem instances are fully enumerated, and as such are constrained to a small size of 11. I use 30 local optima networks of this type as made available by the authors of a previous work [5]. To obtain the local optima, the authors conducted best-improvement pairwise exchange local search. Two nodes are joined by an edge if the destination node can connect from the source through two pairwise exchanges (this is the perturbation operator) followed by single-exchange local search.

TABLE 5.1: QAPLIB instances used; numerical elements indicate problem size

class	instance class	instance names
chr	real-world	12 {a-c}, 15 {a-c}, 18 {a-b}, 20 {a-c}, 22 {a-b}
nug	random grid	12, 14, 15, 16 {a-b}, 17, 18, 20, 21, 22, 25, 27, 28

5.4.2 Local Optima Fitness Difference

As mentioned, I use logarithmic returns [189] to standardise local optima fitness values in preparation for calculations used in the fractal analysis algorithms. This is

to allow the threshold ϵ to be independent of any particular fitness distribution. The logarithmic return is computed as follows:

$$fitness\ difference = ln(f(LO_j)/f(LO_i))$$

The resultant value can be directly compared with ϵ, which is set at 0.5 or 1.0 for the incumbent experimentation.

5.4.3 Algorithms

The essence of the experiments can be split largely in two: those relating to *probabilistic* fractal dimension and those concerning *multifractal* dimension spectra. For both associated algorithms I use as a foundation a box counting algorithm — which is written in C — from the literature [181], augmenting and extending it. I additionally include an untouched version of the algorithm in the results for comparison.

Probabilistic Fractal Dimension Analysis. A separate variant of the box counting algorithm is implemented to calculate probabilistic fractal dimensions, as described in Section 5.3.3. Deciding a suitable value for β is important. Here I use a set of values for this parameter: β in the range $[0.90, 0.96]$ in step sizes of 0.02. A lower value for β is a stricter probability condition. I use this range based on observations from preliminary runs — in particular, the distribution of edge weights present in the networks.

Multifractal Dimension Analysis. Using box counting as a foundation I implement the sandbox algorithm for multifractal analysis [184] in C and extend it for a LON. The process is outlined in Algorithm 2 and described in Section 5.3.2. The values for q are those suggested in the literature (although the choice can be arbitrary): -10 to +10 in increments of 1.00; the number of sandbox centres at each iteration is 10 (because of relatively small network sizes); the set of values for rd ranges from two, to the diameter of the network. The fitness distance threshold ϵ, operating on normalised fitness values, is simply 0.5 (strict) or 1.00 (lenient).

Metaheuristics. The features extracted from the local optima space should be coupled with metaheuristic algorithm performance to produce a practical and useful conclusion. Accordingly, I collect algorithm performance information for the underlying instances, selecting two competitive algorithms for the QAP to this end: Robust Tabu Search (ROTS) [190] and Stutzle's improved Iterated Local Search (ILS) [191]. ROTS is a competitive heuristic for the QAP and includes a best-improvement pairwise exchange local search with a variable-length tabu list tail. For each item-location combination, the most recent point in the search when the item was assigned to the location is retained. A potential move is deemed to be "tabu" (not allowed) if both items involved have been assigned to the prospective locations within the last $tabu_c$ cycles. The value for $tabu_c$ is changed randomly, but is always from the range $[0.9n, 1.1n]$. For the ILS algorithm, I use the first-improvement version with a pairwise exchange as the local search and with $3n/4$ exchanges for the perturbation operator, accepting only improving local optima.

5.4.4 Features

Metaheuristic performance metrics. To quantify performance of the metaheuristics I use two metrics. The first is the number of iterations to reach the global optimum when the run was successful, which I refer to as $algorithm.t$ in subsequent text. The termination condition for these runs is, for ILS, either the global optimum being found or alternatively based on iterations without improvement, *iwi* — this is the termination condition the author uses in the original implementation — and is $iwi = 10\,000$ in the experiments here. For ROTS, the termination condition in the author's implementation is when the global optimum is found or alternatively based on the passage of iterations, ip; I set this at $ip = 10\,000$. The second performance metric captures the fitness after 200 (ILS) or 2000 (ROTS) iterations, as a proportion of the optimum fitness ($algorithm.p$ for short); the discrepancy in allowed iterations is was designed due to observed performance difference between the two algorithms in the chosen configurations. The algorithms are deployed 1000 times on each problem instance and the metrics are averaged over these.

Fractal dimension features. I use the fractal dimension output of LON-specialised fractal analysis (involving fitness distances) as a feature named *det1*; the equivalent output from generic box counting on a LON is referred to with the shorthand *det2*. From multifractal dimension spectra, I extract values from arbitrary points on the spectrum and use labels for them in the format *mfx*, where x is a numeric value. In the case of probabilistic fractal dimensions — these are either identified with the configuration for β with which their production is associated, or alternatively with *probabilisticx*, with x being a number. Whenever a numerical label like this is used, the mapping to the actual semantics is specified clearly in the text.

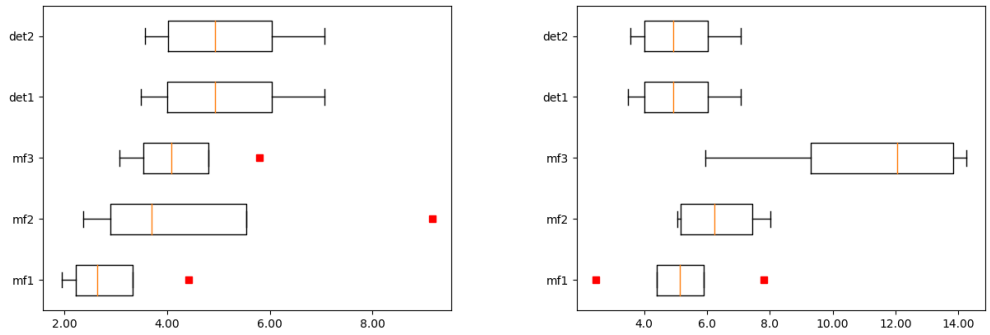
Other landscape features. I additionally include — as a non-fractal landscape feature — the number of local optima in the statistical analysis.

5.5 Results

I calculate the fractal dimensions and associated metrics on a set of LONs extracted from QAP instances. For each problem I direct monofractal dimension analysis, multifractal analysis, and probabilistic fractal dimension analysis. Dimensions from the three are compared with one another to ascertain which are appropriate for LONs. I show the connection between fractal dimensions and search difficulty on the underlying problems.

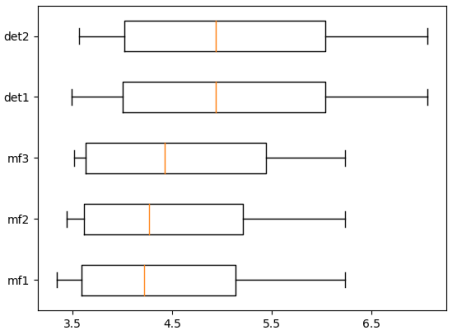
5.5.1 Cardinality of Fractal Dimension

Figure 5.2 shows some of the fractal dimension distributions computed from the LONs. Figure 5.2a represents the "real-like" instances and 5.2b and 5.2c are the *nug* (random grid) and *chr* (real-world) instances from QAPLIB, respectively. Each box represents the spread of dimensions produced using a particular fractal analysis algorithm variant. The upper two boxes illustrate dimensions which were computed using the ordinary box counting algorithm (which generates a single *monofractal* dimension) while the lower three are *multifractal* dimensions for the same LONs, taken from different arbitrary points in the full multifractal spectrum. These are annotated with *mf1*, *mf2* and *mf3*.



(A) random "real-like"

(B) random grid



(C) real-world

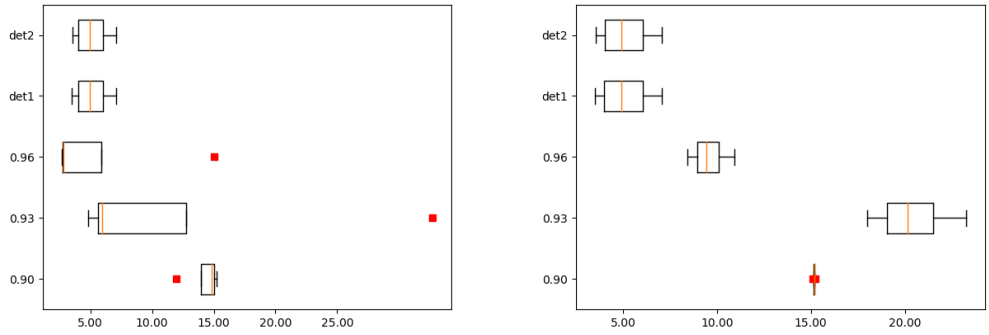
FIGURE 5.2: Monofractal dimension ranges for LONs; *det1* is the fractal dimension where fitness distances have been involved in the boxing procedure, while *det2* reflects the dimensions arising using generic box counting. Multifractal dimension distributions (taken from three arbitrary points on the full spectrum) for LONs form the three lower boxes: *mf1*, *mf2* and *mf3*

Surveying Figure 5.2a we notice that the ranges of the multifractal dimensions (*mf1*, *mf2*, and *mf3*) diverge from the monofractal dimensions (*det1* and *det2*). Generally the multifractal dimensions are smaller. In addition, the three dimension sets extracted from a multifractal spectrum are markedly different to each other. This implies a lack of homogeneity or isotropy with respect to spatial complexity in the LONs, indicating the presence of *multifractal* geometry. A similar phenomenon is attendant in Figure 5.2b, which depicts features of LONs for grid-like problems, although here the dimensions are higher.

The real-world benchmark QAP problems display quite different results: observing Figure 5.2c, we can discern that the multifractal dimensions (boxes *mf1* - *mf3*) are more aligned with the monofractal dimensions in the upper two bars (*det1* and *det2*) than they were in the other two Figures. The lower three boxes also show agreement amongst themselves; this means that taking dimensions as excerpts from different points on the multifractal spectrum resulted in almost the same dimension. Such a similarity infers that these are *monofractal* LONs.

5.5.2 Probabilities in Dimension Calculation

This Section juxtaposes ordinary fractal dimensions against the *probabilistic* fractal dimensions I have proposed. Figures 5.3a and 5.3b show the individual distributions. Figure 5.3a reflects the "real-like" instances; Figure 5.3b is for the QAPLIB instances. Similarly to Figures 5.2a-5.2c, the upper two bars are the values obtained using the established box counting algorithm for complex networks. The lower three consider probabilistic fractal analysis instead and are labelled with the parameter setting used for β (which is the maximum opposite probability i.e. minimum probability that two local optima will be connected during search). Outliers are shown as red squares.



(A) "real-like" (synthetic) (B) benchmark

FIGURE 5.3: Fractal dimensions obtained using standard box counting (upper two boxes): *det1* is the dimension when fitness distance is considered in the calculations while *det2* values are derived using generic box counting for networks. Quantile ranges of the probabilistic fractal dimensions are also shown (the lower three bars: *0.90*, *0.93*, and *0.96*). These are dimensions calculated with β set at 0.90, 0.93 and 0.96, respectively

Figures 5.3a and 5.3b indicate that a routine fractal analysis on a LON might overlook important information relating to *probability* which has been encoded in the network edge weights. Notice that the probabilistic dimensions are usually much higher than the standard dimensions, especially where the benchmark instances are concerned (see Figure 5.3b). That hints at a more complex space-filling behaviour in the network structure than the standard fractal dimensions allude to (observe boxes *det1-2* for comparison). Notice also that dimensions are lower with respect to the stricter box counting constraint of $\beta = 0.90$ than for the more lenient $\beta = 0.93$. I stipulate that this is attributable to pairs of nodes rarely satisfying the criteria and therefore a lack of boxing. A lack of boxing leads to a lower fractal dimension because the number of boxes is the number of units of detail; the units of detail are computed as a ratio of *scale* or resolution in order to produce the fractal dimension.

5.5.3 Correlation Analysis

This Section appraises correlations between LON fractal dimension and metaheuristic search algorithm performance. Figure 5.4 conveys correlations related to multifractal dimension while Figure 5.5 establishes correlations concerning the probabilistic fractal dimension. In the upper triangle of the panel pairwise Spearman correlation coefficients are shown. The correlation over the full LON set is shown in black

text, with a split into *class* of instance marked by use of colour. Red is the LONs derived from QAPLIB (benchmark) instances while green are the "real-like" synthetic ones. Density plots populate the middle diagonal row with scatterplots forming the lower triangles.

From left to right by column, Figure 5.4 includes as features two dimensions taken from different locations on the multifractal spectrum (*mf1* and *mf4)*; ILS performance, denoted as *ILS.t* (i.e. iterations to the global optimum) and *ILS.p* (percentage above the optimum fitness); ROTS performance (*ROTS.t* and *ROTS.p*); the number of local optima (*optima*), and monofractal dimensions (*det1*, calculated using fitness distance during box counting, and *det2*, which are dimensions produced using standard box counting). Figure 5.5 records the same variables with the exception of fractal dimension type. Instead of multifractal dimensions the matrix considers probabilistic dimensions with different values for β: *probabilistic1*, (0.90) *probabilistic2* (0.93) and *probabilistic3* (0.96).

Let us review the correlation matrices in turn. In Figure 5.4 a multifractal dimension feature, *mf1*, has positive correlations of moderate strength with three out of four algorithm performance metrics. Each of these have associated p-value $\leqslant 0.01$, i.e., the probability that these correlations occurred due to chance is less than 1%. The associations can be seen by following along the row labelled *mf1* and checking the intersections with the relevant columns. Notice also that *mf1* has stronger correlations with the four performance metrics than *mono*-fractal dimension features *det1* and *det2* do. The positive associations between multifractal dimension and algorithm performance suggests that this spatial complexity metric is correlated with increased iterations to reach the global optimum and with a larger gap between the obtained fitness and the optimal fitness.

What we notice, however, is that the relationship with algorithm performance is much weaker with respect to the *mf4* variable, which is a dimension excerpt from the same multifractal spectrum as the *mf1* feature. This has two implications: a single fractal dimension is insufficient to characterise the dynamics written in a local optima network (if it was, *mf1* and *mf4* would exhibit similar correlation behaviour), and not all fractal characteristics in the local optima space are consequential. Reviewing the correlation between the two multifractal dimension features, *mf1* and *mf4*, in Figure 5.4 I remark that while they are correlated (and with p-value pointing towards statistical significance) there is a prominent distinction between the benchmark instances and the synthetic "real-like" instances. Notice this by contrasting the correlation in red text (benchmark instances) with those in green (synthetic instances). The strong correlation of dimensions associated with benchmark problems hints that these LONs are *monofractal*, i.e. a more uniform spatial complexity. The dimensions of the generated instances, however, have far weaker correlation. That communicates discrepancies in dimension depending on location in the LON and infers multifractal geometry. This could mean that we must refine the methods for generating instances, such that they better reflect the nature of real-world instances.

Although relatively subtle, observe that *det1* — which is fractal dimension computed according to fitness similarity — is more highly correlated to the performance metrics than the *det2*, which is computed using ordinary box counting.

Redirecting our scrutiny to Figure 5.5, observe that correlations between fractal geometry and algorithm performance are more pronounced for the probabilistic fractal dimensions than with those generated with ordinary box counting. Distinguish this by checking the intersection of *probabilistic1*, for example, against *ILS.t* and *ROTS.t*, and then comparing with the comparable intersections of the standard box counting dimension rows (*det1* and *det2*).

FIGURE 5.4: Correlation matrices for algorithm performance and landscape features, including LON multifractal dimensions (see facet titles). Lower triangle: pairwise scatter plots. Diagonal: density plots. Upper triangle: pairwise Spearman's rank correlation, $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

The probabilistic dimensions exhibit moderate-to-strong correlations with the runtime of metaheuristics with $p < 0.001$ in all cases, suggesting that intricate paths or patterns of local optima — in particular, when those are probable or heavily-traversed search directions — have an correlation with slower performance from metaheuristics. This is more evident for the benchmark instances shown in red than the "real-like" ones in green. There is a patent mismatch between standard fractal dimensions and probabilistic fractal dimensions: these have only a weak-to-moderate correlation. The density plot for *det1* and *probabilistic2* dimensions, as an additional

FIGURE 5.5: Correlation matrices for metaheuristic performance and landscape features, including LON probabilistic fractal dimensions (see facet titles). Lower triangle: pairwise scatter plots. Diagonal: density plots. Upper triangle: pairwise Spearman's rank correlation, $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

example, show unalike distributions. It follows that the two methods provide separate information about fractal topography at the local optima level.

5.6 Conclusion

In this Chapter I brought multifractal analysis to LONs and considered their probability information to help define fractal dimension. I included *sampled* LONs for the first time (extracted from benchmark QAPLIB problems) alongside some *fully enumerated* LONs for smaller problems. Two new approaches for fractal analysis

of LONs were proposed: a fine-grained approach for calculating the probabilistic fractal dimension of a LON, and multifractal analysis. Pairwise relationships were delineated between the fractal dimensions and the performance of two competitive metaheuristic algorithms for the QAP (ILS and ROTS) with correlation analysis. The *probabilistic* fractal dimensions and some of the dimensions taken from the *multifractal* spectrum are correlated to slower runtimes. We saw that the the real-world QAPLIB LONs studied appeared to be monofractal (i.e. able to be characterised with a single fractal dimension), but that "real-like" LONs and grid-based QAPLIB LONs exhibited multifractality. It seems that for some LONs a single fractal dimension is not sufficient to capture the anisotropic spatial complexity encoded in them. Instead a spectrum of dimensions, as I have calculated here, gives more information. Finally, I have shown that probabilistic dimensions are more evidently correlated with search than those derived from standard box counting and may therefore provide a more accurate picture of the complexity in local optima connectivity patterns when taking into account search path probabilities. Further analysis is needed; this is conducted in the next Chapter.

5.7 Summary

Chapters 4 and 5 have indicated relationships between fractal complexity features of LONs and metaheuristic algorithm performance. The problems in Chapter 4 were small in size and were completely enumerated for their local optima. While this was necessary as a proof-of-concept, this size of problem is not realistic. In the present Chapter, larger problem sizes were considered. These were from QAPLIB, and were up to size $N = 28$. This being said, only 25 QAPLIB instances were used, and they were from two of the many classes of problem in QAPLIB. The upcoming Chapter retains the fractal lens on the local optima level and increases the considered QAPLIB instance set from 25 to 85 and the maximum problem size from 28 to 50. In addition, multifractal and probability-based dimensions are used in regression models for performance prediction.

Chapter 6

Fractal Geometry at the Local Optima Level

Chapters 4 and 5 established a foundation for fractal analysis of local optima networks. The present Chapter further develops this branch of knowledge. A larger dataset is included; a recently-proposed LON construction algorithm is used to generate LONs; and enhanced statistical techniques are deployed to ascertain the bind between LON fractal complexity and metaheuristic algorithm performance.

6.1 Introduction

The first study to conduct fractal analysis on fitness landscapes [144] stipulates that for certain problems, landscape ruggedness scales at different levels of abstraction and that this indicates fractal structure. Subsequent studies have reported similar findings [122, 146, 174] and some have emphasised the potential lying dormant in the largely untapped field of fractal analysis for landscapes.

A Local Optima Network (LON) [11] models local optima and their connectivity in a fitness landscape. The nodes of a LON are local optima and the edges are metaheuristic search paths between two local optima under a chosen search operation. As seen in Chapter 3, there is a significant body of evidence suggesting that features of LONs can correlate to, explain, or predict, metaheuristic algorithm performance on the underlying combinatorial problem [4, 5, 8, 52, 56, 142].

Little is known about the fractal complexity in LONs and how their fractal nature relates to metaheuristic algorithm performance. Chapters 4 and 5 indicate that the *fractal dimension* is involved in such a relationship. Nevertheless, Chapter 4 considers only small problem instances (size N = 18 for a binary-encoded problem, NK Landscapes). Chapter 5 centred on the QAP and considers some benchmark instances from QAPLIB [187] up to N = 28, although only two of the library's several instance classes for this problem size range are included. As a consequence, the fractal analysis is conducted on only 25 QAPLIB instances.

I intend to illuminate understanding of the relationships between fractal geometry in LONs and metaheuristic algorithm performance. The QAP serves as a testbed for the analysis in this Chapter and I use QAPLIB problems, increasing the number of instances considered threefold when compared to Chapter 5, and raising the maximum problem size from 28 to 50. A recent and refined LON construction algorithm [24] is used to intelligently build LONs for the QAPLIB instances. Features of the LONs, including fractal dimension features, are computed and the parallel between them and performance is investigated using visual tools, correlation analysis, and linear and random forest regression models.

The contributions of this Chapter can be summarised as follows:

1. A bring new insight into how multifractal geometry at the local optima level can help explain and predict algorithm performance;
2. A significant expansion of the dataset used for fractal analysis in LONs (using more than three times the previous number of QAPLIB instances and raising $N \leqslant 28$ to $N \leqslant 50$, as well as deploying a recent refined and tested sampling algorithm for constructing the LONs);
3. Enhanced statistical techniques for properly validating the use of LON fractal analysis for algorithm explanation and prediction (random forest to model non-linearities; bootstrapping to estimate the sampling distribution of model statistics; using intelligible predictors such as the extent of multifractality and the median fractal dimension).

The structure for the Chapter is this: Section 6.2 contains the necessary background information to render this Chapter self-contained; Section 6.3 details aspects of the methodology used; Section 6.5 specifies the experimental setup, with Section 6.6 presenting the results; finally, Section 6.7 finishes with conclusions and directions for future work.

6.2 Preliminaries

Fractal complexity in local optima networks has been calculated in Chapters 4 and 5 using box counting methods. The box counting algorithm was altered in Chapter 4 to specialise for LONs. For two nodes to be "boxed" as a single "unit" of detail, they must either be a single edge apart *or* they are within m edges of each other, and they also must have a fitness distance less than a set threshold ϵ.

Chapter 5 proposed additional mechanisms for computing and therefore defining the fractal dimension of a LON. A box counting variant which was introduced there utilised LON edge weights during the process. In a LON, edge weights represent the probability that a search path between the connected local optima will be followed. The box counting variant used the probabilities as the scaling factor, in that the criteria for "boxing" is that two nodes have a single edge between them which is weighted with a probability greater than a defined threshold. I refer to values obtained using this method as *probabilistic* fractal dimensions.

In real-world complex systems a single fractal dimension metric can sometimes be insufficient to capture its complexity [192]. Monofractal analysis, such as the box counting described earlier, is based on the assumption that fractal complexity is roughly uniform in the pattern. Some networks have been found to be multifractal [180, 182]. A multifractal algorithm has been used on LONs in Chapter 5 and I deploy this for the experiments involved in this Chapter. The process produces a spectrum of fractal dimensions for a single pattern (which is a LON in this case). Details and pseudo-code for the algorithm are provided later on in Sections 6.4 and 6.5.3.

6.3 Methodology

6.3.1 The Quadratic Assignment Problem

The analysis is conducted on the much-studied QAP [31] which is often used in fitness landscape analysis [5, 30, 32, 33, 34] and has been explained in detail in Section 2.2.

6.3.2 LON Construction: ILS Sampling

LON sampling algorithms are generally augmented on top of an existing optimisation algorithm (see, for example, [19, 24, 34]). I align with this trend here, opting for a recently-introduced construction algorithm which joins an ILS with LON logging for the QAP [24]. This ILS Sampling is executed r times from independent random starting solutions. In the ILS process, the local search is a pairwise exchange of items on the permutation solution, with the perturbation being k pairwise exchanges. Each local optimum encountered during search is stored in the set of nodes alongside its fitness. If two local optima LO_i and LO_j are connected by an ILS iteration (i.e., local search followed by k perturbations) during the search an edge e_{LO_i,LO_j} is stored in the set of LON edges. The nodes and edges logged during r runs of the algorithm are aggregated to form a single local optima network for the problem instance. The construction algorithm differs from that in Chapter 5 because it is based on traces of full ILS runs instead of separate stages to detect nodes and edges, i.e. $2000 \times N$ local searches to identify local optima and then $20N^2$ perturbation and local search cycles from each node to find connectivity between them. All parameters for the algorithm are stated later on in Section 6.5.2.

6.4 Multifractal Analysis

6.4.1 Deterministic Approach

The process for calculating *multifractal* dimension spectra is different to ordinary box counting as it assigns a whole spectrum of fractal dimensions to a single pattern. In Chapter 5, I specialised a multifractal analysis procedure (originally from [184]) to align with LON semantics. This bespoke variant is the foundational algorithm in use during the incumbent fractal analysis experiments. The template matches the previous pseudocode and description of Algorithm 2 and Section 5.3.2, although the parameter choices are different here. These are provided later on. The procedure produces multifractal dimension spectra for LONs.

6.4.2 Probabilistic Approach

Separately, I implement a modified version of the afore-described algorithm where the metaheuristic transition probabilities recorded in LON edge weights are involved — similar to the related approach in Chapter 5, except this is the multifractal case and some adjustments are required to account for the specific network set. With this design, for a node n to be a member of the sandbox with centre c there must be one of two situations: either there is a single edge between n and c (of any probability; this is to guarantee boxing momentum), or there is an edge between a direct neighbour of n and c which is weighted with a probability greater than a specified threshold β. I planned this element with the thought that nodes which are in close proximity to a probable path towards the central node should be included in the sandbox. I removed the fitness distance constraint in this configuration because retaining it resulted in minimal box counting movement due to the distributions of fitness distances and probabilities intrinsic to the LONs under study. Additionally I replace the set of values for "sandbox" radii — which are in use for experiments relating to the other algorithm, detailed just previously in Section 6.4.1 — with a single value: $rd = 2$, which stipulates that only nodes with a single edge separating them can be included in the same sandbox. I argue that larger values for rd do not easily suit the

probability-based paradigm for these particular LONs; when more than one edge is separating nodes, complications arise about how to consider the set of probabilities lying on the path between them — indeed, I noticed that paths which consist of multiple steps often contain only one high probability weight.

The rest of the algorithm matches the deterministic approach (Section 6.4.1) and a spectrum of fractal dimensions is produced. To differentiate the results in the following Sections I refer to fractal dimensions obtained by the probability-centred process as *probabilistic* fractal dimensions, although it should be noticed that they are computed differently to the probabilistic dimensions in Chapter 5. The other fractal analysis algorithm used in the experiments does not include the probability constraint but instead includes "sandbox" radius variation as well as fitness distance constraints and produces values which I refer to as *deterministic* fractal dimensions. The parameters for both algorithms described are stated in Section 6.5.3.

6.5 Experimental Setup

6.5.1 Instances Used

All instances used are from the benchmark library for QAP, QAPLIB [187]. I cap the maximum problem size at 50. Additionally I remove the "esc" instances from the group — I observed that LONs computed from these typically only have one distinct fitness and LON edges are therefore towards a solution of equal fitness, i.e. they form a plateau. After accounting for plateaus by compressing together connected nodes of the same fitness, there are no edges left, i.e., all edges are within the plateaus and have been removed. The absence of 'true' edges would render fractal analysis (which is based on spatial complexity and connectivity between nodes) redundant. The resultant set consists of 85 problems, with sizes ranging from 12 to 50; these are specified in Table 6.1. In all cases the global optimum is known.

