JOHANNES KEPLER
UNIVERSITÄT LINZ | JKU

TNF

Technisch-Naturwissenschaftliche
Fakultät

Applied Fitness Landscape Analysis

DISSERTATION

zur Erlangung des akademischen Grades

Doktor

im Doktoratsstudium der

Technischen Wissenschaften

Eingereicht von:

Dipl.-Ing. (FH) Erik Pitzer

Angefertigt am:

Institut für Formale Modelle und Verifikation

Beurteilung:

Prof. (FH) Priv.-Doz. DI Dr. Michael Affenzeller (Betreuung)

Univ.-Prof. DI Dr. Franz Winkler

Linz, Jänner, 2013

for
Oksana

Acknowledgments

First and foremost I want to thank my adviser Michael Affenzeller. His continuous support, his strategic insight and deep understanding of the topic have helped me tremendously to arrive at a point worthy of a dissertation. However, his support has not only comprised the scientific content of my work but was also of organizational nature. With much forethought and strategic planning, he helped me to be involved in relevant projects or allowed extra time for continuing my research, as well as supported me to attend relevant conferences. So, he really deserves the title "Doktorvater" (doctoral father) as it is called in German, and should be the role model for others.

Another important factor for the success of my thesis was the continuous collaboration with Andreas Beham. Without our repeated and deep discussions, I could have never obtained such a good insight and, with his help, pragmatic view of the subject. He has helped me set up many of the experiments that were important benchmarks for the fitness landscape analysis and was always at hand to discuss further directions. Without his forethought and insight into possible applications, this work would not have been possible.

I also want to thank Franz Winkler wholeheartedly for his valuable time and stimulating discussion that have helped to deeply scrutinize my approaches and get many details right.

Witold Jacak, who started supporting me when I was still an undergrad and believed in me, provided motivational support and has opened many doors that made my academic life much easier. Moreover, the financial support from the University of Applied Sciences Upper Austria that provided a one year sabbatical, where a great deal of the research shown in this thesis was done, was very helpful. In addition, several conference visits for some of the ideas elaborated in this work have been made possible in collaboration with the Josef-Ressel Center "Heurka!".

I also want to thank Stefan Vonolfen for many interesting discussions and possible applications, that has stimulated me to come up with more ideas. Also Stephan Dreiseitl has helped me with some of the more mathematical details and provided moral support.

Without the support of my research group and the marvelous HeuristicLab, I could have never achieved such a plethora of results. In particular the extensibility of HL, thanks to lead developers Stefan Wagner, Gabriel Kronberger, Michael Kommenda and Andreas Beham, I was able to easily integrate new algorithms necessary for all fitness landscape analyses. Later on, the developments of former member Christoph Neumüller and Andreas Scheibenpflug on the distributed computing platform HeuristicLab Hive have enabled me to use the full computing power available at our lab by making it incredibly easy to distribute experiments to our computing infrastructure.

Finally, without so much understanding from my wife Oksana, all the support, and the occasional kick in the pants I probably would have never finished. Thank you!

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die vorliegende Dissertation ist mit dem elektronisch übermittelten Textdokument identisch.

Zusammenfassung

Metaheuristiken sind leistungsstarke Methoden für die Lösung von schwierigen Optimierungsproblemen. Viele verschiedene Methoden wurden entwickelt und zahllose Varianten existieren bereits, von denen jede mehr oder weniger geeignet ist ein bestimmtes Problem zu lösen. Im *No Free Lunch Theorem* wird gezeigt, dass es für zwei Optimierungsalgorithmen immer sowohl ein Problem gibt, wo der erste Algorithmus überlegen ist, als auch ein anderes, wo der zweite Algorithmus besser ist. Es muss also Vorwissen über das vorliegende Problem verwendet werden, um ein geeignetes Verfahren auszuwählen. Daher müssen viele dieser Algorithmen für jede neue Probleminstanz einer bestimmten Problemklasse neu parametriert werden, um gute Resultate in vernünftiger Zeit zu erzielen. Dabei werden problemspezifische Erfahrungen gesammelt, wo oft unklar ist, wie sie auf andere Problemklassen übertragen werden können.

Fitnesslandschaftsanalyse ist eine allgemeine Methodik, um beliebige Optimierungsprobleme zu charakterisieren. Die Grundannahmen der Fitnesslandschaftsanalyse sind sehr allgemein gehalten und bestehen nur aus einer Definition des Lösungsraums, einer Fitnessfunktion und einem Begriff des Zusammenhangs zwischen den Lösungskandidaten, ohne tiefere Einsichten in die Domäne des Problems zu erlauben. Es gibt zwar vielen Arbeiten zu diesem Thema, die meisten beschränken sich aber auf das Studium einzelner Detailaspekte.