Recall from Section 2.2.2 that the nature of QAP instances can be characterised as belonging to one of four classes [191]: uniform random distances and flows; random flows based on grids; real-world; and random "real-world like", which are not real-world but mimic distance and flow patterns seen in real-world presentations of QAP. Table 6.1 shows the QAPLIB instances used in the experiments and presents them in these four categories. Numbers which form part of the instance names indicate the problem size, i.e., the number of locations and flows, and also the length of a permutation solution. The letters in the instance names are shorthand for the instance author's surname, and provide an indication of sub-category into a type of instance specified by that particular author.

6.5.2 LON Construction

For each QAP instance, I construct a local optima network. As stipulated in Section 6.3.2, this is done using an ILS algorithm which is augmented with a cataloguing capacity for building a LON. This amalgamates the unique nodes and edges from 200 ILS runs into a single network. Each run terminates after 10,000 iterations without an improvement. The remaining ILS parameters and setup are detailed shortly in Section 6.5.4.

TABLE 6.1: QAPLIB instances used in the experiments

class	instance names
uniform random	tai { 12a, 15a, 17a, 20a, 25a, 30a, 35a, 40a, 50a } | rou { 12, 15, 20 }
random grid	had { 12, 14, 16, 18, 20 } | nug { 12, 14, 16 {a-b}, 17, 18, 20, 21, 22, 24, 25, 27, 28, 30 } | scr { 12, 15, 20 } | sko { 42, 49 } | tho { 30, 40 } | wil { 50 }
real-world	bur26 {a-h} | chr { 12 {a-c}, 15 {a-c}, 18 {a-b}, 20 {a-c}, 22 {a-b}, 25a } | { els19 } | kra { 30 {a-b}, 32 } | lipa { 20 {a-b}, 30 {a-b}, 40 {a-b}, 50 {a-b} } | ste36 {a-c}
real-world like	tai { 12b, 15b, 20b, 25b, 30b, 35b, 40b, 50b }

6.5.3 Fractal Analysis

In contrast to traditional *monofractal* analysis, to generate *multifractal* dimensions for the LONs a range of arbitrary real-valued numbers is needed. In the deterministic fractal analysis, I set these as 60 values: q in the range $[3.00, 8.90]$ in step sizes of 0.1. The number of "sandbox" centres at each iteration is set at 50 — which I selected due to the moderate size of the networks and to reduce computational cost — and the choice of these centres is randomised. As mentioned, the deterministic multifractal algorithm considers fitness distance in order to specialise to LONs. As was the case in Chapter 5, the comparison between fitness values is handled using logarithmic returns [189]:

$$fitness\ difference = ln(f(LO_i)/f(LO_j))$$

I then compare the fitness difference value with a set threshold ϵ. A range of ten values is used for that algorithm: $\epsilon \in \{0.01, 0.19\}$ in step sizes of 0.02. These values were chosen considering the observed distribution for fitness differences in the LONs. Another essential algorithmic element is the size (i.e., the radius) of the sandboxes. For these I use values rd in the range $[2, diameter - 1]$ (where *diameter* is the LON diameter).

For the probabilistic multifractal algorithm design the fitness constraint is not used and a single value for rd is considered: $rd = 2$. There are again 60 values for q for this setup: q in the range $[3.00, 8.90]$ in step sizes of 0.1. The number of sandbox centres is 50. The probability threshold parameter β must also be chosen. Recall that in essence β controls the minimum edge weight between two nodes to render them eligible for consideration as a single "unit" of detail. In Chapter 5 β was the maximum *opposite* probability, but the LON weights in the present dataset are of a different nature to the ones used in that Chapter — they do not capture an explicit probability (as was the case in the previous LONs) and are in the form of raw hit counts for the search transitions. In light of this, I do not standardise them and instead tried out values for β as the minimum edge weight. In preliminary runs I noticed that if β was set as larger than the minimum weight which is present within the weights distribution (which is often one, i.e., the transition occurred once during runs) then little or no "boxing" occurred. For this reason β is set as the minimum weight present in the distribution; that is, any edge weight which is above the lowest weight is accepted.

An important note. I note here that 32 out of the 85 LONs had only a single edge weight present throughout the network, which means that the LON construction algorithm traversed each edge only once. This renders them ineligible for probabilistic fractal analysis under the specified conditions. Consequently, results which pertain to probabilistic dimensions consider the 53 eligible LONs and their features, while those pertaining to deterministic fractal dimensions cover the complete set of 85 LONs.

6.5.4 Metaheuristic Performance

To obtain algorithm performance information with which to compare the LON features I use two metaheuristic search algorithms for the QAP. Stützle introduced ILS variants for state-of-the-art performance on the QAP [191]. I use this ILS configured as follows: first-improvement pairwise exchanges for local search; $\frac{3n}{4}$ exchanges for perturbation, accepting only improving local optima; and terminating when the global optimum is found or after 100 iterations. Taillard's Robust Tabu Search (ROTS) [190] is also a competitive heuristic for the QAP. This includes a best-improvement pairwise exchange local search with a variable-length tabu list tail. For each item-location combination, the most recent point in the search when the item was assigned to the location is retained. A potential move is deemed to be "tabu" (not allowed) if both items involved have been assigned to the prospective locations within the last $tabu_c$ cycles. The value for $tabu_c$ is changed randomly, but is always from the range $[0.9n, 1.1n]$. A run terminates if the global optimum is found or after 100 iterations. I run the ILS and the ROTS in these configurations on each QAPLIB instance 100 times from different starting solutions. As a measure for their performance, I define the performance gap $performance(algorithm)$ as follows:

$$performance(algorithm) = \frac{O_{algorithm}}{O_{optimal}} \tag{6.1}$$

where $O_{algorithm}$ is the objective value obtained by the metaheuristic algorithm after 100 iterations and $O_{optimal}$ is the objective value of the global optimum. In this way, within the minimisation context of the QAP, a solved run will produce $performance = 1.00$: the metaheuristic has produced an objective value which matches the optimal. Lower-quality objective values are *larger* in the minimisation setting; it follows that higher $performance$ values indicate poorer quality algorithmic output in terms of objective value. In the results that follow, $p(ILS)$ is the mean $performance$ over 100 runs for iterated local search and $p(ROTS)$ is the equivalent for robust tabu search.

6.5.5 LON Features

Features are extracted from the local optima networks. Deterministic fractal dimension sets are calculated for each of the 85 LONs considered. In those sets there are $60 * (diameter - 2) * 10$ dimensions, where 60 is the number of (arbitrary) q values, $diameter$ is the LON diameter (which differs between LONs), and 10 is the number of values for fitness distance threshold ϵ. As we recall from Section 6.5.3, 53 of the 85 LONs are eligible for probabilistic fractal analysis. Those 53 have sets of probabilistic fractal dimensions calculated in addition to the deterministic ones. In each set there are 60 dimensions (one for each value of q), which is less than there are for *deterministic* multifractal analysis; this is because the *probabilistic* variant does not

consider parameter ranges for "sandbox" radius and does not include fitness distance constraints. The statistics I draw from the fractal complexity data are:

1. *median* fractal dimension;
2. *minimum* fractal dimension;
3. *range* of fractal dimensions (calculated as the difference between the largest and smallest values);
4. number of *unique* fractal dimensions.

The last two capture the *extent* of multifractality present. Also considered in the experiments are other LON features which are not related to fractal dimension values:

1. number of *local optima*;
2. *mean fitness* of sampled local optima in the LON;
3. fitness *range*;
4. fitness of *sinks* ("sinks" are nodes which have no out-going edges, i.e. the algorithm used to construct the LON became trapped there);
5. extent of *meta-neutrality*, which is neutrality at the local optima level, computed as $meta\text{-}neutrality = \frac{number\ unique\ fitnesses}{number\ local\ optima}$;
6. mean *out-degree*.

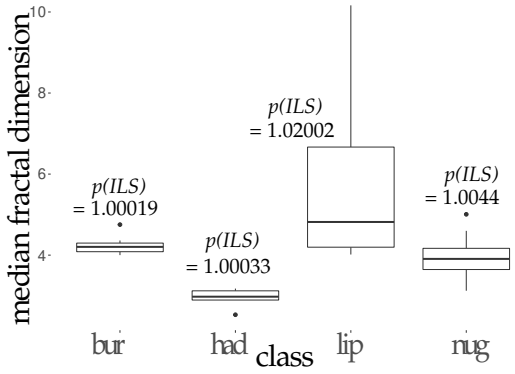
6.5.6 Regression Model Setup

I build algorithm performance models using LON features for predictors and the performance of competitive metaheuristic algorithms as the response variables. The aim is the elucidation of how LON features can contribute to explaining or predicting algorithm proficiency, paying particular attention to the fractal nature of the LON. In pursuit of that I conduct linear and random forest regressions. The number of observations I have is relatively small — 85 for the deterministic dimensions and 53 for probabilistic — so I use *bootstrapping* (random repeated sub-sampling cross-validation) to estimate the sampling distribution for model statistics. The models are bootstrapped for 100 iterations. Pattengale *et al.* found that between 100 - 500 bootstrapping replicates provide the necessary information [178]; I use 100 for computational efficiency, after noticing that separate bootstrapping runs produce similar estimates and that increased iterations have a diminutive effect on the calculated estimates. A training-test split of 80 - 20 is applied. The random forest regression uses 500 trees. All predictors are normalised (due to different value ranges) as follows: $\hat{pre} = \frac{(pre - E(pre))}{sd(pre)}$, with pre being the predictor in question and where $E(pre)$ is the expected predictor value and $sd(pre)$ is the standard deviation. The model statistics I focus on are R^2, which captures the amount of variation in the response variable which can be explained using the predictor set, and *mean squared error*, which expresses the mean squared difference between the model-estimated values and the actual values.

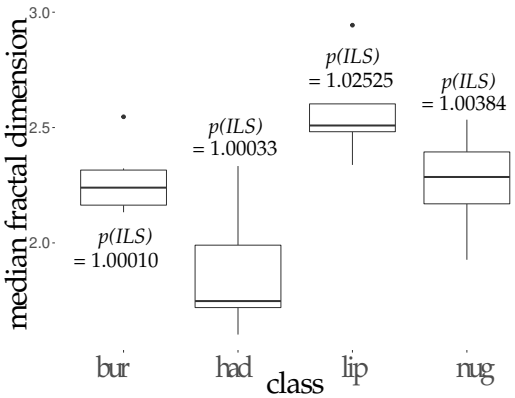
The non-fractal LON predictors used in the models are the mean fitness; fitness range; fitness of sinks; extent of meta-neutrality; out-degree; and the number of global optima. For the deterministic fractal dimensions, I include the minimum fractal dimension and median. In the probabilistic case, these two are replaced with the fractal dimension range and number of unique dimensions.

6.6 Results

6.6.1 Distribution Analysis



(A) deterministic



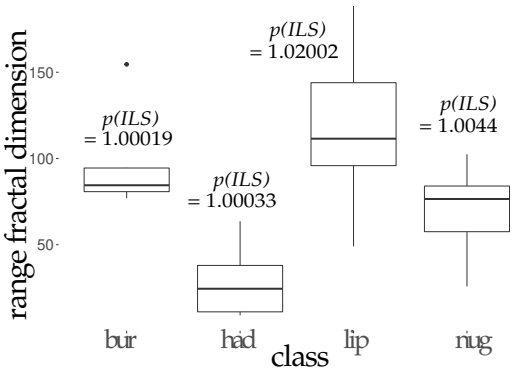
(B) probabilistic

FIGURE 6.1: The distribution of median fractal dimension (over the full set of dimensions produced for the LON). Each box displays values for LONs extracted from one of four QAPLIB instance classes, as indicated along the x-axis. The mean performance of iterated local search, $p(ILS)$, on the QAP instance class for the LONs is also provided

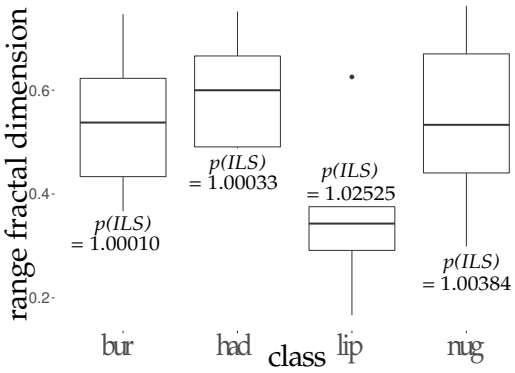
In Figures 6.1 and 6.2, box-plots convey information about the fractal dimensions calculated on the local optima networks. Each box contains values for LONs associated with a particular QAPLIB instance class — those are indicated on the x-axis labels. Only a sub-set of the instance classes which are involved in the central experimentation of this Chapter are considered in these plots. I chose these groups because displaying their distributions alongside each other illustrates evident visual differences between these particular classes. Also provided in the Figures as accompanying text for each box is the performance of iterated local search on the QAP instances associated with those LONs; this is the performance metric $p(ILS)$.

In Figures 6.1a and 6.1b the distributions concern the *median* LON fractal dimension which is associated with using the deterministic and probabilistic methodologies, respectively.

In the case of the deterministic fractal analysis, this is the median value computed over all of the dimensions produced under these conditions; each dimension



(A) deterministic



(B) probabilistic

FIGURE 6.2: The distribution of the range (that is, *maximum value - minimum value*) fractal dimension (over the full set of dimensions produced for the LON). Each box displays values for LONs extracted from one of four QAPLIB instance classes, as indicated along the x-axis. The mean performance of iterated local search, $p(ILS)$, on the QAP instance class for the LONs is also provided

is the output resulting from using a different combination of the fractal analysis parameters q, r and ϵ. The probabilistic median is computed from the spectrum of dimensions associated with the range of values for q.

In both Figure 6.1a and Figure 6.1b, the "lip" class of LONs seem to have the highest values and the "had" group have the lowest. On both plots, the highest value belongs to the "lip" category and the lowest to "had". Notice that in 6.1b the "lip" and the "nug" instances — whose LONs generally have the highest fractal dimensions — also have higher values of $p(ILS)$. As stipulated in Section 6.5.4, values like these reflect that metaheuristic performance was of lower quality. With deterministic analysis, the "lip" group have the largest variation, while the "had" LONs have among the smallest; with probabilistic dimensions (Figure 6.1b), "had" have the largest and "lip" the smallest. Deterministic fractal dimensions appear to be higher than probabilistic fractal dimensions.

Consider now the *range* of fractal dimensions in the deterministic and probabilistic spectra calculated for the LONs, which are given in Figures 6.2a and 6.2b.

The range of fractal dimensions for a LON is a way to quantify the extent of multifractality present and is calculated as *maximum value - minimum value* with respect to the complete set of fractal dimensions produced using either the deterministic or probabilistic paradigm. Also provided is the average ILS performance, $p(ILS)$, for

the QAP instances included in the classes.

Looking at the two plots and noting the different scales used for them, it seems clear that the probabilistic dimension calculation process lends to more compact ranges. This is intuitive: the conditions are stricter for measuring "boxes" during the dimension calculation process. Let us consider in both plots the levels of the black lines (which indicate the distribution median). The "had" group has the lowest in 6.2a and the "lip" group has the highest. That hints that the degree of multifractality in the "lip" group is the most pronounced among the four, and it is the least pronounced in the "had" group. The previous plots told us that "lip" LONs had the highest dimensions, and "had" showed the lowest. It follows that the degree of deterministic multifractality might be associated with lower fractal dimensions. For 6.2b though, "had" LONs have the highest ranges of dimension and "lip" have the lowest — the opposite trend to the deterministic dimensions. With respect to algorithm performance, we can see that the "lip" LONs, associated to problems with the lowest metaheuristic performance ($p(ILS)$), appear to have a higher extent of deterministic multifractality and a lower extent of probabilistic multifractality. In Figure 6.2b, the two problem groups with the best ILS performance have the widest ranges of values for dimension (i.e. amount of multifractality) of the four categories.

6.6.2 Visualisation

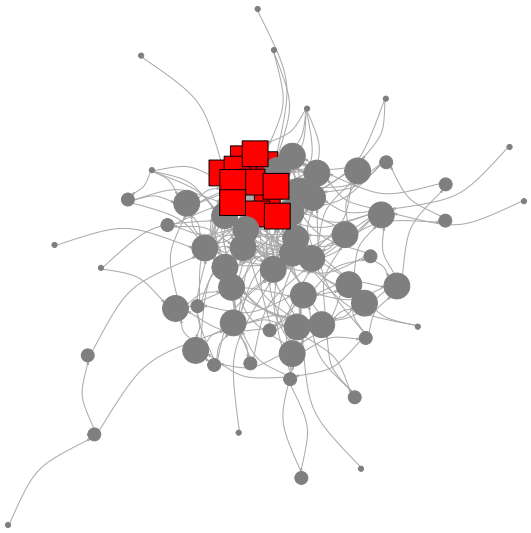
Visual analysis of LONs provides valuable insight into algorithm performance and problem structure, and can augment more empirical or statistical findings [139]. I begin with visualisation before moving onto correlation analysis (Section 6.6.3) and machine learning models (Section 6.6.4) thereafter.

Figure 6.3 shows two partial LONs, each for a different QAPLIB instance. These are laid out according to a force-directed graph layout algorithm in R's iGraph package. Only the fittest 10% of local optima are plotted for visual clarity. Global optima are red squares and all other nodes are grey circles. The node sizes are proportional to the *incoming strength* to that node, which is the weighted incoming degree.

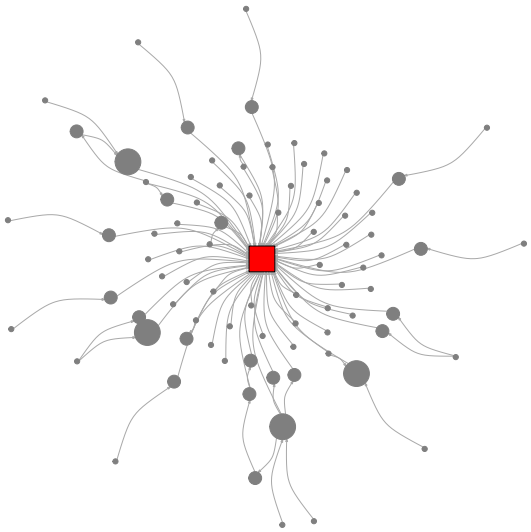
These two LONs were selected from the "had" and "lip" instance classes because the former have lower fractal dimensions than the latter. These two instances chosen have the same problem size, N = 20, and similar numbers of local optima.

In accordance with the higher fractal dimensions, the algorithm performance is lower on the "lip" group of problems. Using as a performance measure *p(ROTS)*, robust tabu search averaged 1.096 on the "lip" instances. For the "had" group this was 1.011. Our task in this Section of the results is to seek explanation in the networks concerning the algorithm performance differences while also paying particular attention to how their fractal nature relates to what is visually seen in the structure.

The median fractal dimension for the LON associated with the "had20" instance, shown in Figure 6.3a, is 2.975; for the "lipa20b" LON, it is 4.015. The range of fractal dimensions for "had20" is around 63, and is around 49 for "lipa20b". An evident difference in the two Figures is the number and connectivity of global optima — Figure 6.3a shows that the "had20" LON has many, and they appear to be densely connected to other nodes. Contrarily, the "lipa20b" LON in Figure 6.3b has a single global optimum, which seems to be more sparsely connected within its network. Also noteworthy is the relative sizes of the non-optimal (grey) nodes. In Figure 6.3a there are many large nodes which are sub-optimal and they have access to the global optima. Figure 6.3b is not the same; in fact, many of the nodes which are one step from the global optimum are very small indeed. That tells us that these nodes have small incoming degree which might hinder ascension through fitness levels during



(A) Partial LON of "had20" QAPLIB instance which has median fractal dimension 2.975 and associated *performance(ROTS)* = 1.011



(B) Partial LON of "lipa20b" QAPLIB instance which has median fractal dimension 4.015 and associated *performance(ROTS)* = 1.154

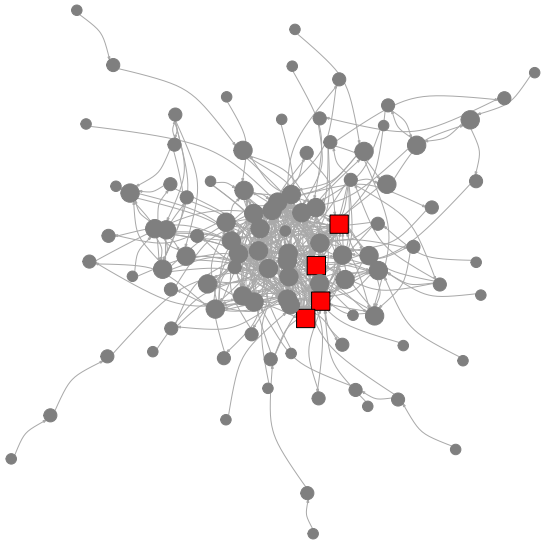
FIGURE 6.3: Partial local optima networks for two QAPLIB instances; only local optima which are in the fittest 10% are shown. Global optima are square and red; all others are grey circles. The size of the nodes captures the incoming "strength" to the node in the LON, i.e. the weighted in-degree

optimisation. These grey nodes are also not well-connected to each other. The opposite is true for the other network. In the "had20" LON (Figure 6.3a), connectivity is so dense in the promising local optima region that visually tracking paths is impossible.

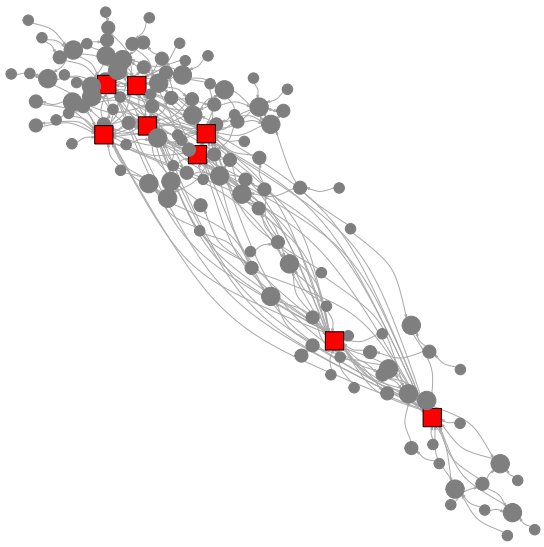
Let us now view Figures 6.3a and 6.3b using an algorithm performance explanation lens. Of course, the number of global optima matters and so does the accessibility of them. The "lipa20b" global optimum has many incoming edges but most of these are sourced from nodes which have low incoming degree themselves. It follows that the global optimum is less accessible. The "had20" LON, which is highly populated with edges in this promising landscape region, is probably easily solvable in part because when an algorithm reaches one of the large grey nodes (this should be probable because they have high incoming degree) there is an abundance of paths to a global optimum. The same trends are present when comparing the two networks in Figures 6.4a and 6.4b.

These are the partial LONs of the "had18" and "nug16b" QAPLIB instances. Figure 6.4a shows "had18", which has a lower median fractal dimension (3.175) and better tabu search performance on the underlying problem (1.011) when compared with "nug16b" shown in Figure 6.4b, which has a median fractal dimension of 4.090 and tabu search performance of 1.055. Surveying the two figures, we can again visually account for the discrepancy in fractal dimension and algorithm performance by looking at the spatial complexity. Although the LON of "nug16b" has more global optima (in red), edges appear less uniformly distributed in their vicinity when compared to the LON of "had18".

In addition we notice that some nodes which are one step from a global optimum in Figure 6.4b are small in size. This tells us that they have low incoming degree and that the probability of search paths reaching them is small. As a consequence potential routes towards the global optima may be missed by algorithms.



(A) Partial LON of "had18" QAPLIB instance which has median fractal dimension 3.175 and associated *performance(ROTS)* = 1.011



(B) Partial LON of "nug16b" QAPLIB instance which has median fractal dimension 4.090 and associated *performance(ROTS)* = 1.055

FIGURE 6.4: Partial local optima networks for two QAPLIB instances; only local optima which are within the fittest 15% are shown. Global optima are square and red; all others are grey circles. The size of the nodes captures the incoming "strength" to the node in the LON, i.e. the weighted in-degree

6.6.3 Correlation Analysis

Figures 6.5 and 6.6 show pairwise Spearman correlations between variables. In Figure 6.5, the statistics reflect deterministic dimensions; in Figure 6.6, the associated dimensions are probabilistic. Included are the metaheuristic performance measures alongside fractal dimension (FD) features of the LONs. For the deterministic fractal dimensions, these are the minimum FD (`FD_min`), median FD (`FD_med`), a dimension from an arbitrary point on the spectrum with rd=3 and ϵ=0.05 (`FD_r3e05`), and another with rd=3 and ϵ=0.09 (`FD_r3e09`). In the case of the probabilistic dimensions, the features are the minimum FD (`FD_min`), range of FD (`FD_rng`), number of unique dimensions (`FD_uni`) and a single dimension from the spectrum (`FD_1`). Different features are considered for the type of dimension (deterministic or probabilistic) because they seem to have different relationships to search; for example, the median *probabilistic* FD does not have a significant ($p < 0.05$) correlation to `ILS` and `tabu`. I therefore shortlist differing FD features depending on FD type. Fitness landscape features which are computed from the LON but are not related to fractal complexity are additionally included. These are the mean outgoing edges from nodes (`outdegree`), the mean fitness of local optima in the LON, which is labelled as `fitness`, and the number of local optima (`optima`).

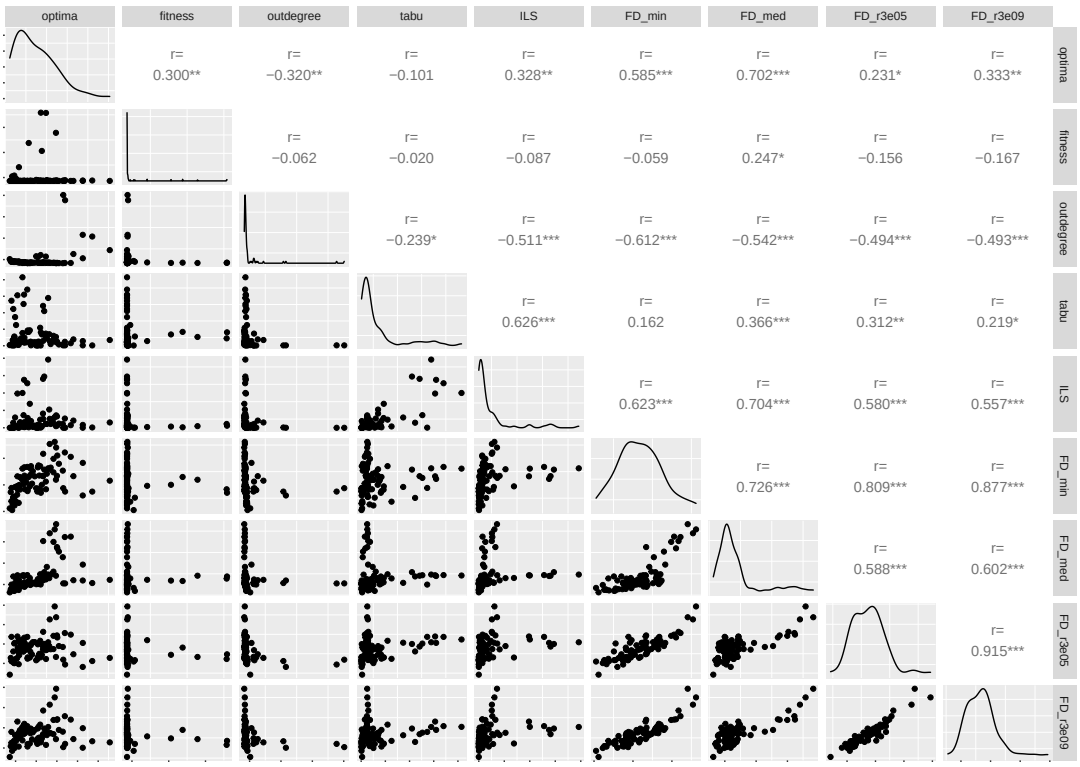


FIGURE 6.5: Spearman correlations between pairs of variables, including algorithm performance measures, deterministic fractal dimension metrics for the LONs (those that include `FD`), and other fitness landscape features

In particular I am interested in the correlation between fractal features of the LON and algorithm performance variation on the associated combinatorial problem. Looking first at the intersections between the `ILS` row and the fractal feature columns in Figure 6.5, this reveals moderate positive correlations with all four (`FD_min`, `FD_med`, `FD_r3e05` and `FD_r3e09`) and all have $p < 0.001$. The strongest of these is the median fractal dimension. Checking the `ILS` column now, notice

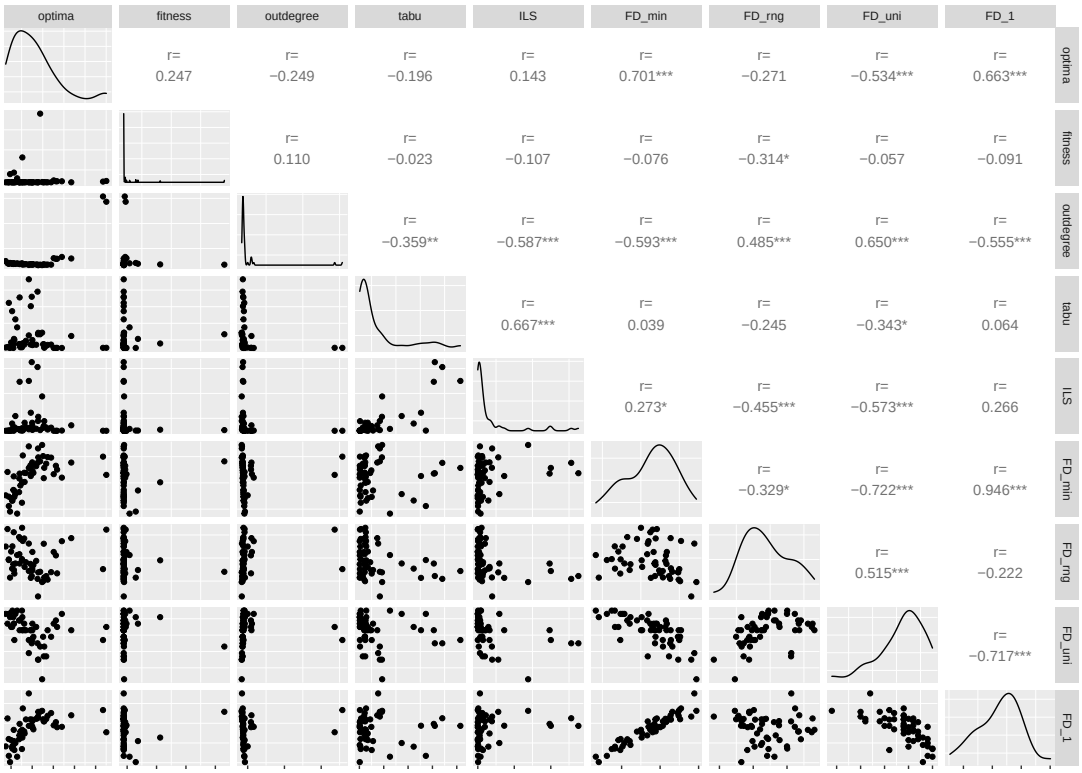


FIGURE 6.6: Spearman correlations between pairs of variables, including algorithm performance measures, probabilistic fractal dimension metrics for the LONs (those that include `FD`), and other fitness landscape features

that these are stronger than the `ILS` correlation with the mean local optima fitness (`fitness`) and the number of local optima (`optima`). There is a moderate negative correlation in the case of the LON `outdegree`. Next, onto the `tabu` feature shown in Figure 6.5. There are weak positive correlations with each of the fractal dimension variables here. Although they are weak, in all except one the p-value indicates significance. The p-value is lowest, and the correlation highest, for `FD_med`, the median fractal dimension.

Next we will consider the correlation plot which includes probabilistic fractal dimension variables in Figure 6.6. Examining the `ILS` and `tabu` rows, observe that the most noteworthy fractal features here appear to be `FD_rng` and `FD_uni` — both of which relate to the extent of *multifractality* present. For these, there are negative correlations with the algorithm performance variables. These are stronger for `ILS` than for `tabu`, and with the former the p-values are less than 0.001. There is also a weak positive correlation with $p < 0.05$ between the minimum probabilistic dimension and `ILS` performance. Excerpting a single dimension from the spectrum does not appear to draw a relationship to the algorithm performances — notice that while `FD_1` is weakly positively correlated to `ILS` and `tabu`, there is no indication of statistical significance present. The correlations between local optima fitness and number of local optima with `ILS` and `tabu` are less than some fractal dimension features (in particular `FD_rng` and `FD_uni`).

6.6.4 Algorithm Performance Regression Models

Deterministic Fractal Dimensions.

Table 6.2 contains regression model statistics whose values are estimated over 100 random repeated sub-sampling (bootstrapping) iterations. Each row represents a particular model set-up. The response variable is shown in the second column. The R^2 and mean squared error (MSE; recall Section 2.7) are given.

TABLE 6.2: Summary statistics estimated with bootstrapping for explaining the performance of ILS and ROTS. Predictors include deterministic fractal dimension LON statistics, as well as other landscape features such as fitness distribution measures

type of regression	response variable	R^2	mean squared error
linear	$p(ILS)$	0.197	0.002
linear	$p(ROTS)$	0.194	0.038
random forest	$p(ILS)$	0.524	0.002
random forest	$p(ROTS)$	0.641	0.005

TABLE 6.3: Variable importance rankings for the random forest models which include deterministic multifractal dimensions in the predictor set. Columns are labelled with the model response variable

feature	$p(ILS)$	$p(ROTS)$
fractal dimension minimum	6	8
fractal dimension median	1	4
fitness of sinks	3	2
fitness range	8	5
mean fitness	2	3
number of global optima	4	7
out-degree	5	6
extent of meta-neutrality	7	1

We can see from the R^2 values that random forest regression produces a stronger model fit. This is likely because random forest trees are adept at considering non-linearities between variables. The amount of variation in the iterated local search and tabu search performance which can be explained using the predictors is higher in the random forest models. The mean squared error is very low in the case of the random forest regression which is explaining $p(ILS)$, indicating a better fit. The strongest model in terms of R^2 is using random forest regression with $p(ROTS)$ as the response, with approximately 64% of variance being explained using the landscape features. Less variance in $p(ILS)$ response, around 52%, is explained using the same type of regression. This model setup does, however, have a lower error rate than the associated $p(ROTS)$ model.

Now let us look at the random forest predictor importance rankings, which are provided in Table 6.3. This is the most common ordering observed over 100 bootstrapping iterations. For explaining $p(ILS)$, the *median fractal dimension* is most important. *Minimum fractal dimension* ranks moderately well, placing sixth out of eight features. The *fitness range* and *extent of meta-neutrality* are the least important factors.

Extent of meta-neutrality is the most important feature for explaining $p(ROTS)$ variance. Metrics relating to the fitness levels in the local optima sample, such as *fitness of sinks*, *mean fitness*, and *meta-neutrality*, dominate the importance rankings for this model setup. The fractal dimension *median* contributes moderately, ranking fourth out of eight. *Miminum fractal dimension* is the least important factor.

Probabilistic Fractal Dimensions.

Table 6.4 presents model statistics where the predictor set includes probabilistic fractal dimension features instead of the deterministic ones seen in Table 6.2. This is followed by the associated random forest predictor rankings in Table 6.5.

TABLE 6.4: Summary statistics estimated with bootstrapping for explaining the performance of ILS and ROTS. Predictors include probabilistic fractal dimension statistics, as well as other landscape features such as fitness distribution measures

type of regression	response variable	R^2	mean squared error
linear	$p(ILS)$	0.263	0.001
linear	$p(ROTS)$	0.304	0.022
random forest	$p(ILS)$	0.315	0.000
random forest	$p(ROTS)$	0.557	0.040

TABLE 6.5: Variable importance rankings for the random forest models which include probabilistic multifractal dimensions in the predictor set. Columns are labelled with the model response variable

feature	$p(ILS)$	$p(ROTS)$
fractal dimension range	5	7
unique fractal dimensions	2	8
fitness of sinks	6	1
fitness range	7	4
mean fitness	4	2
number of global optima	1	5
out-degree	8	6
extent of meta-neutrality	3	3

The random forest $p(ILS)$ model setup expains approximately 31% of variance. It should be reiterated at this point that the data-set is composed of fewer observations here than in the previous models (Tables 6.2 and 6.3). There are 53 observations here, compared with a previous 85. This might impact the formulation of a well-fitting model. Nonetheless both setups with $p(ILS)$ as the response variable have markedly lower mean squared errors than their ROTS counterparts. This is also true in Table 6.2. Back in Table 6.4, the $p(ROTS)$ models have higher mean squared errors but the random forest model is definitely the strongest with respect to search algorithm explanation, with around 56% being accounted for by the predictors. Although a smaller portion of variance is explained in the $p(ILS)$ models, the low mean squared errors are encouraging in accuracy terms.

In the predictor rankings, seen in Table 6.5, observe that *fractal dimension range* comes contributes moderately in in the ILS model and second-last in the ROTS

one. The *unique fractal dimensions* feature is second most important feature in the ILS setup, but ranks last in the ROTS model. The strongest predictors for $p(ROTS)$ appear to be relating to the local optima level fitness distribution. *Outdegree* is of low importance in both models.

6.7 Conclusion

I conducted multifractal analysis on LONs associated with instances belonging to a benchmark combinatorial optimisation problem library: QAPLIB. The QAPLIB instance set was more than three times the size of the set used in Chapter 5 and raised the considered problem sizes from $N \leqslant 28$ to $N \leqslant 50$. A recent and refined LON construction algorithm [24] was used to build the LONs. Relationships between fractal dimension features of LONs and algorithm performance by iterated local search (ILS) were established using correlation analysis, visual analysis tools, and linear and random forest regression with bootstrapping. The results showed that the *extent* of multifractality and the high-ness of values in the dimension spectrum can contribute towards partially predicting or explaining ILS and ROTS algorithm performance. Certain features of the fractal dimension distribution for the LONs also displayed individual pairwise correlations to ILS and ROTS algorithm performance, although this was more pronounced for ILS. While the considered fractal dimension features in LONs contributed as predictors in models, sampled fitness levels in the LON were usually more important for ROTS prediction and explanation. The present Chapter could serve as a foundation for further work within this research avenue which remains untapped. In particular, I would like to expand the maximum size of the problems studied, as well as venturing to other domains and to constrained problems. Finally, I conclude with a remark concerning my interest in studying the relationship between perturbation *strength* used to generate the LONs, and the calculated fractal dimensions of that LON.

6.8 Summary

So far, the Chapters in this thesis have extracted features from LONs to gain insight into algorithm-problem dynamics and reactions. In the case of problems non-trivial in size, this necessitates sampling of the LON, which carries attendant sampling bias. The importance of sampling mindfully can hardly be overstated: features extracted from misleading LONs could also be misleading and lead to erroneous deductions. Chapter 7, which follows from here, makes inroads into a better understanding of sampling-based LON construction algorithms.

Chapter 7

Inferring Future Landscapes: Sampling the Local Optima Level

In the contributions thus far I have used LON features to generate insight about algorithm-problem dynamics. For *sampled* (approximate) LONs this kind of analysis has a foundational assumption: that the sample is accurate. As LON analysis moves closer to real-world applications, it is essential to understand the LON construction algorithms, which dictate the information extracted from LONs, themselves. The upcoming Chapter advances towards this goal. Included is a comparison of fully-enumerated LONs with their sampled counterparts; an "equivalent computation" appraisal of LON construction algorithms; and regression models which combine the most impactful features of ILS Sampling LONs with those of Snowball Sampling LONs together for algorithm performance prediction.

7.1 Abstract

Connection patterns found in LONs can help to explain metaheuristic performance during optimisation. LON research has predominantly considered complete enumeration of a fitness landscape, thereby restricting analysis to problems which are diminutive in size compared to real-life situations. LON *sampling* algorithms are therefore important. In this Chapter I study LON construction algorithms with a focus on the QAP. Initially, a descriptive analysis of the networks themselves is executed. After that, using machine learning, I use LON features to predict algorithm performance for competitive metaheuristics used in the QAP domain. The results show that by using random forest regression, construction algorithms produce LONs with features which can explain the majority of variance in performance on the problem instances. I find that sampled LONs have a more pronounced affiliation with metaheuristic proficiency than fully-enumerated LONs do. The importance of fitness *levels* or layers present in sampled LONs is crystallised here; this is relevant and related to fitness-based genetic algorithm runtime analysis [193]. Features taken from LONs built using different construction mechanisms are combined together in predictive models for the first time, with promising results for this "super-sampling" approach: the model is able to explain the vast majority of variation in tabu search performance. Arguments are made for the use-case of both construction algorithms and for combining the exploitative process of Snowball Sampling with the exploratory optimisation of the ILS Sampling.

7.2 Introduction

LONs are a fitness landscape model specifically designed to understand or predict reactions between configuration spaces and metaheuristic search algorithms. Recall from the perusal of Chapter 3 that studying LON objects has brought colour and clarity to our understanding of how optimisation problems and search algorithms interact [5, 19, 165, 12].

Principally, studies have enumerated the fitness landscape to build a comprehensive LON [11, 194, 5, 4, 8, 23, 25]. Their focus is on smaller problems and in that approach every candidate solution is mapped to a local optimum within its basin of attraction. This baseline and these proof-of-concepts were necessary to establish LONs as a prosperous tool. LONs have attracted more widespread attention recently [195, 13, 126, 127, 196, 59, 22], and will likely be applied to increasing numbers of real-world and much larger problems in the future. In anticipation of these requirements, refined LON sampling methods are needed.

Literature concerning LON construction by sampling is in its embryonic stages, however. A few LON construction algorithms have been proposed recently, but have not been extensively tested; I recap here the ones which were initially implemented for the Quadratic Assignment Problem, although I additionally note that the templates underlying the algorithms could easily serve as the basis for construction algorithms for an arbitrary combinatorial optimisation problem.

Two-phase Sampling. In 2014 a construction algorithm for LONs was introduced [7]. The procedure has separate phases for recording nodes and edges. Initially, local optima are found by hill-climbing from $2000 \times n$ starting solutions and these form the node set. Thereafter, those local optima are individually subject to perturbation and then hill-climbing. If the obtained local optimum is also in the node set, an edge is added (or if it exists already, the edge *weight* is incremented). Two-phase LON Sampling is not used in the present study because it is less recent than the other sampling algorithms and has not been utilised in follow-up studies.

Iterated Local Search Sampling. ILS Sampling [24] is augmented with a competitive ILS algorithm for the QAP [191] and comprises the high-level trajectory of multiple ILS runs, each being an adaptive walk on the local optima space. Local optima and transitions between them are recorded. The same framework for Travelling Salesman Problem LONs has been used with some success [195]; this is also instrumented on top of a competitive ILS heuristic for the domain — Chained Lin-Kernighan [197].

Snowball Sampling. With Snowball Sampling [25], a random walk takes place on the local optima level. From each node in the walk, a recursive branching or *snowballing* expands the node by sampling its local optima neighbours. The local optima are found using perturbation followed by hill-climbing, as in ILS Sampling. All local optima are saved as nodes for the LON and bonds (transitions) between neighbouring local optima are recorded as edges.

This Chapter has a heavy focus on the relevance of sampled LONs to empirical search difficulty and quantifies this relationship with performance prediction using LON features as predictors. The results and experiments of this study come in five parts. First is a descriptive and exploratory comparison concerning the nature of

the networks produced by the two construction algorithms. The subsequent results focus on predictive modelling using features of the LONs: the comparison of sampled LONs with their fully-enumerated equivalents; budgeted LON construction; un-budgeted LON construction with a variety of sampling parameter inputs; and finally the most impactful features of LONs built by ILS Sampling and by Snowball Sampling are combined together into regression models.

The contributions are as follows:

1. A first descriptive and statistical comparison of LONs produced by different construction algorithms;
2. The novel comparison of sampled and fully-enumerated LONs in terms of their predictive potency is given;
3. For the first time LON construction algorithms are given an equal computational budget and the effect is studied;
4. I additionally propose to combine specific features from LONs which are built using separate procedures together in performance prediction regression models.

7.3 Definitions

7.3.1 Funnels

Some of the LON features which are employed in the experimentation of this Chapter relate to the notion of funnels. Recall from Section 3.4 that *funnels* are documented fitness landscape features in combinatorial optimisation [119, 56] but originate in the study of physical energy landscapes [130]. The precise definition in evolutionary computation is an area of active research but a series of papers [56, 195, 167] consider them to be a basin of attraction at the local optima level. In Figure 7.1, notice the presence of multiple small basins; overall though, these conform to two much larger basins. The large basins are *funnels*, and they contain many local optima organised in a fitness hierarchy. In this minimisation example, there exists a shallower sub-optimal funnel (on the left) and a deeper, optimal funnel (on the right). For the purposes of this thesis I consider the associated definition that a *funnel* is the collection of monotonically-improving paths through local optima which terminate at a single local optimum. To find the paths, sink nodes are identified within a LON — these are simply the nodes with no outgoing and improving edges (i.e. edges where the destination node has superior fitness to the source). From those, a breadth-first search is conducted on the LON, exposing the set of paths which terminate at that particular sink. These paths together — a path includes both the nodes and the edges — comprise the funnel which surrounds the sink.

7.4 Methodology

The test instances are QAPs. The solution representation, fitness function, instance format, and objective are detailed in Section 2.2. In the experiments that follow I focus mostly on the much-studied QAPLIB, which boasts a diverse selection of problems — both synthetic and real-world.

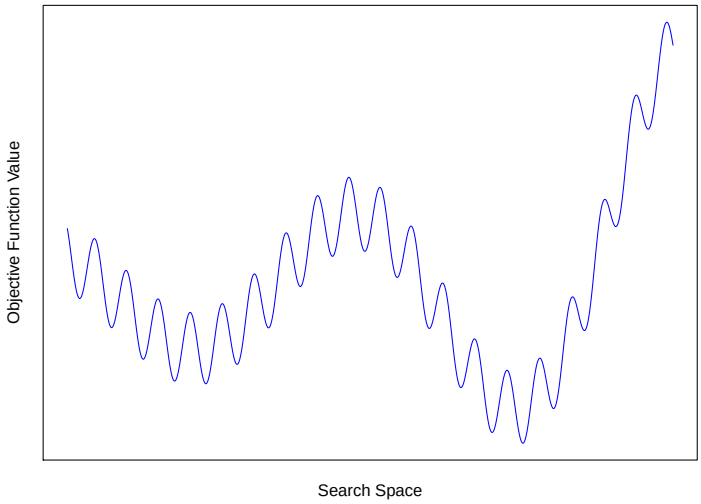


FIGURE 7.1: Abstract depiction of a hypothetical layout for local optima — a sub-optimal funnel on the left and the *primary* (optimal) funnel on the right for an imagined minimisation problem

7.4.1 Instances for Exploratory Comparison

The initial prong of experimentation is an exploratory and descriptive comparison of the networks produced by different LON construction algorithms. For this endeavour, reported in results Section 7.6.1, I include 30 moderately-sized instances from various problem classes. This is chosen with the thought that the range for descriptive statistics should not be overly large. The problem dimensions N are between 25 and 50, meaning 25 - 50 items to be assigned to 25 - 50 locations. Recall from Section 2.2.2 that there are four problem categories for QAP; the instances are now delineated into their associated instance class in Table 7.1.

TABLE 7.1: QAPLIB instances used in Section 7.6.1

class	instance names
uniform random	tai { 25a, 30a, 35a, 40a }
random grid	nug { 25, 27, 28, 30 } | sko42 | tho { 30, 40 } | wil50
real-world	bur26 {a-b} | chr25a | esc32e | kra { 30 {a-b}, 32 }
	lipa { 30 {a-b}, 40 {a-b} } | ste36 {a-c}
real-world like	tai { 25b, 30b, 35b, 40b }

7.4.2 Instances for Regression

For the regression experiments — presented shortly in Sections 7.6.2 - 7.6.5 — I include as many benchmark instances as possible to facilitate proper statistical analysis and evidence-building about predictions. QAPLIB contains 128 instances where the global optimum solutions are provided. I use 124 of these 128 for regression, omitting the largest three due to computational cost (*tho150*, *tai150b*, and *tai256c*), and also *esc16f* because all flow entries are zero in the instance specification. The resultant set contains problems of size $N \leqslant 128$. A set of sixty additional instances (N = 11) which are not from QAPLIB also play a part in the experiments. Their size enables a complete enumeration of the fitness landscapes and LONs, which is required to facilitate comparison between sampled LONs and their exhaustively-enumerated

counterparts. As there is a low number of instances which are small enough for this in the benchmark QAPLIB, the separate non-benchmark set is necessary for statistical analysis to take place.

7.4.3 LON Construction Algorithms

This Section describes, in turn, each LON construction algorithm which is involved in the experiments and findings of this study.

Exhaustive Enumeration

The method for exhaustively enumerating a local optima network was introduced alongside the model itself [11] and then adapted for the QAP [5]. A history of LON construction algorithms, including those which conduct exhaustive enumeration, can be found in Section 3.3. Nevertheless, I will concisely refresh the essentials here to render this Chapter self-contained. LONs are enumerated using a best-improvement local search which considers the elementary operation for QAP — a pairwise exchange of items in the permutation solution. The local optimum LO_i for each solution is found this way and local optima are added as nodes in the network. The escape edges are defined according to the distance function d and a maximum number of applications of d which separate the two local optima; $D > 0$. An edge e_{LO_i,LO_j} is traced between LO_i and LO_j if a solution s exists such that $d(s, LO_i) \leq D$ and $h(s) = LO_j$, where $h(s)$ is the hill-climbing function which maps solution s to a local optimum. The weight of this edge is $W_{LO_i,LO_j} = |\{s \in S : d(s, LO_i) \leq D \text{ and } h(s) = LO_j\}|$. This weight can be normalised by the number of solutions, $|\{s \in S : d(s, LO_i) \leq D\}|$, within reach at distance D. In the present study, I set $D = 2$.

Pseudocode for the best-improvement algorithm is shown in Algorithm 4. The mechanism begins from every possible solution, as is seen at line 2. After that, a candidate replacement solution $solution'$ is set as whichever neighbour of $solution$ (that is, a member of its neighbourhood $\mathcal{N}(solution)$) has the best fitness in the neighbourhood; notice this at line 4.

Algorithm 4 Best-improvement

Input: initial solution
Output: a local optimum

```
1: procedure HILL.CLIMBING
2:     solution ← random initial solution
3:     while solution ≠ local optimum do
4:         set solution' ∈ N(solution), such that f(solution') = min_{y∈N(solution)} f(y)
5:         if f(solution') < f(solution) then
6:             solution ← solution'
```

If $solution'$ has superior fitness to $solution$, then $solution'$ replaces the current working solution — observe this at lines 5-6. This procedure of replacing the working solution with its highest-fitness neighbour continues until the working solution, $solution$, is a local optimum.

Snowball Sampling

Snowball sampling is a strategy originating from social science where each survey respondent asks a few of their friends to also complete the survey [198]. The resultant social network of respondents is analogous to a rolling snowball which grows progressively larger as it advances through time and space.

Snowball Sampling was introduced for LON construction recently [25]. In this context of fitness landscape exploration, the operation is essentially a branching random walk on the space of local optima. The LON construction algorithm is configurable with the parameters l (which is the length of random walk), de (depth of snowballing) and se (the number of sampled edges).

Pseudocode is available in Algorithm 5. To construct a LON sample, the algorithm begins from a random solution and hill-climbs to a local optimum; observe lines 2-3. The local optimum is assigned as the first in a random walk through the local optima space and is also added as a LON node (see line 4). From this local optimum — which is at timestep $t = 0$ in the walk — and also for each successor on the random walk, the snowball procedure expands the node by exploring for neighbouring local optima. The expansion process is captured in the `SNOWBALL` function; a local optimum neighbour is identified at line 4 of that procedure. Local optima are reached by carrying out perturbation followed by hill-climbing. This is best-improvement and uses a random pairwise swap of items as the search operation and the perturbation operator is four swaps. Local optima are added as LON nodes. As transitions between pairs of local optima are traversed, directed edges are added to the LON accordingly (note line 10 of the `SNOWBALL` procedure). Notice also at line 11 that expansion of nodes is recursive, that is, after the neighbours "once removed" are identified, their own neighbours are found — and so on. The depth of sampling de controls the depth of this recursive process.

After the expansion and back at the original node the random walk continues: a neighbour is chosen as the next member of the walk. Line 8 in the first procedure corresponds to this algorithmic element. That neighbour is added as a LON node and is then subject to the same neighbourhood expansion as before: see line 7. The whole process continues until the primary walk is of length l and all l nodes have been expanded to depth de.

ILS Sampling

Iterated Local Search Sampling (ILS Sampling) [24] is an augmented metaheuristic [191]. The system was used to extract LONs and was concisely described in Chapter 6 and in Section 6.3.2. In the incumbent analysis, however, the particular mechanisms of the algorithm are more salient; because of this, I will deconstruct ILS Sampling with attention to detail in this Section.

Pseudocode is shown in Algorithm 6. A LON is constructed according to the paths which the ILS algorithm traverses through the fitness landscape. A run begins from a random starting solution — see line 4 — which is subject to hill-climbing and transforms to a local optimum before being added as a LON node. The hill-climbing in this particular implementation is a first-improvement local search with the pairwise exchange neighbourhood. This operator swaps two positions in the permutation solution. The local optimum at this stage in the Algorithm is then perturbed with strength k — the perturbation operator exchanges k randomly chosen items — and then hill-climbing is applied (notice lines 9-10). The resultant local optimum is recorded as a node; observe line 11. If there has been a fitness improvement

Algorithm 5 LON Snowball Sampling

Input: depth of snowballing de, number sampled edges se, length of walk l
Output: a LON sample

1: **procedure** SAMPLING(search space S, fitness function f, depth of snowball expansion de, number sampled edges se, length of random walk l):
2: Choose initial random solution $solution \in S$
3: $local.optimum \leftarrow$ HILL.CLIMBING($solution$)
4: $nodes \leftarrow \{local.optimum\}$
5: $edges \leftarrow \emptyset$
6: **for** $t \leftarrow 0, \ldots l-1$ **do**
7: SNOWBALL(de, se, $local.optimum_t$)
8: $local.optimum_{t+1} \leftarrow$ WALK($local.optimum_t$)

1: **procedure** SNOWBALL(d, m, $local.optimum$)
2: **if** $de > 0$ **then**:
3: **for** $j \leftarrow 1, \ldots se$ **do**
4: $neighbouring.local.optimum \leftarrow$ HILL.CLIMBING(PERTURB($local.optimum$))
5: $nodes \leftarrow nodes \cup \{neighbouring.local.optimum\}$
6: **if** $(local.optimum, neighbouring.local.optimum) \in edges$ **then**:
7: $edge.weight \leftarrow edge.weight + 1$
8: **else**
9: $edge.weight \leftarrow 1$
10: $edges \leftarrow edges \cup \{[local.optimum, neighbouring.local.optimum, edge.weight]\}$
11: SNOWBALL($de-1$, se, $neighbouring.local.optimum$)

1: **procedure** WALK($local.optimum_t$)
2: $neighbour.set \leftarrow \{local.optimum : (local.optimum_t, local.optimum) \in edges \wedge local.optimum \notin \{local.optimum_0, \ldots, local.optimum_t\}\}$
3: **if** $neighbour.set \neq \emptyset$ **then**:
4: Select randomly $local.optimum_{t+1} \in neighbour.set$
5: **else**
6: $local.optimum_{t+1} \leftarrow$ HILL.CLIMBING($local.optimum$)
7: $nodes \leftarrow nodes \cup \{local.optimum_{t+1}\}$
8:
9: **return** $local.optimum_{t+1}$

from carrying out the local optimum transition then an edge is added to the LON, which happens at line 18 of the Algorithm (if the edge exists already, the weight is incremented).

Each run of r total runs terminates either when the global optimum is found, or after t iterations without improvement.

7.4.4 Visualisation

Figure 7.2 compares LONs which were constructed using ILS Sampling and Snowball Sampling, respectively. Both are extracted from the QAPLIB instance *wil50*. The Figure serves as an abstract precursor to a more formal comparative analysis later on; nonetheless, it captures a remarkable amount of information. The sample in Figure 7.2a is extracted using ILS Sampling, while Snowball Sampling is used to build

Algorithm 6 QAP ILS Sampling

Input: perturbation strength k, stopping condition t, number of runs r
Output: a LON sample

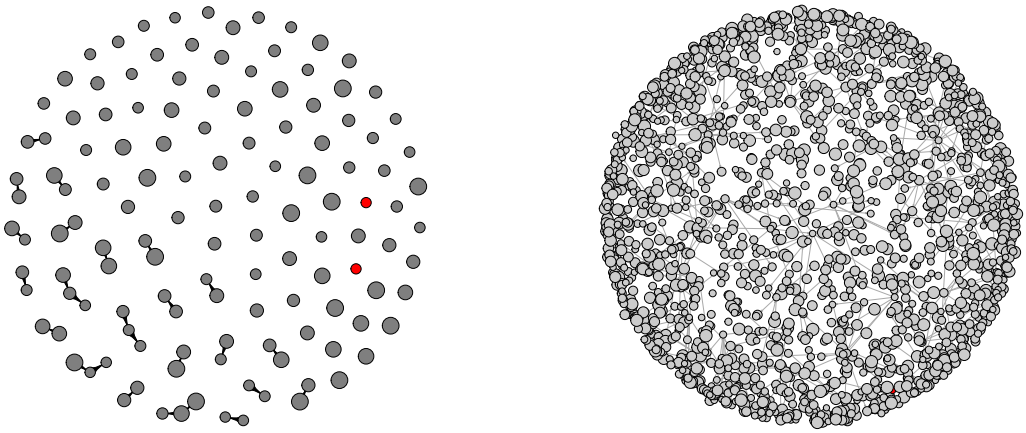
procedure SAMPLING(search space S, fitness function f, perturbation strength k, stopping condition t, number of runs r):
  $runs \leftarrow 0, nodes \leftarrow \emptyset, edges \leftarrow \emptyset$
  repeat
    Choose initial random solution $solution \in S$
    $local.optimum \leftarrow$ HILL.CLIMBING($solution$)
    $nodes \leftarrow \{[local.optimum]\}$
    $iterations \leftarrow 0$
    repeat
      $current.solution \leftarrow$ PERTURB($previous.local.optimum, k$)
      $new.local.optimum \leftarrow$ HILL.CLIMBING($current.solution$)
      $nodes \leftarrow nodes \cup \{new.local.optimum\}$
      if $f(new.local.optimum) \leqslant f(previous.local.optimum)$ **then**:
        $previous.local.optimum \leftarrow new.local.optimum$
        if $(previous.local.optimum, new.local.optimum) \in edges$ **then**
          $edge.weight \leftarrow edge.weight + 1$
        else
          $edge.weight \leftarrow 1$
        $edges \leftarrow edges \cup \{[previous.local.optimum, new.local.optimum, edge.weight]\}$
      $iterations \leftarrow iterations + 1$
    until $iterations \geq t$
    return $previous.local.optimum$
  until $runs == r$

the LON seen in Figure 7.2b.

In red is the global optimum (or the global optima), while all other nodes are grey in colour. Both samples have been capped by fitness: only the most fit 5% (in the case of Figure 7.2a) or 0.05% (Figure 7.2b) are plotted. The threshold is lower for the LON in Figure 7.2b because the sample is orders of magnitude larger and there are edges crowding and obscuring the network.

Comparing the two plots, there are striking differences. Taking them in turn notice that this elite-fitness subset of the LON which ILS Sampling produces (Figure 7.2a) is extremely sparse, and indeed the two global optima are isolated nodes. This could indicate a low probability for search hitting one of these apparently inaccessible nodes; alternatively, it could be that the global optima are in fact well connected but only to inferior fitness levels in the local optima space.

Surveying the LON in Figure 7.2b, built by Snowball Sampling, it is apparent that this is a markedly more dense network. There are significantly more nodes and edges. Recalling that this is only the 0.05% most fitness-elite local optima (which were sampled), the implication is that this sampling method reveals a rich view of the neighbourhood surrounding the global optimum (at the local optima level), and may be more suited to characterising highly-promising regions in the fitness landscape. A global optimum can be seen, albeit partially obscured, in the bottom-right of Figure 7.2b.



(A) ILS Sampling LON; the fittest 5% of local optima are shown. Mean fitness 48858.3.

(B) Snowball Sampling LON; the fittest 0.5% of local optima are shown. Mean fitness 54038.8.

FIGURE 7.2: LONs extracted for the *wil50* instance from QAPLIB; Figures show LONs obtained by two different construction methods. The global optimum (or optima) in the samples is shown in red. Note that in Figure 7.2b the red node is partially obscured, but is visible in the bottom-right segment

7.4.5 LON Features

LON features are used for analysis of the networks themselves and they also serve as predictors in regression models for algorithm performance explanation. Features taken from ILS Sampling LONs are:

1. the number of nodes sampled: *number optima*;
2. edges sampled: *edges*;
3. mean sampled fitness: *mean fitness*;
4. number of outgoing edges from nodes: *out-degree*;
5. diameter of the LON, which is the longest path between nodes: *diameter*.

I additionally include funnel features. In particular:

1. the number of sink nodes (i.e., nodes with no outgoing improving edges in the LON: *sinks*;
2. number of compressed local optima (after connected nodes of the same fitness have been merged during funnel pre-processing: *compressed local optima*;
3. incoming edge weight to sub-optimal sinks in the LON: *sub-optimal sink strength*;
4. the proportion of LON edges pointing at a global optimum sink: *incoming global*);
5. mean sink fitness: *sink fitness*.

Some of the features mentioned so far may not be well-suited in roles as predictors in performance models in the case where the LONs have been produced by Snowball Sampling. Funnel features calculated on them may not possess predictive proficiency because during the LON construction process search paths are restricted and prescribed by the nature of the sampling. In fact, the sampling induces a consistent and extremely large number of apparent "funnels" (at least according to the definition and calculation method for funnels in Section 7.3.1). The short branching paths also result in a LON containing numerous nodes with no outgoing edges, which — when honouring my definition — are identified as sinks or funnel floors. Similarly, standard features of the samples such as cardinality of nodes, edges, and out-degrees are redundant as predictors in this type of LON — they are artefacts of the sampling parameters: length of the random walk, number of edges, and depth of snowball expansion. For Snowball Sampling LONs I extract predictors based on the attributes encoded in the nodes and edges: the fitness distribution and the edge weight distribution. These are more appropriate to what the Snowball sample has to offer. Included is:

1. the mean weight of self-loops (*weight loops*);
2. mean weight disparity of outgoing edges (*weight disparity*);
3. fitness-fitness correlation between neighbours (*fitness correlation*).

Statistics collected during snowballing are included too, namely:

1. mean length of hill-climb to local optimum (*mean HC length*);
2. maximum length of hill-climb to local optimum (*maximum HC length*);
3. maximum number of paths to local optimum (*maximum HC paths*).

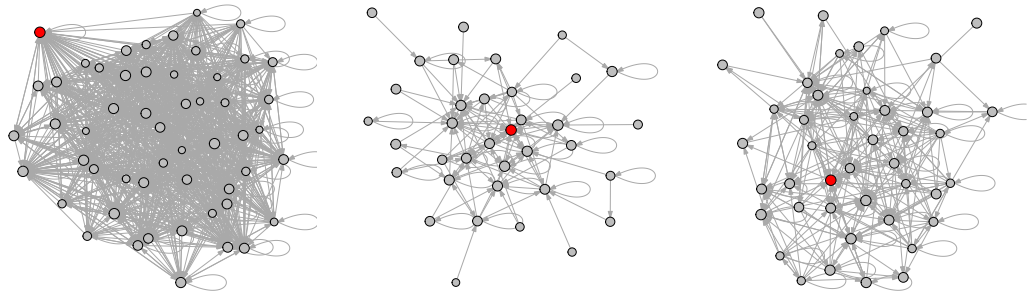
7.4.6 Features Applied

The features used in each experiment are now listed.

- Section 7.6.1 [Both LON types] : compressed local optima, diameter, edges, local optima, mean fitness, out-degree, sink fitness, sinks, sub-optimal sink strength
- Section 7.6.2 [Enumerated LONs] : edges, fitness correlation, incoming global, local optima, mean fitness, outdegree, weight disparity, weight loops
- Section 7.6.2 [ILS LONs] : edges, incoming global, local optima, mean fitness, outdegree, sink fitness
- Section 7.6.2 [Snowball LONs] : fitness correlation, maximum HC length, maximum HC paths, mean fitness, mean HC length, weight disparity, weight loops
- Section 7.6.3 [ILS LONs]: edges, incoming global, mean fitness, number optima, outdegree, sink fitness
- Section 7.6.3 [Snowball LONs] : fitness correlation, mean fitness, maximum HC length, maximum HC paths, mean HC length, weight disparity, weight loops
- Section 7.6.4 [ILS LONs] : diameter, edges, incoming global, local optima, mean fitness, outdegree, sink fitness
- Section 7.6.4 [Snowball LONs] : fitness correlation, maximum HC length, maximum HC paths, mean fitness, mean HC length, weight disparity, weight loops

– Section 7.6.5 [ILS LONs] : edges, local optima, mean fitness, sink fitness
– Section 7.6.5 [Snowball LONs] : fitness correlation, maximum HC length, maximum HC paths, mean fitness, mean HC length

Figure 7.3 shows three LONs; all are derived from a single QAP instance of size eleven, but are constructed using different algorithms. We can see that, visually, LON construction algorithms capture disparate phenomena in the fitness landscape. At a high level the ILS Sampling LON seems to be sparser in terms of nodes, edges, and edge density. In fact, the exhaustive enumeration LON has 53 nodes, 1134 edges; the ILS Sampling LON has 36 nodes, 122 edges; and Snowball Sampling generates a LON with 43 nodes, and 272 edges.



(A) Exhaustive Enumeration (B) ILS Sampling (C) Snowball Sampling

FIGURE 7.3: LONs extracted from a QAP instance with problem dimension of eleven. The three are produced using different construction algorithms, as indicated in the sub-captions. Node size is proportional to fitness (larger is fitter). The global optimum is shown in red with all other nodes in grey

7.5 Experimental Setup

In this Section I supply the setup for the experiments. This includes the LON construction algorithms, the predictive models, and the QAP metaheuristic algorithms used.

7.5.1 ILS Sampling

Seven parameter configurations are supplied to the ILS Sampling algorithm while it constructs LONs for the 124 considered QAPLIB instances. Parameter choices are amongst those provided and suggested by the author of the underlying ILS which forms part of the sampling algorithm [191], and are in the format [k, *pivot rule*]. They are: $[\frac{n}{16}, first]$; $[\frac{n}{16}, best]$; $[\frac{n}{8}, first]$; $[\frac{n}{8}, best]$; $[\frac{n}{4}, first]$; $[\frac{n}{2}, first]$; and $[\frac{3n}{4}, first]$. For all, 200 runs are started from a random solution; each terminates when there has not been an improvement in 1000 iterations. Seven sampling configurations per problem generates 868 ILS Sampling LONs in total for the studied QAPLIB instances. In addition, I construct ILS Sampling LONs for the 60 synthetic QAP instances and also for the fixed computational budget QAPLIB LONs using the configuration $[\frac{n}{2}, first]$.

That choice is attributable to the fact that the setting lends to LONs of a manageable size. In the regression analysis where ILS Sampling LON features are involved, the predictors — whose shorthand names have been stipulated in Section 7.4.5 — are:

1. *number optima*;
2. *edges*;
3. *mean fitness*;
4. *out-degree*;
5. *incoming global*;
6. *sink fitness*;
7. *diameter*.

7.5.2 Snowball Sampling

Parameter configurations for Snowball Sampling are in the format $[l, se, de]$ and are as follows: $[100, 20, 2]$; $[100, 30, 2]$; $[100, 50, 2]$; $[100, 60, 2]$; $[200, 30, 2]$; $[400, 30, 2]$; and $[400, 50, 2]$. The choices are based on those suggested by the algorithm's author [25]. Akin to ILS Sampling, there are 868 LONs (i.e. seven parameter sets $\times$ 124 instances) for QAPLIB produced by this algorithm. Snowball Sampling LONs for the 60 synthetic QAP instances and for the fixed computational budget QAPLIB LONs are constructed using $[100, 60, 2]$, which facilitates the production of reasonably small LONs. In the regression analysis where Snowball Sampling LON features are involved, the predictors — whose shorthand names have been stipulated in Section 7.4.5 — are:

1. *mean fitness*;
2. *fitness correlation*;
3. *weight loops*;
4. *weight disparity*;
5. *mean HC length*;
6. *maximum HC length*;
7. *maximum HC paths*.

7.5.3 Regression Model Setup

Model Setting I. I use linear and random forest regression for algorithm performance prediction in Sections 7.6.3-7.6.5 of the incumbent experimentation. Linear regression models describe the linear effect of predictors on the response variable; random forest is known for capturing non-linear interactions between variables. For each, random repeated sub-sampling cross-validation (also known as bootstrapping) is conducted for 100 iterations, each time shuffling the observations randomly. As asserted in Chapter 6, Pattengale *et al.* found that between 100 - 500 bootstrapping replicates provide the necessary information [178]; I use 100 for computational efficiency, after noticing that separate bootstrapping runs produce similar estimates and that increased iterations have a diminutive effect on the calculated estimates. An 80 - 20 training-test split is applied. The predictors *pre* are normalised and the standard deviation is reduced to one using the formula $\hat{pre} = \frac{(pre - E(pre))}{sd(pre)}$, where $E(pre)$ is

the expected value of *pre*. For the majority of models reported the R^2 is provided as a summary statistic. This quantifies the amount of variance in the response variable explainable using the set of predictors. Also in use is the mean-squared error (MSE), a notion of how accurate predictions from the model are. For the random forest regressions, 500 trees are used and predictor importance rankings are elucidated.

Model Setting II. A separate element of the experimentation concerns the comparison between sampled LONs and their fully-enumerated counterparts. This creates a different environment for modelling. Due to the fact that it must be possible to exhaustively enumerate the fitness landscape, the instances are bounded at size eleven. There are 60 observations (that is, feature vectors for 60 LONs) arising from instances belonging to two synthetic QAP classes: random "real-like" and uniform-random. In the modelling, possible effects attributable to the difference between problem classes should be adjusted for, in particular because of the limited number of observations. These so-called "random effects" can be controlled for in a hierarchical linear mixed model, also known as a mixed-effects model. This is *only* used for the experiments detailed in results Section 7.6.2. To formalise the hierarchical modelling approach, let us take y_{ik} to be a metaheuristic performance observed on instance i from class cl (cl is random "real-like" or uniform-random). The linear model is then:

$$y_{icl} = \delta_0 + \sum_{j=1}^{f} t\delta_j x_{jicl} + \alpha_{cl} + \zeta_{icl}, \qquad \zeta_{icl} \sim \mathcal{N}(0, \sigma^2)$$

where x_{jicl} is the value of predictor j (for example the number of local optima or number of funnels) of instance i from class cl, δ_j is its corresponding fixed effect on the response, α_{cl} are the random effects conditional on problem class cl (which represent random deviations from the common intercept δ_0), and finally ζ_{icl} are the model residuals. The summary statistics used for these regression models are conditional R^2, the marginal R^2, and the Root Mean Square Error (RMSE) as a proportion of the response variable's range. Conditional R^2 is the variation explained by the complete model. Marginal R^2 is the ratio of variation that is explained by exclusively the fixed effects [199] (negating or controlling for the random effects attributable to instance class). The fixed effects are LON features. RMSE captures the standard deviation of the model predictions (also known as *residuals*) and is therefore useful for estimating the variability of predictions (recall Section 2.7). RMSE is in the same unit range as the response variable. For ease of interpretation, in the results that follow RMSE is taken as a proportion of the total range for the response variable (that is, metaheuristic algorithm performance).

7.5.4 Metaheuristic Performance

Two competitive algorithms for the QAP are used to generate empirical difficulty information for the instances: Improved Iterated Local Search (ILS) [191] and Robust Tabu Search (ROTS) [190]. This ILS implementation comes with a wealth of potential parameter configurations. The choices for this study are first improvement hill-climbing in combination with a perturbation strength of $\frac{3n}{4}$ pairwise exchanges of items — a strength which is known to be well-performing [191] and is chosen in pursuit of achieving a sufficiently different incumbent solution when compared to the previous local optimum. The ROTS process and setup has been previously detailed in Section 6.5.4. Each of the metaheuristics executes 100 times on every QAP instance and after 1000 iterations the obtained fitness is taken as a proportion of the

optimal fitness. I use this proportion to quantify how challenging the instance is and this serves as a response variable in most of the upcoming models; it is referred to with $algorithm.p$, where $algorithm$ is the metaheuristic name. This metric is not suitable for the size eleven instances, though; they are straightforward to solve, meaning the obtained fitness is almost always the optimal fitness. For those instances and associated results, I instead consider the number of iterations to reach the global optimum as the algorithm response variable and use the notation $algorithm.t$ for that.

7.6 Results

7.6.1 Network Comparison

This Section investigates the similarity between LONs extracted by ILS Sampling and Snowball Sampling construction algorithms. Table 7.2 provides pairwise Spearman correlations calculated between ILS Sampling LON features and Snowball Sampling LON features. An indication of the p-value is given and is described in the caption. Network and funnel features are included as variables. These were introduced in Section 7.4.5.

TABLE 7.2: Spearman correlations between ILS Sampling LON features and Snowball Sampling LON features; $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

variable	correlation
local optima	0.239^{*}
edges	0.149
mean fitness	0.990^{***}
diameter	-0.212^{*}
out-degree	0.441^{***}
sinks	0.112
compressed local optima	0.249^{*}
sub-optimal sink strength	0.274^{**}
sink fitness	0.990^{***}

Most of the features have a fairly weak correlation, although consistently with an associated $p < 0.05$. This provides evidence against the null hypothesis which stipulates that features from the two samples are completely unrelated. The correlation for the out-degree of the samples is moderate, and has an encouragingly small $p < 0.001$. The two strongest associations concern the mean fitness in the samples and the mean fitness of the sink nodes (*sinkfitness*) respectively. These features show a great deal of agreement and therefore have very strong correlations with p values pointing towards statistical significance.

Table 7.3 appraises the extent to which LONs built by the two construction algorithms are tunable. This is done by looking at the ranges of values for important features of the obtained LONs; the entries in the Table are the minimum value of a feature, represented as a proportion of the maximum value.

Looking first at the ILS Sampling column, we can see that the values are very close to zero. This asserts that the smallest values are completely dwarfed in comparison to the largest. It follows that this construction algorithm is unpredictable with respect to direction in the fitness landscape and also in the emergent structure of the LON it builds. This fact renders the algorithm productive in the sense that

TABLE 7.3: The range of LON features, represented as the minimum value proportional to the maximum value

predictor	ILS Sampling	Snowball Sampling
local optima	0.0003	0.2499
edges	0.0003	0.2499
out-degree	0.0508	0.9999
sinks	0.0003	0.2499

there is no prescriptiveness or artificial direction choices. Note from the Table, for example, that there is a wide range of out-degree values; this hints at the algorithm identifying "hub-and-spoke" network architecture where it is present. Nonetheless, the variability of the construction algorithm could potentially lead to unreliable or tangential samples.

In contrast to these observations, the column showing the range information for Snowball Sampling LONs reveals much larger values (between 0.29 and 0.99, i.e. the range is small), which indicates that this algorithm produces a somewhat predictable and therefore tunable number of nodes and edges. The range for the out-degree is extremely narrow, a situation which arises from the nature of the algorithm mechanisms. This phenomenon may cause the omission of important connectivity between local optima in the LON and indeed "hubs" may never actually be reached and therefore included in the sample. "Hub" nodes are critical in a network topology because much of the flow through the network — in our case, metaheuristic paths in the landscape — routes towards them or from them.

7.6.2 Approximating the LON

Table 7.4 reports summary statistics for regression models associated with features of LONs which are constructed by means of exhaustive enumeration, ILS Sampling, and Snowball Sampling. The experimental environment for these models is detailed in **Model Setting II**, found in Section 7.5.3. In the R^2 columns of Table 7.4, notice that the models are somewhat weak (in terms of variance in algorithm performance being accounted for) after controlling for the random effects of the instance class differences. This is clear when comparing the smaller marginal R^2 values with the conditional R^2 ones.

TABLE 7.4: Mixed-effects model summary statistics where the feature vectors concern QAP instances of size eleven; marginal (fixed effects) R^2 is shown, alongside conditional (fixed and random effects) R^2 and Root Mean Squared Error (RMSE)

construction method	response variable	marginal R^2	conditional R^2	RMSE
exhaustive enumeration	*ILS.t*	0.159	0.832	0.21
ILS Sampling	*ILS.t*	0.406	0.406	0.20
Snowball Sampling	*ILS.t*	0.370	0.374	0.20
exhaustive enumeration	*ROTS.t*	0.148	0.719	0.18
ILS Sampling	*ROTS.t*	0.220	0.816	0.20
Snowball Sampling	*ROTS.t*	0.147	0.745	0.23

The quantity of algorithm performance which is explainable using the predictors

is likely low partially because of the limited number of observations, and smaller-still number which belong to a particular instance class (there are 30 each for random "real-like" and uniform-random problems). Notwithstanding the model weakness, observations can still be made from a relative standpoint concerning the LON construction methods which produced the predictors. In terms of marginal R^2 the two strongest model settings are the prediction of *ILS.t* response with ILS Sampling and Snowball Sampling LONs. This means that sampled LONs explain more variation than their fully-enumerated counterparts with respect to the ILS response variable. Preliminary intuition might dictate that the fully-enumerated LON would provide superior or more comprehensive information about the problem. Even so, ILS Sampling and Snowball Sampling include search operator combinations which are also used in practice in QAP optimisation. In contrast, the exhaustive enumeration construction algorithm simply hill-climbs from every possible solution to assign a local optimum. LON Edges are calculated mathematically as neighbourhood adjacencies, typically with respect to the most elemental operation; in addition, all possible connections are traced, no matter how improbable they may be. This paradigm may not mirror actual stochastic algorithm-problem interactions and therefore the emergent LONs might not match empirical algorithm performance. The result is interesting given that most LON literature concerns fully-enumerated LONs (see for example, [11, 5, 8]). Observe in the Table the strange fact that the marginal R^2 and conditional R^2 are equal in the case of ILS Sampling. This means that ILS Sampling LON features displayed no patterns which are attributable to the random effects caused by differing QAP instance classes.

7.6.3 Budgeted LON Construction

In this Section, LONs are produced using a computational budget which is intended to be closer to equivalent (than the algorithms would otherwise be). The setup for the regression used in these experiments is outlined in **Model Setting I** of Section 7.5.3. During the execution runs, construction algorithms are allowed 50,000 fitness function evaluations before the production process is brought to a close and the LON is finished. Under this experimental environment, I aim to investigate which construction algorithm generates LONs containing more predictive potency under these restricted computational conditions. Table 7.5 presents information about and summary statistics for models which involve features of LONs constructed using the algorithmic budget; this includes indication of construction algorithm, regression type (either linear or random forest), response variable (*ILS.p* or *ROTS.p*), the R^2, and the mean squared error (MSE).

Fix your attention on the rows with *ILS.p* as the response variable entry in Table 7.5. Notice that the MSE levels are comparable between LON construction algorithms; however, the R^2 values are higher with respect to ILS Sampling over Snowball Sampling (for both regression types). Nevertheless, I note that Snowball Sampling is superior to its competitor with respect to *ROTS.p* as the response variable for both regression types. The R^2 values are higher and the error rates are smaller in the Snowball Sampling models when compared to the equivalent ILS Sampling settings. In fact, using Snowball Sampling LON features as predictors for *ROTS.p* response with random forest regression creates the strongest model shown — this setting facilitates the explanation of around 52% of variance in the algorithm performance. The divergent nature of this model within the group is conspicuous when considering that, typically, the R^2 values are smaller for random forest regression over linear, and indeed this model is the only exception.

Tables 7.6 and 7.7 record random forest predictor rankings for regression models which incorporate features from ILS Sampling LONs and Snowball Sampling LONs respectively.

Progress your attention through Table 7.6 initially. The highest-ranking predictor for *ILS.p* is a feature relating to funnels — the proportional incoming edge strength to global optimum sinks, i.e. funnel floors. This is followed by the out-degree. Low-ranking predictors include the mean fitness and fitness of sinks. In the *ROTS.p* column the opposite phenomenon can be seen: fitness-focussed features rank highly while connectivity-focussed ones rank lower.

Now onto Table 7.7. The predictors are different here, as specified in Section-7.4.5. The maximum length of hill-climbs ranks highest for prediction of performance for both metaheuristic algorithms, and the mean fitness is also cardinal as a predictor in each. Overall, there is a clear dissimilarity between the orderings of the two columns; it follows that the influence of particular features on algorithm prediction is highly algorithm-dependent. Nonetheless, there are direct observations to be made. ILS Sampling LON fitness distributions are important for *ROTS.p* prediction. The maximum length of hill-climbs during the Snowball Sampling procedure matters for the performance of both metaheuristics.

7.6.4 Unbudgeted LON Construction

The set of 1,736 LON samplings (two construction algorithms $\times$ seven parameter settings $\times$ 124 instances) is involved in this Section. Each of these are the product of a process which runs according to the supplied sampling parameter settings; they are therefore not limited by the computational budget implemented earlier in Section 7.6.3. An example number of fitness evaluations taken by Snowball Sampling without a specified budget is 5,412,066; for ILS Sampling an example expenditure is 1,314,464 fitness evaluations. The setup for the regression used in these experiments is outlined in **Model Setting I** of Section 7.5.3. Table 7.8 shows algorithmic performance prediction model statistics where features of these LONs serve as predictors. The R^2 and MSE are presented.

With respect to linear regression for *ILS.p* response, the results show that neither construction algorithm produces features which can build a suitable model, as demonstrated by their low R^2 values. However, the equivalent random forest models are more robust, with 64.5% (ILS Sampling) and 80% (Snowball Sampling)

TABLE 7.5: Linear and random forest regression model summary statistics; predictors consist of LON features which are produced within a fixed algorithmic budget alongside some features of the procedure trajectory itself (in the case of Snowball Sampling); R_2 and mean-squared error are given

construction algorithm	regression type	response variable	R^2	MSE
ILS Sampling	linear	*ILS.p*	0.471	0.002
Snowball Sampling	linear	*ILS.p*	0.303	0.002
ILS Sampling	linear	*ROTS.p*	0.336	0.132
Snowball Sampling	linear	*ROTS.p*	0.418	0.024
ILS Sampling	random forest	*ILS.p*	0.245	0.002
Snowball Sampling	random forest	*ILS.p*	0.230	0.002
ILS Sampling	random forest	*ROTS.p*	0.154	0.144
Snowball Sampling	random forest	*ROTS.p*	0.521	0.013

TABLE 7.6: Variable importance rankings for the random forest models; predictors are features from ILS Sampling LONs built using a fixed algorithmic budget. Columns are labelled with the model response variable. Fitness features in italics

feature	$p(ILS)$	$p(ROTS)$
mean fitness	4	1
sink fitness	6	2
number optima	3	4
incoming global	1	6
out-degree	2	5
edges	5	3

TABLE 7.7: Variable importance rankings for the random forest models; predictors are features from Snowball Sampling LONs built using a fixed algorithmic budget. Columns are labelled with the model response variable. Fitness features in italics

feature	$p(ILS)$	$p(ROTS)$
mean fitness	3	2
fitness correlation	5	4
weight loops	6	6
weight disparity	2	5
maximum HC length	1	1
maximum HC paths	4	7
mean HC length	7	3

TABLE 7.8: Linear and random forest regression model summary statistics; predictors are produced with no-budget ILS Sampling and Snowball Sampling; R_2 and MSE are given

construction algorithm	regression type	response variable	R^2	MSE
ILS Sampling	linear	*ILS.p*	0.043	0.002
Snowball Sampling	linear	*ILS.p*	0.057	0.003
ILS Sampling	linear	*ROTS.p*	0.180	0.081
Snowball Sampling	linear	*ROTS.p*	0.252	0.029
ILS Sampling	random forest	*ILS.p*	0.645	0.000
Snowball Sampling	random forest	*ILS.p*	0.804	0.000
ILS Sampling	random forest	*ROTS.p*	**0.925**	0.008
Snowball Sampling	random forest	*ROTS.p*	**0.922**	0.003

of variance explained with diminutive associated MSE values. This could reflect the capacity of regression trees to capture non-linearity and complex interactions between predictors. The same trend is seen with respect to *ROTS.p* response in Table 7.8 — linear models are weak, with small R^2 and comparatively larger MSE. The random forest results are strong, with over 90% of variance explained by features of LONs produced by both LON construction algorithms. Snowball Sampling is slightly more proficient than ILS Sampling at generating LONs for *ILS.p* prediction; for *ROTS.p*, the two are roughly indistinguishable for predictive power — ILS

Sampling has slightly higher R^2 but also higher MSE.

Tables 7.9 and 7.10 record predictor rankings for random forest regressions. The models contain features of LONs which are built without an algorithmic budget and also some features of the procedure trajectory itself (in the case of Snowball Sampling). They cover ILS Sampling and Snowball Sampling respectively.

TABLE 7.9: Variable importance rankings for the random forest models; predictors are features from ILS Sampling LONs built without a fixed algorithmic budget. Columns are labelled with the model response variable. Fitness features in italics

feature	$p(ILS)$	$p(ROTS)$
mean fitness	1	1
sink fitness	2	2
number optima	3	5
incoming global	4	7
out-degree	7	6
edges	5	3
diameter	6	4

TABLE 7.10: Variable importance rankings for the random forest models; predictors are features from Snowball Sampling LONs built without a fixed algorithmic budget. Columns are labelled with the model response variable. Fitness features in italics

feature	$p(ILS)$	$p(ROTS)$
mean fitness	1	1
fitness correlation	3	2
weight loops	7	7
weight disparity	5	4
maximum HC length	6	5
maximum HC paths	2	3
mean HC length	4	6

Immediately apparent is the importance of the sampled local optima level fitness distribution. In fact, notice across the two Tables that mean fitness is the top predictor for all four model settings. Sink fitness is the second highest-ranking predictor for both models which consider ILS Sampling features, emphasising the role of sampled fitness levels in prediction. Out-degree is insignificant within the Table 7.9 rankings, placing lowest and second-lowest.

Focus your attention on Table 7.10 in particular now. Fitness correlation appears to be salient, in conjunction with mean fitness and maximum number of hill-climbing paths. Ranking lowest is weight of self-loops, which pertains to unsuccessful escape attempts from local optima. The pattern of fitness features being among the most crucial is unusual: the bulk of LON literature focuses heavily on edge connectivity and density in the network (see for example [8, 176, 167]) rather than the local optima level fitness distribution which has been sampled.

7.6.5 Predictive Potency with "Super Sampling"

The construction algorithms under study produce LONs which can capture separate fitness landscape information. Features which concern edge and node density, for example, are redundant as predictors if they are extracted from Snowball Sampling LONs — they are preordained by choice of the sampling algorithm parameters. I stipulate that combining features from LONs which are built using independent construction algorithms in regression may result in stronger models than using features derived from a single construction algorithm. The three most important predictors (according to Section 7.6.4) for each response variable (*ILS.p* and *ROTS.p*) and each construction algorithm are united together into predictive models. Settings for the regression used in these experiments are outlined in **Model Setting I** of Section 7.5.3. Model summary statistics are presented in Table 7.11; variable importance rankings are displayed in Table 7.12.

TABLE 7.11: Model summary statistics; predictors are from ILS Sampling LONs and also Snowball Sampling; they are the most important three features from LONs produced by each method (according to random forest predictor rankings) with respect to the results in the previous Section

regression type	response variable	R^2	MSE
linear	*ILS.p*	0.076	0.003
linear	*ROTS.p*	0.254	0.026
random forest	*ILS.p*	**0.972**	**0.000**
random forest	*ROTS.p*	**0.991**	**0.000**

TABLE 7.12: Variable importance rankings for the random forest models; variables are features of ILS Sampling and Snowball Sampling (models contain a combination of both). Columns are labelled with the model response variable. SS in parentheses beside a feature means that it is calculated on Snowball Sampling LONs; ILS means it is calculated from ILS Sampling LONs. Fitness features in italics

feature	$p(ILS)$	feature	$p(ROTS)$
mean fitness (SS)	1	*mean fitness* (SS)	1
sink fitness (ILS)	2	*sink fitness* (ILS)	2
mean fitness (ILS)	3	*mean fitness* (ILS)	3
number optima (ILS)	4	edges (ILS)	6
maximum HC length (SS)	5	maximum HC paths (SS)	4
mean HC length (SS)	6	*fitness correlation* (SS)	5

Table 7.11 illuminates the notion that linear regression may not suit performance prediction with LONs: the models have low associated R^2 values. This is in contrast to the equivalent random forest settings — these are markedly stronger than the linear models. In fact, they account for approximately 97% (*ILS.p*) and 99% (*ROTS.p*) of algorithm performance. The error rates in the MSE column are low relative to the linear setups. The R^2 values surpass those seen when using features associated with the product of a single LON construction algorithm.

As in Section 7.6.4, the dominance of the fitness features is obvious in Table 7.12. Both models share the same three fitness-focussed highest-ranked features: the mean

fitness in Snowball Sampling LONs is first; in second place is sink fitness, which is calculated on ILS Sampling LONs only; in third place is the ILS Sampling mean fitness. As in Section 7.6.4, the role of LON connectivity — the number of edges here — is comparatively weak.

7.7 Discussion

ILS Sampling LONs have not been validated against exhaustively-enumerated (complete) LONs before. Complete LONs can be regarded as a "ground truth" for the local optima level in a fitness landscape. They concern a foundational connectivity, with respect to basic best-improvement hill-climbing, between local optima in terms of search operation. Section 7.6.2 showed us that for QAP instances of size eleven, the two sampling construction algorithms generate LONs with more predictive power than those built by exhaustive enumeration. This is important because the vast majority of LON research (see for example [11, 5, 4, 8]) considers in their analysis exhaustively-enumerated LONs, and indeed their features have been used for algorithm performance prediction [17, 126, 200]. The results in this Chapter suggest that sampling construction algorithms for LONs may better approximate or infer prospective fitness landscapes when compared to the best-improvement local search procedure employed by the exhaustive enumeration algorithm. Observe that because sampling-based LON construction algorithms are either augmented with existing metaheuristics or use similar procedures to them, the patterns of search operation — and therefore reactions with the problem space — which are encoded in the LON object align with those seen in empirical metaheuristic performance.

Although this is encouraging with respect to the utility of sampling algorithms, caution must be exercised when extrapolating these results to larger problems. The sampling algorithms I have studied are designed for large problems but the "ground truth" comparison presented in Section 7.6.2 is limited to small problems because of computational limits. In particular, I argue that Snowball Sampling may not be appropriate for certain types of landscape analysis on large problems (although this is based on deductive thought and not on experimentation): the algorithm traverses short, restricted paths through local optima and is not based on the trajectory of a stochastic metaheuristic process. Therefore, it is argued Snowball Sampling may be highly dependent on the random starting solution — the rest of the sample is based around that. In a large fitness landscape, the obtained sample may actually be rather low-quality local optima, far from the promising regions near the global optimum. A promising avenue which I leave for future work is investigation of the relationship between LON sampling parameter choices and problem dimension for both algorithms. In Section 7.6.3 we saw that supplying the LON construction algorithms with a budget of 50 000 fitness function evaluations produces LONs whose features can explain up to 52% of algorithm performance. Snowball Sampling generated LONs more closely related to Tabu Search performance then ILS Sampling; ILS Sampling LONs were slightly more associated with Iterated Local Search as a response variable.

LONs which were constructed without an algorithmic budget demonstrated considerable predictive potency concerning the considered metaheuristics in Section 7.6.4. Both ILS Sampling and Snowball Sampling seem to produce LONs that infer prospective fitness landscapes well, with a substantial amount of variance in their performance being explainable using the LON features. Sections 7.6.4 and 7.6.5 presented

the fact that random forest trees yield more promising regression results than a linear setting does for this type of predictive model. This provides evidence supporting the use of random forest in future algorithm performance prediction models using fitness landscape features. I stipulate that this should be done in pursuit of capturing complex variable interactions. Another aspect of the results is the apparent importance of fitness levels sampled by the LON construction algorithms, which we noticed in Sections 7.6.4 and 7.6.5. Features concerning fitness distribution of the samples were repeatedly among the top predictors for random forest models.

The strongest models which are reported in this Chapter combine together features of LONs produced using both ILS Sampling and Snowball Sampling. These were presented in Section 7.6.5, and the predictors accounted for the large majority of algorithm performance when using random forest regression. It follows that involving a LON "super sample" (that is, samples drawn by different procedures) in regression is promising for algorithm explanation and prediction.

There are certainly limitations to the approach and indeed the results presented: the results are valid, but currently hold only for the search operators chosen, the configurations for the LON construction algorithms, choices of QAP metaheuristic algorithms, specification of algorithm performance metric, neighbourhood function, and the particular features extracted from the LONs. There is also the potential issue of stochasticity in the sampling algorithms.

7.8 Conclusions

In this Chapter, LON construction algorithms have been scrutinised to gain information about their ability to infer possible prospective fitness landscapes (i.e. landscapes which may arise with the use of metaheuristics on the instances). The two most recent sampling algorithms from the literature are included, with the aim of comparing them and investigating the quality of the resultant LON samples for predicting algorithm performance on the underlying problems. I additionally involve an exhaustive enumeration approach for building LONs. All three construction procedures are applied to the Quadratic Assignment Problem here, but their algorithmic templates could easily be applied to an arbitrary combinatorial problem. The QAP instance set comprised both benchmark and generated problems for this Chapter.

I found that from the initial network comparison that the two construction methods exhibit some agreement in the LONs they build and that, therefore, they do not generate completely unrelated samples. The algorithms are distinct from each other in some foundational ways: Snowball Sampling is tunable and predictable, while ILS Sampling varies widely but appears adept at identifying the true emergent structure of the LON in the fitness landscape. I then conducted a comparative appraisal of the predictive power for the LON samples and the "ground truth" exhaustively-enumerated counterpart. Then the sampling algorithms were provided a budget of 50 000 fitness evaluations to produce LONs. The relationship of the resultant LONs to algorithm performance was ascertained. After that, the construction algorithms were supplied with several sampling parameter configurations and were not limited by an algorithmic budget. QAPLIB instances with $N \leq 128$ were each mapped to fourteen LONs (two construction algorithms $\times$ seven parameter configurations) and features of these were observations in performance prediction models. Finally, the most important variables for each sampling algorithm were combined together to form a "super sample" and then used as predictors in modelling.

The results suggest that random forest trees better capture the non-linear relationships between variables in fitness landscapes. The apparent salient role of fitness layers at the local optima level in explaining metaheuristic performance was evident in the analysis. A lot of LON literature focuses on edge connectivity patterns and not the fitness distribution amongst local optima; I present evidence that this may be misguided. An interesting result was that exhaustively-enumerated LONs have less predictive potency than their sampled counterparts. One may initially imagine that a more accurate prediction could be obtained from more extensive fitness landscape information (i.e. exhaustive enumeration). This is not the case here. The result suggests "quality over quantity". I conclude this study with a note that much work remains to be done in this field. In particular, I acknowledge that the results are of course dependent on the algorithm configurations, neighbourhood function choice, choices for search algorithms, choice of problem domain, and so on. The pursuit of different choices is left for future work.

7.9 Summary

The present Chapter aimed to understand LON construction algorithms more intelligibly. The prospect of analysing future real-world problems is among the motivations for doing so. In the next piece of work, I move towards modelling non-benchmark problems with LONs, using a highly-constrained problem from healthcare. Although there are certain abstractions in the formulation which deviate from real life, departing from well-known benchmark optimisation problems is necessary for the evolution of LON analysis.

Chapter 8

The Local Optima Level in Chemotherapy Schedule Optimisation

Chapter 7 conducted an appraisal of LON construction processes with the aim that they can be better understood and refined when applied to unknown problems in the future. The present Chapter takes a step in the direction of modelling the reactions between non-benchmark combinatorial problems and metaheuristic search algorithms using LON construction.

8.1 Abstract

In this Chapter a multi-drug Chemotherapy Schedule Optimisation Problem (CSOP) is subject to LON design. CSOPs have not previously been a testbed for fitness landscape analysis. I fill this gap; LONs are constructed and studied for meaningful structure. The CSOP formulation brings a topic from healthcare to LON analysis, and presents novel challenges and questions for the model because there are infeasible regions in the fitness landscape and an unknown global optimum. Two LON construction algorithms are proposed for sampling CSOP fitness landscapes: ILS Sampling and Memetic Sampling. The results provide new insight into LONs which are associated with highly-constrained problems, and into the proficiency of search operators on the CSOP. ILS and Memetic Search, which form the foundations for the LON construction procedures, are found to markedly out-perform a GA from the literature.

8.2 Introduction

Analysis of LONs provides insight into how optimisation problems and search algorithms interact together. They capture *global* patterns at the local optima level in landscapes and have mostly been extracted for benchmark combinatorial optimisation problems such as NK Landscapes [11, 50, 8], the QAP [5, 7, 25], and the Travelling Salesman Problem [136, 23, 142].

Studies in non-benchmark problem domains have been sparse and have consisted of computational protein modelling [59] and feature selection [14]. These were steps towards bringing LON analysis to unmapped real-world problems. This type of case study, demonstrating LON efficacy, is needed for convincing possible industry collaborators. Large and highly-constrained problems should ideally be used in case studies (by "large" I mean hundreds of dimensions), in pursuit of simulating environments typical of real-world optimisation problems.

Chemotherapy Schedule Optimisation Problem (CSOP) [201] instances have been the subject of several research papers in evolutionary computation [202, 203, 204, 205, 206]. One instance was formulated to reflect real-life chemotherapy drug response closely [202] and the tumour shrinkage model used in the fitness function has been subject to extensive clinical testing [207]. The instance, alongside other CSOP formulations, has not been subject to fitness landscape analysis (although some authors have made passing remarks about CSOP landscapes [208, 209]).

I conduct a first fitness landscape analysis on CSOP, focussing on local optima connectivity with the use of LONs. Two LON construction algorithms are proposed — the first has ILS as its foundation; the second has Memetic Search (MS) as the foundation. LONs are then produced and their attributes and fitness distributions are compared. A study of the feasibility trajectories in the LONs is also presented. Later on, algorithm performance results suggest that the search algorithms which underlie the construction processes outperform a Genetic Algorithm (GA) from the literature for the CSOP. In summary, this Chapter contributes to the literature in the following ways:

1. A first fitness landscape analysis of CSOP, lending to new insights of the problem interacting with search operators;
2. The presence of infeasible solutions in the landscapes is new to LON research;
3. LON construction algorithms for the CSOP are proposed (which can also be easily applied to an arbitrary binary-encoded problem);
4. Metaheuristic search algorithms are offered which outperform a GA from the literature (ILS and MS; a separate MS has been used on a CSOP formulation before but with a different fitness function, different constraints, and different solution encoding [208]).

8.3 Background

I use a multi-drug CSOP which was initially formulated and described in 1998 [202] and then further studied in later research [210, 203, 204, 205, 206]. As asserted in the original paper, a multi-drug CSOP can have a binary representation where each gene, i, is set iff a particular concentration of a particular drug (of number n_{drugs}) is administered at a particular time interval (t, from within defined time intervals). As suggested in the literature [210], I set the number of drugs $n_{drugs} = 10$ and the number of time intervals for doses, t_{int}, also at 10. There are four bits, so sixteen allowed concentrations for each drug, $conc = 4$, giving each binary solution a length of 400, i.e. $n_{drugs} \times t_{int} \times conc$. The number of possible solutions, and the size of the configuration space, is extremely large at 2^{400}.

8.3.1 Fitness Function

I consider curative chemotherapy treatment here, meaning tumour eradication is the aim. This is the primary (and only) objective. For this single-objective case, fitness is calculated with respect to the chemotherapy schedule *minimising* tumour size (in number of cells). This is done through *maximising* the combined effect of drugs in the schedule against the tumour. In considering the tumour's shrinkage response, a mathematical function is needed. The most popular model in the literature is called

the *Gompertz Growth Model* [207], which has a linear cell-loss effect and has been validated by significant clinical experiments. The formula is given in Equation 8.1:

$$\frac{dN}{dt} = N(t)\left(\sigma \ln\left(\frac{\theta}{N(t)}\right) - \sum_{j=1}^{d} ef_j \sum_{n}^{i=1} C_{ij}\left\{H(t-t_i) - H(t-t_{i+1})\right\}\right) \tag{8.1}$$

with the components are as follows: $N(t)$ is the cancerous cell count at time interval t; σ and θ are parameters pertaining to tumour growth; $H(t)$ is the Heaviside step function [211], which evaluates to zero when $t \leq 0$ and one otherwise; ef_j denotes the efficacy of chemotherapy drugs; and C_{ij} is the concentration levels of the drugs administered.

The actual fitness function is quite complex, including penalties based on feasibility distances, and I refer the interested reader to a comprehensive description [202] (pp. 106 - 107). In essence, initial fitness is calculated with respect to the total impact on the tumour for the treatment schedule. Individual impacts for each drug are known. The objective is to maximise the combined impact of all the drugs in the schedule (at the specified concentrations, and at the specified time-slots). The *maximisation* of this will *minimise* the tumour.

Following the drug impact fitness calculation, the solution is checked for constraint violations and the fitness is penalised accordingly (see [202] for details). Any violation will result in a fitness below zero. A feasible solution has fitness zero or above. The constraints are as follows: the tumour is not allowed above a particular size; the maximum cumulative dose of drugs cannot exceed the specified limits for each individual drug; and the limit on toxic chemotherapy side-effects cannot be exceeded (for each time interval). In all cases the magnitude of the violation is captured through proportional subtraction from the fitness sum.

Mathematically the fitness function is subject to these constraints:

- Maximum allowable cumulative C_{cum} dosage for each drug:

$$g_1(c) = \left\{ C_{cum\,j} - \sum_{i=1}^{n} C_{ij} \geq 0 \vdots \forall j \in \overline{1,d} \right\} \tag{8.2}$$

- Maximum allowable size of the tumour, i.e. number of cancerous cells, N:

$$g_2(c) = \left\{ N_{max} - N(t_i) \geq 0 \vdots \forall i \in \overline{1,n} \right\} \tag{8.3}$$

- A threshold for the known toxic side-effects of using multiple drugs in chemotherapy treatment:

$$g_3(c) = \left\{ C_{s-e_z} - \sum_{j=1}^{d} \eta_{zj} C_{ij} \geq 0 \vdots \forall i \in \overline{1,n}, \forall z \in \overline{1,m} \right\} \tag{8.4}$$

In the constraint seen in Equation 8.4, the variable η_{zj} is the known possibility of harming the z^{th} organ (for example, the heart) through administering the j^{th} drug.

8.3.2 Metaheuristic Search Algorithms

Evolutionary algorithms have been used with success for CSOPs; in particular, GAs have dominated [202, 210, 212, 213], although other approaches have been utilised,

such as Estimation of Distribution Algorithms [204, 206]; SA variants [214, 213]; Memetic Algorithms (MAs) [208]; and Evolutionary Strategies [213, 215]. A GA from the literature [206] is used as the foundation for the Memetic Sampling LON Construction algorithm proposed here (detailed later in Section 8.5) and is also used later on in conducting optimisation on the problem to collect search difficulty information.

8.4 Methodology

This section describes the LON construction algorithms proposed for studying CSOP fitness landscapes. The aim is examining the topological features which form when optimisation search operators are used on the CSOP configuration space.

8.5 LON Construction Algorithms

8.5.1 Iterated Local Search Sampling (ILS Sampling)

To align with existing LON construction algorithms for benchmark domains [19, 24, 25] I implement an algorithm using ILS as the vehicle. This metaheuristic combines the intensification of local search with the exploratory nature of perturbation. ILS is naturally suited to constructing LONs because each iteration identifies a transformation between local optima and this can be added as an edge to the LON. I refer to the ILS-driven LON construction procedure as ILS Sampling. The general schema for the algorithmic process used in this Chapter mirrors that seen in Algorithm 6, which has already been comprehensively explained in Section 7.4.3. There are, however, some minor adjustments to the design in this Chapter; the pseudocode is given in Algorithm 7. Hill-climbing in this procedure is best-improvement and a solution is deemed a local optimum after 100 iterations of that. This value was chosen after observation of initial runs in order to reduce runtimes. Perturbation is thirty bit-flips.

Within Algorithm 6 only improving local optima are accepted during the search and the resultant LON edges represent the acceptance of an improving local optimum (that is, an edge is constructed only when an improvement in fitness at the local optima level is identified). In Algorithm 7, notice that improving or equal local optima are always accepted, but deteriorating local optima are also accepted with a 10% probability (this can be observed at the **if** statement execution block beginning at line 16). The status of acceptance is captured with the *accepted* flag in the Algorithm pseudocode. If the local optimum is accepted as the new working solution, then an edge is added to the LON — despite the reality that this is sometimes directed towards lower-quality fitness. I designed the algorithm this way after experimenting with the local optima acceptance conditions during preliminary runs and noticing that allowing some deteriorating moves aided in fitness advancement. As with the previous algorithm, ILS runs terminate after t iterations and nodes and edges from r runs are amalgamated together to form a single LON for the problem. Parameters are stated shortly in Section 8.7.

8.5.2 Memetic Search Sampling (MS Sampling)

I propose and implement Memetic Search Sampling (MS Sampling) for the CSOP here. Recall from Section 2.3 that the term *Memetic* means the combination of more

Algorithm 7 CSOP ILS Sampling

Input: perturbation strength k, stopping condition t, number of runs r
Output: a LON sample

```
1: procedure SAMPLING(search space S, fitness function f, perturbation strength k,
   stopping condition t, number of runs r):
2:     runs ← 0, nodes ← ∅, edges ← ∅
3:     repeat
4:         Choose initial random solution s ∈ S
5:         local.optimum ← HILL.CLIMBING(s)
6:         iterations ← 0
7:         repeat
8:             current.solution ← PERTURB(previous.local.optimum, k)
9:             new.local.optimum ← HILL.CLIMBING(current.solution)
10:            nodes ← nodes ∪ {new.local.optimum}
11:            accepted ← 0
12:            if f(new.local.optimum) ⩽ f(previous.local.optimum) then:
13:                previous.local.optimum ← new.local.optimum
14:                accepted ← 1
15:            else
16:                if random.float ⩽ 0.1 then:
17:                    previous.local.optimum ← new.local.optimum
18:                    accepted ← 1
19:            if accepted == 1 then:
20:                if (previous.local.optimum, new.local.optimum) ∈ edges then:
21:                    edge.weight ← edge.weight + 1
22:                else
23:                    edge.weight ← 1
24:                edges ← edges ∪ {[previous.local.optimum, new.local.optimum, edge.weight]}
25:            iterations ← iterations + 1
26:        until iterations ≥ t
27:        return previous.local.optimum
28:    until runs == r
```

than one algorithmic strategy, and that a typical example is merging a GA (composed of selection, recombination, and mutation) with local search. LONs have been constructed with respect to memetic operations before, although in different forms and contexts — in continuous optimisation, where a LON edge constitutes a memetic differential evolution iteration [64]; for the Asymmetric Travelling Salesman Problem, with an edge representing the application of a domain-specific crossover operation followed by k-opt local search [23]; and in genetic improvement of software programs, where an edge denotes a uniform crossover followed by best-improvement local search [141].

Algorithm 8 MS Sampling

Input: percent fittest individuals pf, length evolution path lp
Output: a LON sample

procedure MS.SAMPLING(population size ps, generations g, percent fittest individuals pf, mutation probability mp, crossover probability cp, length evolution path lp, search space S, fitness function f):
    $nodes \leftarrow \emptyset, edges \leftarrow \emptyset$
    $population \leftarrow$ RANDOM.POPULATION(S, ps)
    $iterations \leftarrow 0$
    repeat
        $population \leftarrow$ GENETIC.ALGORITHM($population$, ps, cp, mp)
        $fittest \leftarrow$SELECT.FITTEST($population$, pf)
        for $solution \in fittest$ **do**
            $solution \leftarrow$ HILL.CLIMBING($solution$)
        $nodes \leftarrow \{[fittest]\}$
        for $parent_1 \in fittest$ **do**
            for $parent_2 \in fittest$ **do**
                $offspring_1, offspring_2 \leftarrow$ MEMETIC.EVOLUTION($parent_1, parent_2$)
                $iterations \leftarrow iterations + 1$
    until $iterations \geq g$
    for $parent_1 \in nodes$ **do**
        for $parent_2 \in nodes$ **do**
            $steps \leftarrow 0$
            repeat
                $offspring_1, offspring_2 \leftarrow$ MEMETIC.EVOLUTION($parent_1, parent_2$)
                $parent_1, parent_2 \leftarrow offspring_1, offspring_2$
                $steps \leftarrow steps + 1$
            until $steps \geq lp$

procedure GENETIC.ALGORITHM($population$, ps, cp, mp):
    repeat
        $parent_1$, $parent_2 \leftarrow$ SELECTION($population$)
        if cp **then**:
            $offspring_1, offspring_2 \leftarrow$ CROSSOVER($parent_1$, $parent_2$)
        if mp **then**:
            $offspring_1, offspring_2 \leftarrow$ MUTATION($offspring_1, offspring_2$)
        $population[parent_1] \leftarrow offspring_1$
        $population[parent_2] \leftarrow offspring_2$
    until $iterations \geq ps/2$

procedure MEMETIC.EVOLUTION($parent_1, parent_2$):
    $child_1, child_2 \leftarrow$ CROSSOVER($parent_1, parent_2$)
    if mp **then**:
        $offspring_1, offspring_2 \leftarrow$ MUTATION($offspring_1, offspring_2$)
    $offspring_1, offspring_2 \leftarrow$ HILL.CLIMBING($offspring_1, offspring_2$)
    $nodes \leftarrow nodes \cup \{offspring_1, offspring_2\}$
    $candidate.edges \leftarrow \{[parent_1 \rightarrow offspring_1], [parent_1 \rightarrow offspring_2], [parent_2 \rightarrow offspring_1], [parent_2 \rightarrow offspring_2]\}$
    $weights \leftarrow \emptyset$
    for each $edge$ in $candidate.edges$:
        $weights[edge] \leftarrow 1$ **if** $edge \notin edges$ **else** $weights[edge] \leftarrow$ existing weight of $edge \in edges + 1$
    $edges \leftarrow edges \cup \{[candidate.edges, weights]\}$

    return $offspring_1, offspring_2$

GAs have been successful in finding approximate solutions for CSOPs before [205]; it follows that a GA is a reasonable foundation for a LON construction algorithm. The procedure I enact originates from a generational GA used in previous literature [203]. By definition LONs contain only local optima, but GAs do not guarantee local optima in the population; this necessitates the addition of local search to the system, resulting in a Memetic Search (MS) or algorithm. The Memetic Search-driven LON tracking process is hereafter referred to as Memetic Search Sampling (MS Sampling for short). Pseudocode for MS Sampling is shown in Algorithm 8.

A random starting population is generated initially, at line 3. Then g generations of the genetic algorithm execute — observe line 6 where the call is made to the `GENETIC.ALGORITHM` function.

Inside the `GENETIC.ALGORITHM` procedure, which originates from previous literature [203] , the following is repeated $ps/2$ times (ps is the population size): parents are selected (line 26) and recombined before the resultant offspring are subject to mutation. The offspring replace the parents inside the population (line 31). The next element in the `MS.SAMPLING` procedure selects a percentage pf of the population which are the fittest; focus on line 7 of the first function. Each of the fittest are subject to hill-climbing at line 9, transforming them into local optima. Hill-climbing in this case is best-improvement on a single bit-slip neighbourhood and a solution is deemed locally optimal when it is the incumbent solution after 100 iterations of this. The local optima are added to the LON at line 10. Pairs of them are subsequently passed to the `MEMETIC.EVOLUTION` function as $parent_1$ and $parent_2$ (line 13). Inside that, which begins at line 34, they are then deterministically recombined with one another (line 35); resultant offspring are mutated according to the mutation rate mp, before being subject to local search (line 38) and then added as LON nodes at line 39. After that, the weights are calculated for the four directed edges which are associated with the recombination of $parent_1$ and $parent_2$ which produces $offspring_1$ and $offspring_2$ — notice lines 42-43. Following this, the edges and their weights are written into the LON at line 44.

After g generations, the LON nodes undergo another evolutionary process, back at line 20. For each pairwise combination of nodes, the local optima are passed into the `MEMETIC.EVOLUTION` procedure. Inside that, as before, they are recombined and the offspring are subject to mutation and to local search. Those are added as nodes and the transitions as edges. The offspring subsequently become the parents for the next iteration of the same process; notice that the output of the call to `MEMETIC.EVOLUTION` — $offspring_1$ and $offspring_2$ — are assigned as $parent_1$ and $parent_2$ at line 21. This set of instructions repeats lp times; in this way, each pair of original LON nodes (identified during the g generations of genetic and memetic search) are the ancestors in an lp-generation evolutionary trajectory. This was a deliberate design choice: it guarantees the inclusion of sequences of evolution for local optima inside the LON. I noticed that without this algorithmic element during construction, the LON consists of many isolated pairs of nodes and proves difficult to study for meaningful network structure.

8.6 Visualisations

Visual analysis of LONs can provide an abstracted view of the local optima space in a fitness landscape, which is a high-dimensional complex system. Sometimes, patterns observable in high-level visual analysis help to explain metaheuristic algorithm performance on the associated combinatorial problem.

ILS Sampling and MS Sampling produce networks with thousands of nodes. To facilitate meaningful visualisation, pruned sub-networks are constructed. The "elite" nodes which are present in the LONs are chosen for this. For the ILS Sampling LON these are local optima in the top 2% of the sample. The MS Sampling LON has more nodes; as such, only the top 0.05% are visualised. It follows that examination of these particular landscape sections may lift the veil on the emergent architecture present among high-quality local optima in the CSOP.

Figure 8.1 shows plots for the ILS Sampling LON and MS Sampling LON, respectively. Edges encode sequences of search operations. In the case of ILS Sampling the sequence comprises perturbation $\longrightarrow$ local search; for MS Sampling, this is recombination $\longrightarrow$ probabilistic random mutation $\longrightarrow$ local search. Nodes with the highest fitness in the sample are coloured red; all other local optima are grey.

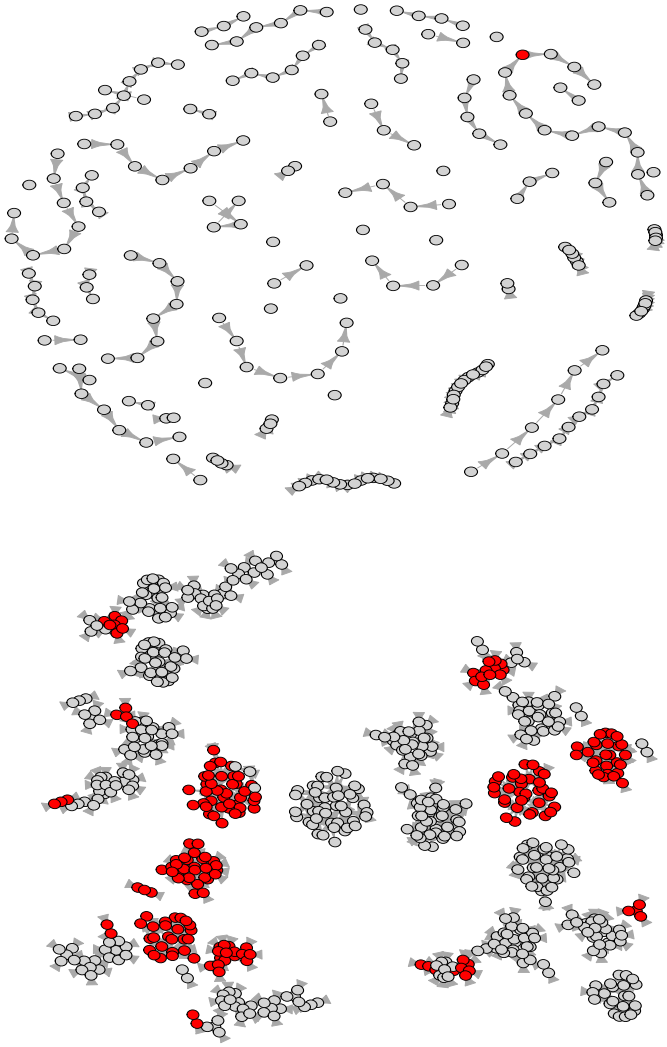


FIGURE 8.1: LONs extracted from a CSOP instance. In the higher plot the fittest 2% of local optima in the ILS Sampling network are shown, while the top 0.05% of the MS Sampling network are plotted in the lower image. Nodes with the highest fitness in the sample are coloured red, while all other local optima are grey. For the ILS Sampling LON the highest fitness is 1.707677. For the MS Sampling LON, it is 1.707826

Examining the Figure, note that ILS Sampling generates the sparser network. There are neat sequences of local optima and nodes typically have one incoming edge and one outgoing edge. Sequences are separate, in that they do not have bridges connecting them. The highest-fitness node, which is coloured red, is located within such a sequence. Visual appraisal of this structure within the object as a whole implies that the node would only be reached if the ILS arrived by coincidence at that particular sequence of local optima. The MS Sampling LON is denser and instead of linear sequences, clusters of nodes are seen. Some clusters are connected to other clusters and some are isolated. There are many distinct solutions which have the highest fitness in the sample and these are found in different groups. The presence of clusters instead of linear sequences hints that at lower fitness levels (which are not plotted because of the fitness threshold) the clusters are larger. This, along with the fact that there are multiple top-fitness nodes spread across the network, presumably provides more opportunity for connection to a node with the highest fitness than we saw for ILS Sampling.

8.7 Experimental Setup

8.7.1 Assumed Optimal Fitness

The global optimum is not known for the problem under study. In previous LON research there has always been a known optimum fitness. To simulate this for the CSOP at hand, I conduct several runs of the GA, MS, and ILS metaheuristics and consider the highest obtained fitness across all runs to be the *pseudo*-optimal fitness. This value is 1.71.

8.7.2 ILS Sampling

As stipulated in Section 8.5, ILS Sampling is an ILS framework. As such, local search handles intensification and the perturbation mechanism contributes diversification. The initial intention was to mirror parameter choices in previous LON construction works (such as [24]) but those choices were for much smaller search space sizes and the computation was therefore more feasible for their circumstance. The sampling parameters in the present study, as well as the ILS design, are chosen through observation of preliminary tests, with the aim that the emergent LON be generated within a reasonable timeframe; that it contains sufficient node and edge density to facilitate proper study of the network; and that it comprises high-quality solutions (the fitness of solutions are checked against the pseudo-optimal fitness described in the previous Section).

Local search in the system uses a single bit-flip operator and best-improvement as the pivot rule. A solution is deemed a local optimum after 100 iterations of that. Perturbation is thirty bit-flips and improving local optima are always accepted. Deteriorating local optima are accepted 10% of the time and individual runs terminate after 1000 iterations. Thirty independent ILS runs are conducted, with each accepted local optimum added as a LON node and each transformation between two local optima added as a LON edge. The parameters are shown in Table 8.1.

As well as ILS Sampling, whose purpose is the construction of LONs, I extract the ILS foundation from the algorithm to use as a metaheuristic for collecting search information about the problem. The algorithmic setting remains the same except the best-improvement pivot rule changes to the best of 100; this is in pursuit of efficiency, because I execute the metaheuristic 100 times on the instance.

TABLE 8.1: ILS Sampling design and sampling parameter settings

parameter	value
local search iterations	100
termination iterations, t	1000
pivot rule	best-improvement
neighbourhood	1 bit-flip
perturbation strength, k	30 bit-flips
number of runs, r	30

8.7.3 MS Sampling

MS Sampling is instrumented on top of a GA for the domain ([202]). A previous study using statistical inference found that only two GA parameters are significant with respect to this CSOP instance when solutions are binary-encoded: crossover probability cp and mutation probability mp [210]. Inside the MS Sampling designed in this Chapter, those parameters are the ones recommended in their paper (cp = 0.614, mp = 0.198); the others are from a related study [203] (which used integer encoding for the problem), in the absence of reported values in the binary-encoded study: a random starting population of 76 individuals, all binary strings with n = 400, is created. Elitism is implemented for the fittest two individuals; the selection method is linear roulette-wheel (parents are selected with probability proportionate to their fitness ranking); selection pressure is seven; and there are six points of crossover, with the crossover type being uniform.

Similarly to the design process for ILS Sampling, the *construction*-related parameters for MS Sampling — as well as those relating to the memetic operations which I added to the GA procedure detailed above — are decided based on initial runs with the aim that the emergent LON be extracted in a reasonable timeframe and that it contains sufficient node and edge density for network analysis.

I augment the GA with local search, rendering it memetic. The local search is best-improvement on a single bit-slip neighbourhood and a solution is deemed locally optimal when it is the incumbent solution after 100 iterations of this. Local search is applied to the fittest 10% of individuals at each generation. Those individuals are added as LON nodes, recombined, and the trajectories from parent to offspring are recorded as LON edges. After 100 generations, pairwise combinations of LON nodes are recursively recombined 10 times: offspring from the first recombination are subject to local search and then become the parents for the next. Nodes and edges are added to the LON during this process.

As was the case with the ILS Sampling framework, I also extract the baseline MS for use in the analysis as a metaheuristic instead of a construction algorithm. The design is altered to facilitate quicker searches: the percentage of individuals subject to local search at each generation becomes 5%, not 10; the best-improvement local search becomes first-improvement; and the local search operator is ten bit-flips instead of one. All parameter settings for the GA component remain the same. One hundred generations are allowed.

8.8 Results

8.8.1 MS Samping LON

The network produced by MS Sampling for this study has 124 497 nodes and 1 264 500 edges, which is an edge-to-node ratio of 10:1. The average fitness is 0.909 698; recall

that the pseudo-optimal fitness is 1.71 in our maximisation environment — it follows that the normalised gap between the two is 0.468. The maximum fitness present in the LON is 1.707 826, which is within 0.001% of the pseudo-optimal fitness. There are 217 separate solutions with this fitness. The lowest objective value in the sample is approximately -107 (remember that a value below zero renders a solution infeasible). The vast majority of local optima included in the sample are feasible solutions.

Because LON edges are directed, we can notice the fitness gradient for them. Around 14.5% of edges are between nodes of equivalent fitness; 43.6% are improving; and 41.8% are deteriorating. The *assortativity* coefficient of a network is the Pearson's correlation for the degrees of connected nodes. For the MS Sampling LON the mean is 0.794 687, implying that it is probable for a node to connect to nodes which have similar degree. The median degree for a node in the LON is 10; the mean is 20.31; the 0.75 quantile is 16; and the maximum is very large at 179 154. Most nodes have relatively low degree (≤ 16) and only 0.1% of nodes have degree $\geq$ 241. The presence of a single node with excessively high degree — 179 154 — hints at the presence of a "hub-and-spoke" network architecture.

8.8.2 ILS Sampling LON

There are 11 393 nodes and 209 489 edges in the ILS sampling LON, for an edge-to-node ratio of approximately 20:1. The average sampled fitness is 0.638 913 — there is a normalised gap of around 0.63 between that and the pseudo-optimal fitness. This is noticeably lower quality than the average fitness in the MS Sampling LON. The maximum fitness in this sample is 1.707 677, which is lower than the maximum in the MS Sampling LON but is still within 0.002% of the pseudo-optimal value. The minimum is approximately -62, which is roughly twice as fit as the lowest in the MS Sampling. This makes sense given the unguided nature of selection inside the MS algorithm when compared to the guided trajectory of the ILS process.

In the LON, around 64% of edges are deteriorating (that is, they orient towards a worse fitness); 26% are improving; and around 9% direct towards equal fitness. These proportions hint at the scarcity of improving moves on the local optima level which are available using these search operations. Recall that the equivalent MS Sampling proportions were 43.6% improving and 41.8% deteriorating. A judicious conclusion might be that the recombination $\longrightarrow$ local search sequence of the MS Sampling algorithm has more *evolvability* potential on the local optima level than does the perturbation $\longrightarrow$ local search sequence of the ILS Sampling algorithm.

The assortativity coefficient is 0.996 704, which stipulates that nodes are highly likely to be connected to nodes which have the same degree as them. This is evidence against the presence of a "hub-and-spoke" network structure in this LON because that phenomenon is defined by heterogeneous degree distribution. The median degree in the LON is 34; the mean is close by at around 37; the 0.75 quantile is 52; and the maximum degree is 526. Only 0.01% of nodes have degree ≥ 128. The range of values in the degree distribution is much less extreme than was present in the MS Sampling LON.

8.8.3 Solution Feasibility in LONs

The existence of infeasible solutions in CSOP fitness landscapes brings new possibilities for the features calculated from LONs. One consideration is whether all of the sampled local optima are feasible. In fact, some are not, although the majority are: 93.6% of nodes have fitness above zero in the MS Sampling product, and this is

86.2% in the ILS Sampling LON. This implies that the former more heavily exploits feasible regions at the level of local optima. Another detail that can be studied for the LONs is the notion of *feasibility gradient*. This is the direction of fitness feasibility that the LON edges encode. For example, an edge could be oriented from an infeasible local optimum towards a feasible local optimum, which is a desirable situation. The distribution of feasibility gradients in the LON therefore captures the ability of the construction algorithm to "escape" infeasible regions.

Table 8.2 provides, for the ILS Sampling network, the feasibility gradients as proportions. View these with the consideration that the algorithm used to construct the edges always accepts improving local optima, but also accepts deteriorating local optima 10% of the time.

TABLE 8.2: Proportions of ILS Sampling LON edges in terms of feasibility gradient

orientation	proportion
infeasible $\longrightarrow$ feasible	$\approx$77%
infeasible $\longrightarrow$ infeasible	$\approx$14%
feasible $\longrightarrow$ feasible	$\approx$5%
feasible $\longrightarrow$ infeasible	$\approx$4%

TABLE 8.3: Proportions of MS Sampling LON edges in terms of feasibility gradient

parameter	proportion
infeasible $\longrightarrow$ feasible	$\approx$7.7%
infeasible $\longrightarrow$ infeasible	$\approx$4.1%
feasible $\longrightarrow$ feasible	$\approx$84.8%
feasible $\longrightarrow$ infeasible	$\approx$3.4%

Encouragingly, the large majority (77%) of edges orient from infeasible to feasible local optima. That implies the operator sequence often succeeds in traversing portals out of infeasible regions. Transformations from feasible $\longrightarrow$ feasible are far fewer at approximately 5% of total edges. This perhaps implies that the operator sequence is not proficient at exploiting within the feasible regions in the search space.

Table 8.3 shows the feasibility gradient percentages seen in the MS Sampling LON. Here a vast majority (84.8%) of the orientations are feasible $\longrightarrow$ feasible. This hints that the operator sequence is competent at intensification within promising areas in the local optima layer of the fitness landscape. The percentage of directions from infeasible $\longrightarrow$ feasible is small, which could also be important — maybe the algorithm struggles to escape infeasible areas. It could be, however, that this small percentage is attributable to the fact that the number of infeasible nodes in the network is low. A surprisingly low percentage of edges (3.4%) lead from feasible $\longrightarrow$ infeasible solutions. This is interesting, because there is no acceptance condition for nodes during the construction. It seems that recombining already-fit solutions before refining the offspring with local search results in fit solutions.

8.8.4 Metaheuristic Performance Comparison

As stipulated in Section 8.7 I use the ILS and MS which form the foundation for the construction algorithms as metaheuristics, alongside the GA from the literature

[210], to conduct optimisation and compare algorithm performance. The configurations have been stated previously. Tables 8.4 and 8.5 summarise distributions for the fitnesses obtained by the metaheuristics over 100 runs. Table 8.4 displays results obtained without a specified computational budget; Table 8.5 shows results from the versions which are budgeted 50 000 fitness evaluations. Each row contains information about an algorithm variant. Indications of the variant are found in the *algorithm* and *seeded* columns. The seed is not chosen due to good fitness (in fact, the fitness is infeasible and heavily-penalised) but rather to provide some consistency of location between algorithms. In the case of GA and MS this is a single individual in the initial populations; for the ILS, it serves as the starting solution. As asserted in Section 8.7.1 I assume the pseudo-optimal fitness value of 1.71 for the purposes of this study.

TABLE 8.4: Distribution of fitnesses obtained by metaheuristics over 100 runs on the CSOP; in the case of the EAs, the best fitness in the population is used as the end fitness; no specific computational budget is ordained for these runs. Large numbers have been rounded to the nearest integer, for visual clarity

algorithm	seeded	0.00 quantile (minimum)	0.25 quantile	0.5 quantile (median)	mean	0.75 quantile	1.00 (maximum)
ILS	no	−25.43	−7.39	−2.09	−4.49	−0.48	**1.66**
GA	no	−71 556	−97.64	−32.15	−3739	−11.67	**0.40**
MS	no	**1.46**	**1.70**	**1.71**	**1.70**	**1.71**	**1.71**
ILS	yes	−62.54	−6.61	−1.87	−4.86	−0.31	**1.67**
GA	yes	−137 601	−212.50	−32.79	−4084	−7.55	−0.19
MS	yes	**1.45**	**1.70**	**1.71**	**1.68**	**1.71**	**1.71**

TABLE 8.5: Distribution of fitnesses obtained by metaheuristics over 100 runs on the CSOP; in the case of the EAs, the best fitness in the population is used as the end fitness; a computational budget of 50 000 fitness function evaluations is ordained for these runs. Large numbers have been rounded to the nearest integer, for visual clarity

algorithm	seeded	0.00 quantile (minimum)	0.25 quantile	0.5 quantile (median)	mean	0.75 quantile	1.00 (maximum)
ILS	no	−55.89	−10.22	−4.26	−7.38	−0.64	**1.66**
GA	no	−235 584	−2538	−75.86	−12 476	−32.63	−4.07
MS	no	**1.15**	**1.46**	**1.66**	**1.59**	**1.68**	**1.70**
ILS	yes	−46.84	−6.25	−1.96	−4.18	**0.15**	**1.44**
GA	yes	−271 039	−4572	−63.20	−10 765	−27.74	**1.19**
MS	yes	**1.31**	**1.45**	**1.67**	**1.60**	**1.68**	**1.70**

Observe that across both budgeted and unbudgeted runs, and also seeded and non-seeded runs, the GA is definitely the least capable of the three in identifying high-quality (or even feasible) solutions. This can be seen by, for example, comparing the median of the GA rows with the median of the MS or ILS rows in either of the two Tables. Sometimes the difference in fitness is several orders of magnitude, and this is especially apparent in the *minimum* columns.

The MS performs by far the best of the three algorithms. In all cases (budgeted and unbudgeted, seeded and non-seeded), 100% of the runs conclude with at least one feasible solution present in the population. That is shown in the third and sixth rows of both Tables. The best ILS runs of the 100 (i.e. the values in the *maximum* rows) end with a desirable fitness although the majority of runs produce a solution with infeasible fitness. It is of note, however, that the distributions comprise fitness values which are "almost" feasible in many cases. I argue that the success (or lack thereof) of ILS on this CSOP — as is the case in many optimisation domains — depends on the starting location of the search. The "almost" feasible fitness values, I stipulate, are the dead-end termination points for sub-optimal funnels of local optima. The consistency of the ILS lies somewhere between the performances of the

GA and the MS: although often the end fitness is infeasible, the range of values in the distributions is tight compared to the GA and usually spans roughly -10 to +1.66. The solution output of an ILS run could be provided as a seed to another highly-exploitative algorithm (such as MS) to complete the optimisation.

The vast range of fitness values obtained by the GA suggests a lack of reliability. Sometimes a feasible solution may be found (a previous paper found that it was approximately 5% of runs after 100 generations [203]) but other times a population filled with infeasible individuals may be produced. Contrarily, the MS appears to be rather uniformly consistent: all of the total 400 runs conclude with a feasible fitness, and this is additionally always ≥ 1.15. The success of the MS tells us that using the sequence of recombination, random mutation, and guided local search together aligns well with this configuration space in directing the search towards promising feasible regions.

8.9 Conclusions

In this Chapter I have modelled a non-benchmark and highly-constrained problem from healthcare with LONs — a Chemotherapy Schedule Optimisation Problem. CSOPs have not been subject to fitness landscape analysis previously. The study brought the added complications of infeasible regions in the fitness landscapes, which is a novel consideration for LON analysis. Two algorithms were offered for the purpose of constructing LONs for CSOP: ILS Sampling and MS Sampling.

Analysis of the LON properties provided ideas for why MS is a superior search algorithm to ILS on this CSOP. The MS-constructed LON contained several global optima, while the ILS-constructed LON contained one, which appeared to be poorly connected to the rest of the network. This difference could help explain why MS reached higher fitnesses with more consistency than ILS.

Examination of the feasibility gradients within the LONs revealed that an ILS framework, for this CSOP and within the confines of the specified experimental environment, may be better at identifying portals out of infeasible regions but may lack in exploitation ability at the feasible local optima level. MS Sampling appears proficient in exploitation within feasible areas but did not carry out many "escapes" from infeasible areas. These findings provide preliminary insight into how the CSOP under study interacts with sequences of search operators. I showed that the MS and ILS which formed the base of the construction algorithms outperform a GA from the literature on this problem instance, even when mandating an equivalent fitness function budget across algorithms. This is particularly apparent in the case of the MS. In light of that result I stipulate that, on this particular CSOP, the selection process of the GA does not have sufficient exploitative potency (perhaps due to the small population size) but that this can be introduced by adding local search. Importantly, the best results are obtained when using the recombination and random mutation of the GA *together with* a guided local search in a memetic framework. It follows that — at least under this experimental setting — the former brings innovation and diversification, while the latter ensures intensification and facilitates propagation of high-quality genes.

I conclude this Chapter by stipulating that the results within it should be viewed as an observational study only: a single instance of CSOP was investigated. It follows that additional or alternative instances, different parameter configurations, or different features of the LONs may result in separate findings. Nevertheless, this

work serves as an initial foundation for this kind of analysis on CSOPs and on constrained problems, and I leave the inclusion of additional instances, algorithms, and parameter choices for future work.

Chapter 9

Discussion

9.1 Contributions

In this section, the particular contributions of the thesis are discussed and their context with regard to current research is considered.

9.1.1 Fractal Analysis of Local Optima Networks

In Chapter 4, fractal geometry, a new perspective on the study of local optima networks was introduced. Previous literature concerning fractal analysis in fitness landscapes has focussed mostly on the solution level [29, 145] instead of the local optima level, which was the subject of attention throughout this thesis. One paper did concentrate on local optima in particular [121] although they did not consider fractal dimensions but instead conducted fitness-distance analysis and used elemental operators to define distance between solutions — factors which distinguish their contributions from those presented in this thesis.

Chapter 4 proposed a method for calculating fractal dimension which is specific to the semantics of a LON. Novel insight into the information encoded within LONs was gained from the fractal analysis. In particular, it was found in regression analysis that fractal dimension features combined with the number of local optima and the number of funnels as predictors produce extremely strong models for metaheuristic performance explanation and prediction. In many cases, fractal dimension predictors ranked higher in variable importance rankings than the number of funnels did. Fractal dimension which was computed with the LON-specific algorithm design were generally more effective as predictors than those produced using ordinary box counting for complex networks. The number of funnels usually ranked as the most important variable for explaining algorithm hit-rate. Although funnels were not the focal point of the study, this is interesting because funnels are basins of attraction at the local optima level and are — in this respect — a fractal pattern or structure in the landscape.

9.1.2 Approaches to Computing Fractal Dimension

In Chapter 5, a new approach for calculating fractal dimension for LONs: multifractal analysis was proposed and implemented. In addition, *probabilistic* fractal analysis was presented. This used the metaheuristic transition probabilities encoded in the network edge weights for dimension calculation. The analysis involved sampled LONs as well as exhaustively enumerated ones. While the preceding work in Chapter 4 was the first fractal analysis on LONs, Chapter 5 contributed the inclusion of sampled LONs alongside the novel procedures for computing dimension.

Other complex networks have previously been identified as multifractal systems [180, 186, 182] and Chapter 5 confirmed that LONs associated with certain instance classes of QAP are. Pairwise correlations were revealed between the dimensions and metaheuristic algorithm performance. Probabilistic fractal dimensions, which consider the likelihood of connection between local optima, were correlated with slower runtimes. These new dimensions were more strongly correlated to search than dimensions calculated using the previously-proposed mechanisms of the previous Chapter.

9.1.3 Predictive Potency of Fractal Dimensions

The experiments in Chapter 6 concluded the sequence of fractal analysis work with a detailed investigation into the predictive capabilities of LON fractal dimensions. This endeavour involved a significant expansion of the number of included instances as well as the inclusion of larger problem sizes. The originality of the work and its contributions lies in this expansion, as well as enhanced statistical techniques for the predictive modelling and the use of more intuitive fractal dimension features.

It was shown that the *extent* of multifractality in the system, as well as the size of fractal dimensions could contribute as predictors in models with the performance of iterated local search or tabu search as the response variable. Features based on the local optima level fitness distribution were dominant in the variable importances for the models which concerned ROTS as response variable.

9.1.4 Construction Algorithm Appraisal

The majority of works concerning local optima networks conduct exhaustive enumeration of them, which bounds the analysis to problems diminutive in size [11, 4, 21, 8]. More recently, sampling algorithms have been proposed for constructing LONs for larger problem sizes [19, 24, 34]; however, little is known about them and about the quality of the sample they produce.

The primary contribution in Chapter 7 is the comparison and analysis of LON construction algorithms; this involved not only a descriptive appraisal but also a contest for the relative predictive power of LONs produced by the algorithms. It was found that Snowball Sampling was tuneable and predictable; ILS Sampling produced highly-variable LONs and seemed to capture potential metaheuristic flow more accurately.

The modelling in Chapter 7 demonstrated that Snowball Sampling and ILS Sampling each infer metaheuristic performance very well (at least for this dataset and under these experimental conditions).

A salient endeavour presented in this chapter involved the comparison of sampled LONs against their exhaustively-enumerated counterparts. This has not been analysed before, although a previous paper contrasted the attributes of sampled networks against complete ones [25]. We established from this competition that sampled LONs are superior to exhaustively-enumerated ones in this respect.

In addition, the construction algorithms were supplied a fixed budget of 50 000 fitness function evaluations in pursuit of a comparison which is closer to being equitable. While sampling parameters have been varied before [25] the algorithms have not been subject to a budget which is agnostic of the particular construction process in prior literature. Under the budget, Snowball Sampling LONs were more proficient at explaining tabu search performance variance, while ILS Sampling LONs were better-suited to iterated local search as a response variable. Another venture

from the Chapter concerned combining together features of LONs produced using different construction algorithms into predictive regressions for the first time which resulted in highly-promising models.

Two overarching trends which became evident from the analysis in Chapter 7 were that — firstly — random forest regression appears to better capture non-linearities in LON performance prediction data and, secondly, that features relating to the fitness distribution in LON samples play an important role in metaheuristic performance explanation and prediction.

9.1.5 LONs for a Constrained Problem

In Chapter 8 we modelled a problem from healthcare with LONs. Most LON works have used benchmark combinatorial problems, such as NK Landscapes [11], Quadratic Assignment [5], and the Travelling Salesman Problem [19]. This Chapter pursued the consideration of non-benchmark, constrained, and more specialised problems. A particular originality arose from this: the study of a fitness landscape with infeasible regions using LON analysis (see Chapter 3 for a review of LON literature to date).

In Chapter 8 the interactions between a Chemotherapy Schedule Optimisation Problem (CSOP) and metaheuristic search operations were considered by constructing LONs with an iterated local search-driven algorithm and a memetic search-driven algorithm. Through appraisal of the LON products, it was found that iterated local search — for this particular instance and under the chosen experimental environment — is competent at exiting infeasible regions in fitness landscapes associated with this problem; memetic search is, contrarily, proficient at exploiting feasible regions more comprehensively. Memetic search produced the highest-quality solutions on a consistent basis and it is therefore stipulated that the CSOP responds best to the combination of crossover with random mutation and local search. Iterated local search rarely reached the highest known fitness, although the obtained solution quality was often close. Both algorithms performed better than a Genetic Algorithm from the literature on the CSOP under study.

9.2 Evaluation of Hypothesis

Hypothesis. *Valuable information is encoded within Local Optima Networks about reactions between metaheuristic algorithms and combinatorial optimisation problems; this can be used for visualising, explaining, or predicting the proficiency of those algorithms.*

Throughout this thesis, Local Optima Network features are used to gain insight about optimisation, and have fulfilled prominent roles in explaining or predicting metaheuristic algorithm performance. Multiple linear regression models, mixed-effects models, random forest regression, correlation analysis, and visual analysis are among the instruments which facilitate these results. The universal thread binding together each contribution is that the LON features which have been considered are associated with metaheuristic performance.

Findings are presented across a range of problem dimensions (from the $N = 11$ QAP instances of Chapter 3 to $N = 128$ in Chapter 7, and additionally across both benchmark (Chapters 4, 5, 6, and 7) and non-benchmark instances (Chapters 5 and 8). As well as belonging to separate problems domains, a wealth of different instance classes are involved.

In light of the broad application of the analysis, the collection of statistical tools, the variety of different LON features included, and the overarching trend that LON features can capture potent predictive information about metaheuristic search, it is proposed that the contributions of this thesis have provided a significant body of evidence to support the stipulated hypothesis.

9.3 Future Work

9.3.1 Adaptive Optimisation

The most intuitive and important future work lies in the use of LON analysis during optimisation. That is, it should be considered for the implementation of online local optima network estimation, to facilitate adaptive choices of operators or parameters. Some ground work has been established with this thesis; the knowledge that LON features are linked to metaheuristic algorithm proficiency when the relationship is examined "post-mortem" emphasises the potential in analysing them during the optimisation process.

The reason that the proposed future work is important is that, at the moment, construction of a LON is computationally-expensive and the offline analysis of the object can be time-consuming. Although it is argued that this approach was productive for the purpose of the thesis contributions — it allowed thorough and careful study of the objects — with any fitness landscape analysis, the benefit of using it instead of going straight to optimisation with a metaheuristic should be evident. This is particularly true for non-benchmark problems where fitness function evaluations may be slow and real-world partners may be under time constraints.

Integrating the LON method as a decision mechanism into a metaheuristic algorithm could generate an attractive tool for industry partners and should be less computationally-expensive overall if done mindfully. The procedure should consider only a "rolling" partial local optima network, or possibly retain summary statistics of the historical local optima connectivity within memory. Such features could be used on-the-fly to adaptively design the algorithm in light of the perceived local optima topology present within the particular instance at hand.

9.3.2 Cost of Construction

As has been mentioned, an evident disadvantage to local optima network analysis is the computational cost. While it is argued that the insights gained from such analysis have thus far outweighed the price, the future appeal and usage of LONs would benefit greatly from a thorough appraisal of their efficiency and budget. Chapter 7 carried out comparisons of LON construction algorithms. Although progress was made, a priority in the future should be reducing the computational cost of construction while still producing a representative LON sample.

Deciding upon a suitable sampling effort will probably not be a trivial task. In the most optimistic scenario for rendering generic sampling algorithms but also adapted to the specific problem, the amount of computation might be tuned according to the problem dimension. The reality of this endeavour will be highly nuanced, in that the amount of computation required may well be domain-specific, if not instance-specific (as is arguably the case, in essence, for algorithm selection [164]) and that problem dimension may be too simplistic to use alone as an indicator for appropriate sampling effort.

9.3.3 Gradients for Fractal Analysis

The analysis in Chapters 4, 5 and 6 examined the local optima level for any emergent fractal geometry and complexity. A promising direction in the future would be using this work as the foundation for comparing the observed geometry of this level against that of the baseline solution layer in the landscape. The two can be viewed as different levels of resolution for patterns in the fitness landscape; using fractal analysis terminology, different scaling factors or lengths of measurement. It follows that examining the geometries for similarity could provide evidence for or against patterns occurring at different scales.

In principle, the solution level could be analysed for the number and steepness of gradient changes using the information-theoretic metrics [93] mentioned in Section 3.1.1. The same technique could then be used on a random (or even adaptive) walk on the local optima level. Features of the walks could be contrasted and investigated for the presence of similar spatial patterns.

9.3.4 Multifractal LON System Tracking

In Chapter 5 the results showed that LONs of some combinatorial optimisation problems are *multi-fractal* systems. This means that the scaling or geometric complexity is significantly different in parts. An interesting advancement in light of this result would be identifying the sub-systems in the LON and meticulously tracking live metaheuristic searches as they move through those same areas. The tracking would facilitate comparison of how searches progress through the different areas; it may be that where one region is particularly spatially complex (that is, that area of the local optima level has high *fractal dimension* relative to other LON sections) a different search operator would be better suited. In this way, involving the multifractality of a LON in the pursuit of understanding metaheuristic performance and in selecting operators could be productive.

9.3.5 Expansion of CSOP LON Analysis

I mentioned at the conclusion of Chapter 8 that the results within that Chapter should be viewed as observational at this point and cannot be extrapolated to other instances. Future work to this end would involve the inclusion of additional CSOP instances (possibly the original CSOP instance with edited instance parameters); statistical analysis between features of the LONs and metaheuristic performance on the instances; and alternative designs for the LON construction algorithms and features extracted. In this way, the preliminary foundation which has been established in Chapter 8 could be properly validated and fortified, and further insight into the interplay between CSOPs and metaheuristic algorithms could be gained.

9.4 Concluding Remarks

This thesis has examined local optima networks in pursuit of knowledge about reactions between combinatorial optimisation problems and metaheuristic search algorithms. Features were proposed, computed, compared, and visualised. Predictive models were built with the features and used to explain or predict search algorithm proficiency across multiple domains, problem sizes, and instance classes. It is hoped that the insight born from the experiments and contributions can inspire further

innovation and understanding of how to conduct more mindful optimisation with awareness of fitness landscape topology.

Bibliography

[1] Peter F Stadler. "Fitness Landscapes". In: *Biological evolution and statistical physics*. Springer, 2002, pp. 183–204.

[2] Melanie Mitchell, Stephanie Forrest, and John H Holland. "The royal road for genetic algorithms: Fitness landscapes and GA performance". In: *Proceedings of the first european conference on artificial life*. 1992, pp. 245–254.

[3] Holger H Hoos, Kevin Smyth, and Thomas Stützle. "Search space features underlying the performance of stochastic local search algorithms for MAX-SAT". In: *PPSN*. Springer. 2004, pp. 51–60.

[4] Sébastien Verel, Fabio Daolio, Gabriela Ochoa, and Marco Tomassini. "Local Optima Networks with Escape Edges". In: *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer. 2011, pp. 49–60.

[5] Fabio Daolio, Marco Tomassini, Sébastien Vérel, and Gabriela Ochoa. "Communities of Minima in Local Optima Networks of Combinatorial Spaces". In: *Physica A: Statistical Mechanics and its Applications* 390.9 (2011), pp. 1684–1694.

[6] Katherine M Malan and Andries P Engelbrecht. "Ruggedness, funnels and gradients in fitness landscapes and the effect on PSO performance". In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE. 2013, pp. 963–970.

[7] David Iclanzan, Fabio Daolio, and Marco Tomassini. "Data-driven local optima network characterization of QAPLIB instances". In: *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM. 2014, pp. 453–460.

[8] Sebastian Herrmann, Gabriela Ochoa, and Franz Rothlauf. "Communities of Local Optima as Funnels in Fitness Landscapes". In: *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*. ACM. 2016, pp. 325–331.

[9] Pascal Kerschke, Hao Wang, Mike Preuss, Christian Grimme, André Deutz, Heike Trautmann, and Michael Emmerich. "Towards analyzing multimodality of continuous multiobjective landscapes". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2016, pp. 962–972.

[10] Jean-Paul Watson. "An introduction to fitness landscape analysis and cost models for local search". In: *Handbook of metaheuristics*. Springer, 2010, pp. 599–623.

[11] Gabriela Ochoa, Marco Tomassini, Sebástien Vérel, and Christian Darabos. "A Study of NK Landscapes' Basins and Local Optima Networks". In: *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. ACM. 2008, pp. 555–562.

[12] Leticia Hernando, Fabio Daolio, Nadarajen Veerapen, and Gabriela Ochoa. "Local Optima Networks of the Permutation Flowshop Scheduling Problem: Makespan vs. Total Flow Time". In: *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2017, pp. 1964–1971.

[13] Wojciech Bożejko, Andrzej Gnatowski, Teodor Niżyński, Michael Affenzeller, and Andreas Beham. "Local Optima Networks in Solving Algorithm Selection Problem for TSP". In: *International Conference on Dependability and Complex Systems*. Springer. 2018, pp. 83–93.

[14] Werner Mostert, Katherine M Malan, Gabriela Ochoa, and Andries P Engelbrecht. "Insights into the Feature Selection Problem Using Local Optima Networks". In: *European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar)*. Springer. 2019, pp. 147–162.

[15] Marco Baioletti, Alfredo Milani, Valentino Santucci, and Marco Tomassini. "Search moves in the local optima networks of permutation spaces: the QAP case". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM. 2019, pp. 1535–1542.

[16] Marcella Scoczynski Ribeiro Martins, Mohamed El Yafrani, Myriam RBS Delgado, and Ricardo Luders. "Multi-layer local optima networks for the analysis of advanced local search-based algorithms". In: *arXiv preprint arXiv:2004.13936* (2020).

[17] Fabio Daolio, Sébastien Verel, Gabriela Ochoa, and Marco Tomassini. "Local optima networks and the performance of iterated local search". In: *Proceedings of the 14th annual conference on Genetic and evolutionary computation*. ACM. 2012, pp. 369–376.

[18] Sebastian Herrmann and Franz Rothlauf. "Predicting heuristic search performance with pageRank centrality in local optima networks". In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM. 2015, pp. 401–408.

[19] Gabriela Ochoa and Nadarajen Veerapen. "Deconstructing the Big Valley Search Space Hypothesis". In: *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer. 2016, pp. 58–73.

[20] Piotr Lipinski and Krzysztof Michalak. "Deriving knowledge from local optima networks for evolutionary optimization in inventory routing problem". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM. 2019, pp. 1551–1558.

[21] Francisco Chicano, Fabio Daolio, Gabriela Ochoa, Sébastien Vérel, Marco Tomassini, and Enrique Alba. "Local Optima Networks, Landscape Autocorrelation and Heuristic Search Performance". In: *Parallel Problem Solving from Nature-PPSN XII* (2012), pp. 337–347.

[22] Jing Liu, Hussein A Abbass, and Kay Chen Tan. "Problem Difficulty Analysis Based on Complex Networks". In: *Evolutionary Computation and Complex Networks*. Springer, 2019, pp. 39–52.

[23] Nadarajen Veerapen, Gabriela Ochoa, Renato Tinós, and Darrell Whitley. "Tunnelling Crossover Networks for the Asymmetric TSP". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2016, pp. 994–1003.

[24] Gabriela Ochoa and Sebastian Herrmann. "Perturbation Strength and the Global Structure of QAP Fitness Landscapes". In: *15th International Conference, Coimbra, Portugal, September 8–12, 2018, Proceedings, Part II*. Jan. 2018, pp. 245–256. ISBN: 978-3-319-99258-7.

[25] Sébastien Verel, Fabio Daolio, Gabriela Ochoa, and Marco Tomassini. "Sampling Local Optima Networks of Large Combinatorial Search Spaces: The QAP Case". In: *Parallel Problem Solving from Nature – PPSN XV*. Ed. by Anne Auger, Carlos M. Fonseca, Nuno Lourenço, Penousal Machado, Luís Paquete, and Darrell Whitley. Springer International Publishing, 2018, pp. 257–268.

[26] Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

[27] Kalyanmoy Deb. "Multi-objective optimisation using evolutionary algorithms: an introduction". In: *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer, 2011, pp. 3–34.

[28] Stuart Kauffman and Simon Levin. "Towards a general theory of adaptive walks on rugged landscapes". In: *Journal of theoretical Biology* 128.1 (1987), pp. 11–45.

[29] Edward Weinberger. "Correlated and uncorrelated fitness landscapes and how to tell the difference". In: *Biological cybernetics* 63.5 (1990), pp. 325–336.

[30] Peter Merz. "Advanced fitness landscape analysis and the performance of memetic algorithms". In: *Evolutionary Computation* 12.3 (2004), pp. 303–325.

[31] Eugene L Lawler. "The quadratic assignment problem". In: *Management science* 9.4 (1963), pp. 586–599.

[32] Peter Merz and Bernd Freisleben. "Fitness landscape analysis and memetic algorithms for the quadratic assignment problem". In: *IEEE transactions on evolutionary computation* 4.4 (2000), pp. 337–352.

[33] Erik Pitzer and Michael Affenzeller. "A Comprehensive Survey on Fitness Landscape Analysis". In: *Recent Advances in Intelligent Engineering Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 161–191.

[34] Sébastien Verel, Fabio Daolio, Gabriela Ochoa, and Marco Tomassini. "Sampling local optima networks of large combinatorial search spaces: The qap case". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2018, pp. 257–268.

[35] Michel Gendreau and Jean-Yves Potvin. *Handbook of metaheuristics*. Vol. 2. Springer, 2010.

[36] Thomas Bäck, David B Fogel, and Zbigniew Michalewicz. *Handbook of evolutionary computation*. CRC Press, 1997.

[37] Sewall Wright. *The roles of mutation, inbreeding, crossbreeding, and selection in evolution*. Vol. 1. na, 1932.

[38] Bernard Derrida and Luca Peliti. "Evolution in a flat fitness landscape". In: *Bulletin of mathematical biology* 53.3 (1991), pp. 355–382.

[39] L Hadany and T Beker. "Fitness-associated recombination on rugged adaptive landscapes". In: *Journal of evolutionary biology* 16.5 (2003), pp. 862–870.

[40] Frank J Poelwijk, Daniel J Kiviet, Daniel M Weinreich, and Sander J Tans. "Empirical fitness landscapes reveal accessible evolutionary paths". In: *Nature* 445.7126 (2007), p. 383.

[41] Neil T Miller and Robert L Burnap. "Navigating the fitness landscape using multiallele genome editing". In: *Proceedings of the National Academy of Sciences* 115.50 (2018), pp. 12547–12549.

[42] Ariel Fernández and Eugene I Shakhnovich. "Activation-energy landscape for metastable RNA folding". In: *Physical Review A* 42.6 (1990), p. 3657.

[43] Hans Frauenfelder, Stephen G Sligar, and Peter G Wolynes. "The energy landscapes and motions of proteins". In: *Science* 254.5038 (1991), pp. 1598–1603.

[44] Olivier Martin, Steve W Otto, and Edward W Felten. "Large-step Markov chains for the TSP incorporating local search heuristics". In: *Operations Research Letters* 11.4 (1992), pp. 219–224.

[45] James E Komianos and Garegin A Papoian. "Stochastic Ratcheting on a Funneled Energy Landscape is Necessary for Highly Efficient Contractility of Actomyosin Force Dipoles". In: *Physical Review X* 8.2 (2018), p. 021006.

[46] Jozef Adamcik and Raffaele Mezzenga. "Amyloid polymorphism in the protein folding and aggregation energy landscape". In: *Angewandte Chemie International Edition* 57.28 (2018), pp. 8370–8382.

[47] Terry Jones. "Evolutionary algorithms, fitness landscapes and search". PhD thesis. Citeseer, 1995.

[48] Mark EJ Newman. "The structure and function of complex networks". In: *SIAM review* 45.2 (2003), pp. 167–256.

[49] Leticia Hernando, Alexander Mendiburu, and Jose A Lozano. "Anatomy of the attraction basins: Breaking with the intuition". In: *Evolutionary computation* 27.3 (2019), pp. 435–466.

[50] Sébastien Verel, Gabriela Ochoa, and Marco Tomassini. "The connectivity of NK landscapes' basins: A network analysis". In: *arXiv preprint arXiv:0810.3492* (2008).

[51] David Iclănzan, Fabio Daolio, and Marco Tomassini. "Learning inherent networks from stochastic search methods". In: *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer. 2014, pp. 157–169.

[52] Fabio Daolio, Sébastien Verel, Gabriela Ochoa, and Marco Tomassini. "Local optima networks of the quadratic assignment problem". In: *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE. 2010, pp. 1–8.

[53] Gabriela Ochoa, Sébastien Verel, Fabio Daolio, and Marco Tomassini. "Clustering of Local Optima in Combinatorial Fitness landscapes". In: *International Conference on Learning and Intelligent Optimization*. Springer. 2011, pp. 454–457.

[54] Fabio Daolio, Sébastien Verel, Gabriela Ochoa, and Marco Tomassini. "Local optima networks of the permutation flow-shop problem". In: *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer. 2013, pp. 41–52.

[55] Gabriela Ochoa, Francisco Chicano, Renato Tinós, and Darrell Whitley. "Tunnelling Crossover Networks". In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM. 2015, pp. 449–456.

[56] Gabriela Ochoa, Nadarajen Veerapen, Fabio Daolio, and Marco Tomassini. "Understanding Phase Transitions with Local Optima Networks: Number Partitioning as a Case Study". In: *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer. 2017, pp. 233–248.

[57] William B Langdon, Nadarajen Veerapen, and Gabriela Ochoa. "Visualising the search landscape of the triangle program". In: *European Conference on Genetic Programming*. Springer. 2017, pp. 96–113.

[58] Nadarajen Veerapen, Fabio Daolio, and Gabriela Ochoa. "Modelling genetic improvement landscapes with local optima networks". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM. 2017, pp. 1543–1548.

[59] David Simoncini, Sophie Barbe, Thomas Schiex, and Sébastien Verel. "Fitness Landscape Analysis Around the Optimum in Computational Protein Design". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM. 2018, pp. 355–362.

[60] Arnaud Liefooghe, Bilel Derbel, Sébastien Verel, Manuel López-Ibáñez, Hernan Aguirre, and Kiyoshi Tanaka. "On Pareto local optimal solutions networks". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2018, pp. 232–244.

[61] Jonathan E Fieldsend and Khulood Alyahya. "Visualising the landscape of multi-objective problems using local optima networks". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM. 2019, pp. 1421–1429.

[62] Tamás Vinkó and Kitti Gelle. "Basin Hopping Networks of continuous global optimization problems". In: *Central European Journal of Operations Research* 25.4 (2017), pp. 985–1006.

[63] Jason Adair, Gabriela Ochoa, and Katherine M Malan. "Local optima networks for continuous fitness landscapes". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM. 2019, pp. 1407–1414.

[64] Viktor Homolya and Tamás Vinkó. "Memetic differential evolution using network centrality measures". In: *AIP Conference Proceedings*. Vol. 2070. 1. AIP Publishing. 2019, p. 020023.

[65] Gabriela Ochoa and Francisco Chicano. "Local optima network analysis for MAX-SAT". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM. 2019, pp. 1430–1437.

[66] German Treimun-Costa, Elizabeth Montero, Gabriela Ochoa, and Nicolás Rojas-Morales. "Modelling parameter configuration spaces with local optima networks". In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 2020, pp. 751–759.

[67] Gabriela Ochoa, Sébastien Verel, and Marco Tomassini. "First-improvement vs. Best-improvement Local Optima Networks of NK Landscapes". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2010, pp. 104–113.

[68] Benoit B Mandelbrot. "Stochastic models for the Earth's relief, the shape and the fractal dimension of the coastlines, and the number-area rule for islands". In: *Proceedings of the National Academy of Sciences* 72.10 (1975), pp. 3825–3828.

[69] LJ Hadjileontiadis and E Douka. "Crack detection in plates using fractal dimension". In: *Engineering Structures* 29.7 (2007), pp. 1612–1625.

[70] Marcel Dicke and PETER A BURROUGH. "Using fractal dimensions for characterizing tortuosity of animal trails". In: *Physiological Entomology* 13.4 (1988), pp. 393–398.

[71] Curtis B Caldwell, Sandra J Stapleton, David W Holdsworth, Roberta A Jong, William J Weiser, Gabriel Cooke, and Martin J Yaffe. "Characterisation of mammographic parenchymal pattern by fractal dimension". In: *Physics in medicine & biology* 35.2 (1990), p. 235.

[72] Abdelrahim Nasser Esgiar, Raouf NG Naguib, Bayan S Sharif, Mark K Bennett, and Alan Murray. "Fractal analysis in the detection of colonic cancer images". In: *IEEE transactions on information technology in biomedicine* 6.1 (2002), pp. 54–58.

[73] Benoit B Mandelbrot. *The fractal geometry of nature*. Vol. 173. WH freeman New York, 1983.

[74] David Harte. *Multifractals: theory and applications*. CRC Press, 2001.

[75] David A Freedman. *Statistical models: theory and practice*. cambridge university press, 2009.

[76] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282.

[77] John Fox and Sanford Weisberg. *An R companion to applied regression*. Sage publications, 2018.

[78] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[79] Richard Taylor. "Interpretation of the correlation coefficient: a basic review". In: *Journal of diagnostic medical sonography* 6.1 (1990), pp. 35–39.

[80] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. "Pearson correlation coefficient". In: *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.

[81] Nian Shong Chok. "Pearson's versus Spearman's and Kendall's correlation coefficients for continuous data". PhD thesis. University of Pittsburgh, 2010.

[82] Joost CF de Winter, Samuel D Gosling, and Jeff Potter. "Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data." In: *Psychological methods* 21.3 (2016), p. 273.

[83] Charles Spearman. "The proof and measurement of association between two things". In: *The American journal of psychology* 100.3/4 (1987), pp. 441–471.

[84] Daniel J Ozer. "Correlation and the coefficient of determination." In: *Psychological bulletin* 97.2 (1985), p. 307.

[85] Ronald Aylmer Fisher. "Statistical methods for research workers". In: *Breakthroughs in statistics*. Springer, 1992, pp. 66–70.

[86] Giovanni Di Leo and Francesco Sardanelli. "Statistical significance: p value, 0.05 threshold, and applications to radiomics—reasons for a conservative approach". In: *European Radiology Experimental* 4.1 (2020), pp. 1–8.

[87] Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

[88] Rob J Hyndman and Anne B Koehler. "Another look at measures of forecast accuracy". In: *International journal of forecasting* 22.4 (2006), pp. 679–688.

[89] David A Freedman. "Bootstrapping regression models". In: *The Annals of Statistics* 9.6 (1981), pp. 1218–1228.

[90] Wim Hordijk and Peter F Stadler. "Amplitude spectra of fitness landscapes". In: *Advances in Complex Systems* 1.01 (1998), pp. 39–66.

[91] Hui Lu, Jinhua Shi, Zongming Fei, Qianlin Zhou, and Kefei Mao. "Measures in the time and frequency domains for fitness landscape analysis of dynamic optimization problems". In: *Applied Soft Computing* 51 (2017), pp. 192–208.

[92] Hui Lu, Rongrong Zhou, Zongming Fei, and Chongchong Guan. "Spatial-domain fitness landscape analysis for combinatorial optimization". In: *Information Sciences* 472 (2019), pp. 126–144.

[93] Vesselin K Vassilev, Terence C Fogarty, and Julian F Miller. "Information characteristics and the structure of landscapes". In: *Evolutionary computation* 8.1 (2000), pp. 31–60.

[94] Yossi Borenstein and Riccardo Poli. "Information landscapes". In: *GECCO*. Vol. 5. Citeseer. 2005, pp. 1515–1522.

[95] Katherine M Malan and Andries P Engelbrecht. "A survey of techniques for characterising fitness landscapes and some possible ways forward". In: *Information Sciences* 241 (2013), pp. 148–163.

[96] Erik Pitzer, Michael Affenzeller, Andreas Beham, and Stefan Wagner. "Comprehensive and automatic fitness landscape analysis using heuristiclab". In: *Computer Aided Systems Theory–EUROCAST 2011* (2012), pp. 424–431.

[97] Erik Pitzer, Michael Affenzeller, and Andreas Beham. "A closer look down the basins of attraction". In: *2010 UK Workshop on Computational Intelligence (UKCI)*. IEEE. 2010, pp. 1–6.

[98] Leticia Hernando, Alexander Mendiburu, and Jose A Lozano. "Estimating Attraction Basin Sizes". In: *Conference of the Spanish Association for Artificial Intelligence*. Springer. 2016, pp. 458–467.

[99] J Garnier and L Kallel. "How to detect all maxima of a function". In: *Theoretical aspects of evolutionary computing*. Springer, 2001, pp. 343–370.

[100] Colin R Reeves and Anton V Eremeev. "Statistical analysis of local search landscapes". In: *Journal of the Operational Research Society* 55.7 (2004), pp. 687–693.

[101] Khulood Alyahya and Jonathan E Rowe. "Simple random sampling estimation of the number of local optima". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2016, pp. 932–941.

[102] Marie-Eléonore Marmion, Clarisse Dhaenens, Laetitia Jourdan, Arnaud Liefooghe, and Sébastien Verel. "NILS: a neutrality-based iterated local search and its application to flowshop scheduling". In: *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer. 2011, pp. 191–202.

[103] Christian M Reidys and Peter F Stadler. "Neutrality in fitness landscapes". In: *Applied Mathematics and Computation* 117.2 (2001), pp. 321–350.

[104] Lionel Barnett. "Netcrawling-optimal evolutionary search with neutral networks". In: *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*. Vol. 1. IEEE. 2001, pp. 30–37.

[105] Tina Yu and Julian Miller. "Finding needles in haystacks is not hard with neutrality". In: *European Conference on Genetic Programming*. Springer. 2002, pp. 13–25.

[106] Leonardo Vanneschi, Marco Tomassini, Philippe Collard, Sébastien Vérel, Yuri Pirola, and Giancarlo Mauri. "A comprehensive view of fitness landscapes with neutrality and fitness clouds". In: *European Conference on Genetic Programming*. Springer. 2007, pp. 241–250.

[107] Sébastien Verel, Philippe Collard, and Manuel Clergue. "Where are bottlenecks in nk fitness landscapes?" In: *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*. Vol. 1. IEEE. 2003, pp. 273–280.

[108] Tom Smith, Phil Husbands, and Michael O'Shea. "Fitness landscapes and evolvability". In: *Evolutionary computation* 10.1 (2002), pp. 1–34.

[109] Leonardo Vanneschi, Marco Tomassini, Philippe Collard, and Sébastien Vérel. "Negative slope coefficient: A measure to characterize genetic programming fitness landscapes". In: *European Conference on Genetic Programming*. Springer. 2006, pp. 178–189.

[110] Guanzhou Lu, Jinlong Li, and Xin Yao. "Fitness-probability cloud and a measure of problem hardness for evolutionary algorithms". In: *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer. 2011, pp. 108–117.

[111] Terry Jones and Stephanie Forrest. "Fitness distance correlation as a measure of problem didculty for genetic algorithms". In: *Proc. 6th Internat. Conf. on Genetic Algorithms*. 1995.

[112] Kenneth Dean Boese. *Cost versus distance in the traveling salesman problem*. UCLA Computer Science Department Los Angeles, 1995.

[113] Peter Merz and Bernd Freisleben. "Fitness landscapes and memetic algorithm design". In: *New ideas in optimization* (1999), pp. 245–260.

[114] Colin R Reeves. "Landscapes, operators and heuristic search". In: *Annals of Operations Research* 86 (1999), pp. 473–490.

[115] Marco Tomassini, Leonardo Vanneschi, Philippe Collard, and Manuel Clergue. "A study of fitness distance correlation as a difficulty measure in genetic programming". In: *Evolutionary computation* 13.2 (2005), pp. 213–239.

[116] Marek Kubiak. "Distance measures and fitness-distance analysis for the capacitated vehicle routing problem". In: *Metaheuristics*. Springer, 2007, pp. 345–364.

[117] Gabriela Ochoa, Rong Qu, and Edmund K Burke. "Analyzing the landscape of a graph based hyper-heuristic for timetabling problems". In: *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. ACM. 2009, pp. 341–348.

[118] Fuqing Zhao, Feilong Xue, Guoqiang Yang, Weimin Ma, Chuck Zhang, and Houbin Song. "A Fitness Landscape Analysis for the No-Wait Flow Shop Scheduling Problem With Factorial Representation". In: *IEEE Access* 7 (2019), pp. 21032–21047.

[119] Doug R Hains, L Darrell Whitley, and Adele E Howe. "Revisiting the Big Valley Search Space Structure in the TSP". In: *Journal of the Operational Research Society* 62.2 (2011), pp. 305–312.

[120] Lee Altenberg. "Fitness Distance Correlation Analysis: An Instructive Counterexample." In: *ICGA*. Citeseer. 1997, pp. 57–64.

[121] Yong-Hyuk Kim and Byung-Ro Moon. "Investigation of the fitness landscapes in graph bipartitioning: An empirical study". In: *Journal of Heuristics* 10.2 (2004), pp. 111–133.

[122] Marco Locatelli. "On the multilevel structure of global optimization problems". In: *Computational Optimization and Applications* 30.1 (2005), pp. 5–22.

[123] Paul McMenemy, Nadarajen Veerapen, and Gabriela Ochoa. "How Perturbation Strength Shapes the Global Structure of TSP Fitness Landscapes". In: *European Conference on Evolutionary Computation in Combinatorial Optimization*. Jan. 2018, pp. 34–49. ISBN: 978-3-319-77448-0.

[124] Sébastien Verel, Gabriela Ochoa, and Marco Tomassini. "Local optima networks of NK landscapes with neutrality". In: *IEEE Transactions on Evolutionary Computation* 15.6 (2010), pp. 783–797.

[125] Sebastian Herrmann and Franz Rothlauf. "Predicting Heuristic Search Performance with PageRank Centrality in Local Optima Networks". In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. GECCO '15. Madrid, Spain: ACM, 2015, pp. 401–408. ISBN: 978-1-4503-3472-3. DOI: 10.1145/2739480.2754691. URL: http://doi.acm.org/10.1145/2739480.2754691.

[126] Sebastian Herrmann, Gabriela Ochoa, and Franz Rothlauf. "PageRank centrality for performance prediction: the impact of the local optima network model". In: *Journal of Heuristics* 24.3 (2018), pp. 243–264.

[127] Jonathan E Fieldsend. "Computationally Efficient Local Optima Network Construction". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM. 2018, pp. 1481–1488.

[128] Shen Lin and Brian W Kernighan. "An effective heuristic algorithm for the traveling-salesman problem". In: *Operations research* 21.2 (1973), pp. 498–516.

[129] Francisco Chicano, Darrell Whitley, Gabriela Ochoa, and Renato Tinós. "Optimizing one million variable NK landscapes by hybridizing deterministic recombination and local search". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM. 2017, pp. 753–760.

[130] Jonathan PK Doye, Mark A Miller, and David J Wales. "The Double-funnel Energy Landscape of the 38-atom Lennard-Jones Cluster". In: *The Journal of Chemical Physics* 110.14 (1999), pp. 6896–6906.

[131] Konstantin Klemm, Christoph Flamm, and Peter F Stadler. "Funnels in energy landscapes". In: *The European Physical Journal B* 63.3 (2008), pp. 387–391.

[132] Oren M Becker and Martin Karplus. "The topology of multidimensional potential energy surfaces: Theory and application to peptide structure and kinetics". In: *The Journal of chemical physics* 106.4 (1997), pp. 1495–1517.

[133] Jonathan Hallam and A Prugel-Bennett. "Large barrier trees for studying search". In: *IEEE Transactions on Evolutionary Computation* 9.4 (2005), pp. 385–397.

[134] Konstantin Klemm, Jing Qin, and Peter F Stadler. "Geometry and coarse-grained representations of landscapes". In: *Recent advances in the theory and application of fitness landscapes*. Springer, 2014, pp. 153–176.

[135] Monte Lunacek, Darrell Whitley, and Andrew Sutton. "The impact of global structure on search". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2008, pp. 498–507.

[136] Gabriela Ochoa, Nadarajen Veerapen, Darrell Whitley, and Edmund K Burke. "The Multi-funnel Structure of TSP Fitness Landscapes: a Visual Exploration". In: *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer. 2015, pp. 1–13.

[137] Pascal Kerschke, Mike Preuss, Simon Wessing, and Heike Trautmann. "Detecting funnel structures by means of exploratory landscape analysis". In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM. 2015, pp. 265–272.

[138] Sebastian Herrmann, Gabriela Ochoa, and Franz Rothlauf. "Coarse-Grained Barrier Trees of Fitness Landscapes". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2016, pp. 901–910.

[139] Gabriela Ochoa and Nadarajen Veerapen. "Additional Dimensions to the Study of Funnels in Combinatorial Landscapes". In: *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. ACM. 2016, pp. 373–380.

[140] Sarah L Thomson, Fabio Daolio, and Gabriela Ochoa. "Comparing communities of optima with funnels in combinatorial fitness landscapes". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM. 2017, pp. 377–384.

[141] Nadarajen Veerapen and Gabriela Ochoa. "Visualising the global structure of search landscapes: genetic improvement as a case study". In: *Genetic programming and evolvable machines* 19.3 (2018), pp. 317–349.

[142] Gabriela Ochoa and Nadarajen Veerapen. "Mapping the Global Structure of TSP Fitness Landscapes". In: *Journal of Heuristics* (2017), pp. 1–30.

[143] Gregory B Sorkin. "Efficient simulated annealing on fractal energy landscapes". In: *Algorithmica* 6.1-6 (1991), p. 367.

[144] Edward D Weinberger and Peter F Stadler. "Why some fitness landscapes are fractal". In: *Journal of Theoretical Biology* 163.2 (1993), pp. 255–275.

[145] Hendrik Richter. "Analyzing dynamic fitness landscapes of the targeting problem of chaotic systems". In: *European Conference on the Applications of Evolutionary Computation*. Springer. 2012, pp. 83–92.

[146] Hendrik Richter. "Scale-invariance of ruggedness measures in fractal fitness landscapes". In: *International Journal of Parallel, Emergent and Distributed Systems* (2017), pp. 1–14.

[147] Stephen Chen and Vincent Lupien. "Optimization in Fractal and Fractured Landscapes Using Locust Swarms". In: *Australian Conference on Artificial Life*. Springer. 2009, pp. 232–241.

[148] C Fonlupt, D Robilliard, and P Preux. "Fitness landscape and the behavior of heuristics". In: *Evolution Artificielle*. Vol. 97. Citeseer. 1997, p. 56.

[149] Peter Merz and Bernd Freisleben. "Memetic algorithms and the fitness landscape of the graph bi-partitioning problem". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 1998, pp. 765–774.

[150] Cyril Fonlupt, Denis Robilliard, Philippe Preux, and El-Ghazali Talbi. "Fitness landscapes and performance of meta-heuristics". In: *Meta-Heuristics*. Springer, 1999, pp. 257–268.

[151] Joshua D Knowles and David Corne. "Towards Landscape Analyses to Inform the Design of Hybrid Local Search for the Multiobjective Quadratic Assignment Problem." In: *HIS* 87 (2002), pp. 271–279.

[152] Marie-Éléonore Marmion, Laetitia Jourdan, and Clarisse Dhaenens. "Fitness landscape analysis and metaheuristics efficiency". In: *Journal of Mathematical Modelling and Algorithms in Operations Research* 12.1 (2013), pp. 3–26.

[153] Katherine Mary Malan and Andries Petrus Engelbrecht. "Characterising the searchability of continuous optimisation problems for PSO". In: *Swarm intelligence* 8.4 (2014), pp. 275–302.

[154] Katherine M Malan and Andries P Engelbrecht. "Particle swarm optimisation failure prediction based on fitness landscape characteristics". In: *Swarm Intelligence (SIS), 2014 IEEE Symposium on*. IEEE. 2014, pp. 1–9.

[155] M Veluscek, T Kalganova, and P Broomhead. "Improving ant colony optimization performance through prediction of best termination condition". In: *2015 IEEE International Conference on Industrial Technology (ICIT)*. IEEE. 2015, pp. 2394–2402.

[156] Fabio Daolio, Arnaud Liefooghe, Sébastien Verel, Hernán Aguirre, and Kiyoshi Tanaka. "Global vs local search on multi-objective NK-landscapes: contrasting the impact of problem features". In: *proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM. 2015, pp. 369–376.

[157] Fabio Daolio, Arnaud Liefooghe, Sébastien Verel, Hernán Aguirre, and Kiyoshi Tanaka. "Problem Features Versus Algorithm Performance on Rugged Multiobjective Combinatorial Fitness Landscapes". In: *Evolutionary computation* 25.4 (2017), pp. 555–585.

[158] Arnaud Liefooghe, Fabio Daolio, Sébastien Verel, Bilel Derbel, Hernan Aguirre, and Kiyoshi Tanaka. "Landscape-aware performance prediction for evolutionary multi-objective optimization". In: *IEEE Transactions on Evolutionary Computation* (2019).

[159] Olaf Mersmann, Mike Preuss, and Heike Trautmann. "Benchmarking evolutionary algorithms: Towards exploratory landscape analysis". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2010, pp. 73–82.

[160] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. "Exploratory landscape analysis". In: *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. 2011, pp. 829–836.

[161] Pascal Kerschke, Holger H Hoos, Frank Neumann, and Heike Trautmann. "Automated algorithm selecton: Survey and perspectives". In: *Evolutionary computation* 27.1 (2019), pp. 3–45.

[162] Bilel Derbel, Arnaud Liefooghe, Sébastien Verel, Hernan Aguirre, and Kiyoshi Tanaka. "New features for continuous exploratory landscape analysis based on the SOO tree". In: *Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*. 2019, pp. 72–86.

[163] Kate A Smith-Miles. "Cross-disciplinary perspectives on meta-learning for algorithm selection". In: *ACM Computing Surveys (CSUR)* 41.1 (2009), pp. 1–25.

[164] Frank Hutter, Lin Xu, Holger H Hoos, and Kevin Leyton-Brown. "Algorithm runtime prediction: Methods & evaluation". In: *Artificial Intelligence* 206 (2014), pp. 79–111.

[165] Sebastian Herrmann. "Determining the difficulty of landscapes by PageRank centrality in local optima networks". In: *Evolutionary Computation in Combinatorial Optimization*. Springer. 2016, pp. 74–87.

[166] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab, 1999.

[167] Paul McMenemy, Nadarajen Veerapen, and Gabriela Ochoa. "How perturbation strength shapes the global structure of TSP fitness landscapes". In: *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer. 2018, pp. 34–49.

[168] John AW McCall, Lee A Christie, and Alexander EI Brownlee. "Generating easy and hard problems using the proximate optimality principle". In: *Proceedings of the companion publication of the 2015 annual conference on genetic and evolutionary computation*. 2015, pp. 767–768.

[169] Jun He, Tianshi Chen, and Xin Yao. "On the easiest and hardest fitness functions". In: *IEEE Transactions on evolutionary computation* 19.2 (2014), pp. 295–305.

[170] David H Wolpert and William G Macready. "No free lunch theorems for optimization". In: *IEEE transactions on evolutionary computation* 1.1 (1997), pp. 67–82.

[171] Yang Chen, Jinglu Hu, Kotaro Hirasawa, and Songnian Yu. "Solving deceptive problems using a genetic algorithm with reserve selection". In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE. 2008, pp. 884–889.

[172] Benoit Mandelbrot. "How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension". In: *Science* 156.3775 (1967), pp. 636–638. DOI: 10.1126/science.156.3775.636. eprint: http://www.sciencemag.org/content/156/3775/636.full.pdf. URL: http://www.sciencemag.org/content/156/3775/636.abstract.

[173] Chaoming Song, Lazaros K Gallos, Shlomo Havlin, and Hernán A Makse. "How to calculate the fractal dimension of a complex network: the box covering algorithm". In: *Journal of Statistical Mechanics: Theory and Experiment* 2007.03 (2007), P03006.

[174] Ivan Zelinka, Oldrich Zmeskal, and Petr Saloun. "Fractal analysis of fitness landscapes". In: *Recent Advances in the Theory and Application of Fitness Landscapes*. Springer, 2014, pp. 427–456.

[175] Stuart A Kauffman. *The origins of order: Self-organization and selection in evolution*. Oxford University Press, USA, 1993.

[176] Sarah L. Thomson, Fabio Daolio, and Gabriela Ochoa. "Comparing Communities of Optima with Funnels in Combinatorial Fitness Landscapes". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '17. Berlin, Germany: ACM, 2017, pp. 377–384. ISBN: 978-1-4503-4920-8. DOI: 10.1145/3071178.3071211. URL: http://doi.acm.org/10.1145/3071178.3071211.

[177] Sébastien Cahon, Nordine Melab, and E-G Talbi. "Paradiseo: A framework for the reusable design of parallel and distributed metaheuristics". In: *Journal of heuristics* 10.3 (2004), pp. 357–380.

[178] Nicholas D Pattengale, Masoud Alipour, Olaf RP Bininda-Emonds, Bernard ME Moret, and Alexandros Stamatakis. "How many bootstrap replicates are necessary?" In: *Journal of computational biology* 17.3 (2010), pp. 337–354.

[179] Matthew S Thiese, Brenden Ronna, and Ulrike Ott. "P value interpretations and considerations". In: *Journal of thoracic disease* 8.9 (2016), E928.

[180] Yu-Qin Song, Jin-Long Liu, Zu-Guo Yu, and Bao-Gen Li. "Multifractal analysis of weighted networks by a modified sandbox algorithm". In: *Scientific reports* 5 (2015).

[181] Chaoming Song, Shlomo Havlin, and Hernan A Makse. "Self-similarity of complex networks". In: *arXiv preprint cond-mat/0503078* (2005).

[182] Shuhei Furuya and Kousuke Yakubo. "Multifractality of complex networks". In: *Physical Review E* 84.3 (2011), p. 036118.

[183] Benoit B Mandelbrot. "Possible refinement of the lognormal hypothesis concerning the distribution of energy dissipation in intermittent turbulence". In: *Statistical models and turbulence*. Springer, 1972, pp. 333–351.

[184] Jin-Long Liu, Zu-Guo Yu, and Vo Anh. "Determination of multifractal dimensions of complex networks by means of the sandbox algorithm". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25.2 (2015), p. 023103.

[185] Witold Kinsner. "A unified approach to fractal dimensions". In: *Cognitive Informatics, 2005.(ICCI 2005). Fourth IEEE Conference on*. IEEE. 2005, pp. 58–72.

[186] Stephanie Rendón de la Torre, Jaan Kalda, Robert Kitt, and Jüri Engelbrecht. "Fractal and multifractal analysis of complex networks: Estonian network of payments". In: *arXiv preprint arXiv:1710.03534* (2017).

[187] Rainer E Burkard, Stefan E Karisch, and Franz Rendl. "QAPLIB — A Quadratic Assignment Problem Library". In: *Journal of Global optimization* 10.4 (1997), pp. 391–403.

[188] Joshua Knowles and David Corne. "Instance Generators and Test Suites for the Multiobjective Quadratic Assignment Problem". In: *Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal, April 2003, Proceedings*. Ed. by Carlos Fonseca, Peter Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. LNCS 2632. Springer, 2003, pp. 295–310.

[189] Robert S Hudson and Andros Gregoriou. "Calculating and comparing security returns is harder than you think: A comparison between logarithmic and simple returns". In: *International Review of Financial Analysis* 38 (2015), pp. 151–162.

[190] Éric Taillard. "Robust Taboo Search for the Quadratic Assignment Problem". In: *Parallel computing* 17.4-5 (1991), pp. 443–455.

[191] Thomas Stützle. "Iterated Local Search for the Quadratic Assignment Problem". In: *European Journal of Operational Research* 174.3 (2006), pp. 1519–1539.

[192] Benoit B Mandelbrot, Adlai J Fisher, and Laurent E Calvet. "A multifractal model of asset returns". In: (1997).

[193] Dogan Corus, Duc-Cuong Dang, Anton V Eremeev, and Per Kristian Lehre. "Level-based analysis of genetic algorithms and other search processes". In: *IEEE Transactions on Evolutionary Computation* 22.5 (2017), pp. 707–719.

[194] Marco Tomassini, Sébastien Verel, and Gabriela Ochoa. "Complex-network analysis of combinatorial spaces: The N K landscape case". In: *Physical Review E* 78.6 (2008), p. 066114.

[195] Gabriela Ochoa and Nadarajen Veerapen. "Mapping the Global Structure of TSP Fitness Landscapes". In: *Journal of Heuristics* 24.3 (2018), pp. 265–294.

[196] Francisco Chicano, Gabriela Ochoa, Darrell Whitley, and Renato Tinós. "Enhancing partition crossover with articulation points analysis". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM. 2018, pp. 269–276.

[197] David Applegate, William Cook, and André Rohe. "Chained Lin-Kernighan for Large Traveling Salesman Problems". In: *INFORMS Journal on Computing* 15.1 (2003), pp. 82–92.

[198] Penelope Hawe, Cynthia Webster, and Alan Shiell. "A glossary of terms for navigating the field of social network analysis". In: *Journal of Epidemiology & Community Health* 58.12 (2004), pp. 971–975.

[199] Shinichi Nakagawa and Holger Schielzeth. "A General and Simple Method for Obtaining R2 From Generalized Linear Mixed-effects Models". In: *Methods in Ecology and Evolution* 4.2 (2013), pp. 133–142.

[200] Sarah L Thomson, Sébastien Verel, Gabriela Ochoa, Nadarajen Veerapen, and Paul McMenemy. "On the Fractal Nature of Local Optima Networks". In: *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer. 2018, pp. 18–33.

[201] John McCall and Andrei Petrovski. "A decision support system for cancer chemotherapy using genetic algorithms". In: *Proceedings of the international conference on computational intelligence for modeling, control and automation*. 1999, pp. 65–70.

[202] Andrei Petrovski. "An application of genetic algorithms to chemotherapy treatment". In: *PhD Thesis*. 1998.

[203] Andrei Petrovski, Alexander Brownlee, and John McCall. "Statistical optimisation and tuning of GA factors". In: *2005 IEEE Congress on Evolutionary Computation*. Vol. 1. IEEE. 2005, pp. 758–764.

[204] Andrei Petrovski, Siddhartha Shakya, and John McCall. "Optimising cancer chemotherapy using an estimation of distribution algorithm and genetic algorithms". In: *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM. 2006, pp. 413–418.

[205] John McCall, Andrei Petrovski, and Siddhartha Shakya. "Evolutionary algorithms for cancer chemotherapy optimization". In: *Computational Intelligence in Bioinformatics 7* (2007), p. 265.

[206] Alexander EI Brownlee, Martin Pelikan, John AW McCall, and Andrei Petrovski. "An application of a multivariate estimation of distribution algorithm to cancer chemotherapy". In: *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. 2008, pp. 463–464.

[207] Tom E Wheldon. *Mathematical models in cancer research*. Taylor & Francis, 1988.

[208] Sui-Man Tse, Yong Liang, Kwong-Sak Leung, Kin-Hong Lee, and Tony Shu-Kam Mok. "A memetic algorithm for multiple-drug cancer chemotherapy schedule optimization". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37.1 (2007), pp. 84–91.

[209] Robert Barbour, David Corne, and John McCall. "Accelerated optimisation of chemotherapy dose schedules using fitness inheritance". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2010, pp. 1–8.

[210] Andrei Petrovski, Alex Wilson, and John Mccall. "Statistical identification and optimisation of significant GA factors". In: *Proceedings of the 5th Joint Conference on Information Sciences. Atlantic City, USA*. Vol. 1. 2000, pp. 1027–1030.

[211] Lokenath Debnath and Dambaru Bhatta. *Integral transforms and their applications*. CRC press, 2014.

[212] Andrei Petrovski and John McCall. "Multi-objective optimisation of cancer chemotherapy using evolutionary algorithms". In: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer. 2001, pp. 531–545.

[213] Minaya Villasana and Gabriela Ochoa. "Heuristic design of cancer chemotherapies". In: *IEEE Transactions on Evolutionary Computation* 8.6 (2004), pp. 513–521.

[214] Zvia Agur, Refael Hassin, and Sigal Levy. "Optimizing chemotherapy scheduling using local search heuristics". In: *Operations research* 54.5 (2006), pp. 829–846.

[215] Gabriela Ochoa, Minaya Villasana, and Edmund K Burke. "An evolutionary approach to cancer chemotherapy scheduling". In: *Genetic Programming and Evolvable Machines* 8.4 (2007), pp. 301–318.