In dieser Arbeit wird ein integrativer Ansatz verfolgt, der viele bereits existierende Techniken kombiniert, um ein umfassenderes Verständnis für Optimierungsprobleme zu bekommen. Als Basis dient eine umfangreiche Aufarbeitung bestehender Analyseverfahren, um die volle Diversität und das Potential dieser Ansätze ausnutzen zu können. Darauf aufbauend werden mehrere neue Analysemethoden entwickelt, um komplementäre Aspekte von Fitnesslandschaften zu beleuchten. Insbesondere wird eine Analyse der Isotropie in Angriff genommen, die in früheren Arbeiten oft erwähnt aber nur wenig hinterfragt wird. Dabei werden erste Antworten auf die Frage einer quantitativen Analyse der Isotropie gegeben, also in welchem Ausmaß sich die Eigenschaften einer Fitnesslandschaft in unterschiedlichen Bereichen ändern. Schlussendlich werden frühere Ansätze zu einer allgemeinen Problembeschreibung kombiniert, die dann verwendet werden kann, um automatisch Beziehung zwischen Problemen und Algorithmen zu finden. Es werden umfangreiche Analysen mit schnellen und praktikablen Vorhersagen gemacht, sowohl für die Problemcharakterisierung, als auch für die Problemschwierigkeit oder die Algorithmendominanz und erste Resultate in der automatischen Algorithmen- und Parameterauswahl erzielt.

In dieser Arbeit soll ein Schritt in Richtung automatischer Algorithmen- und Parameterauswahl gemacht werden. Die anfängliche Wahl eines geeigneten Optimierungsverfahrens muss dann nicht auf einen Standardalgorithmus fallen sondern es kann durch eine schnelle und automatische Analyse, mit geeigneten Messungen und aufbereiteten Erfahrungen eine bessere Wahl getroffen werden. Auf dem Weg dorthin kann die Fitnesslandschaftsanalyse helfen, das Verständnis von Problemen zu vertiefen, die bereits mit problemspezifischen Ansätzen untersucht wurden, indem deren Charakteristika von einem allgemeineren Standpunkt aus betrachtet werden können. Mit Hilfe der Fitnesslandschaftsanalyse, deren Resultaten und Nomenklatur wird es Forschern verschiedener Domänen ermöglicht ein gemeinsames Vokabular zur Beschreibung ihrer Probleme zu verwenden und Ideen auszutauschen, wie diese möglichst effizient gelöst werden können.

Abstract

Metaheuristics are powerful methods for solving hard optimization problems. Many different methods have been developed of which countless variants exist. Each of which are more or less applicable to a particular problem. The *no free lunch theorem* even shows that for two arbitrary optimization algorithms there will always be cases, where either algorithm dominates the other depending only on the particular problem instance. Therefore, preexisting knowledge and experience about the problem at hand have to be used to choose an appropriate method. For each new problem instance in each problem class, these algorithms usually have to be parameterized and tuned to achieve good results in reasonable time. In this process problem-specific experience is gained for which it is not clear how it can be transferred to other problem classes.

Fitness landscape analysis is a general methodology to characterize any optimization problem. The underlying assumptions for the formulation of a fitness landscape are very basic and comprise only the definition of the solution space, a fitness function and a notion of connectedness between solution candidates without allowing any further reasoning about the underlying problem domain. While a large body of research exists on this topic, most efforts have concentrated on particular details of fitness landscapes.

In this work a unifying approach is used, combining many of the previously existing techniques to arrive at a more comprehensive understanding of optimization problems. The base for this work is founded by an extensive review of existing fitness landscape analysis methods to fully exploit the diversity and potential of these techniques. On top of these techniques, several new analysis methods are developed to elucidate complementary aspects of fitness landscapes. In particular, the analysis of isotropy is tackled which is often mentioned in previous works but has received relatively little attention in terms of research in the past. First answers to the question of quantifying isotropy in fitness landscapes are provided. In other words, the variability of different properties over different areas of a fitness landscape are studied. Finally, many previous techniques are combined into a general problem description scheme that can then be used to automatically reason about the relationship between problems and algorithms. Moreover, large scale analyses are carried out with fast and practical results for problem characterization, hardness prediction, algorithm dominance prediction, and first results in automatic algorithm and parameter selection.

This work should be a first step towards fully automatic algorithm and parameter selection. Therefore, initial algorithm selection does not have to fall back to more or less universal one-size-fits-all default algorithm. Instead fast and automatic fitness landscape analysis can provide the necessary measures that can be used together with formalized experiences to arrive at a better-informed initial choice. On the way to this vision, fitness landscape analysis, can help to deepen the understanding of problems that have already been studied using problem-specific approaches, by allowing a more distant general view of the problem's characteristics. With the help of fitness landscape analysis results and nomenclature, researchers from different domains receive a common vocabulary for describing their problems and to exchange ideas about how these problems can best be tackled.

Contents

Chapter 1

Introduction

While in the past the focus has been mainly on optimization methods, algorithms, and variants, increasingly the desire to understand why they work so well has emerged. The *Fitness Landscape* is one of the most fundamental notions that has been used to define the underlying structure of an optimization problem and, therefore, must be the starting point for such investigations.

In essence, a fitness landscape can be described as the perception of the solution space by an optimization method and is often portrayed as real landscape with hills and valleys. Many analogies have been drawn in the past to arrive at a seemingly intuitive understanding of a connected solution space. However, these visualizations are sometimes oversimplifications and can lead to dangerous non-generalizable conclusions. Therefore, even though, this metaphor is in widespread use, one should carefully evaluate the difference between idealized image and the actual structure and cohesion of optimization spaces.

This dissertation will focus particularity on generally applicable implementations of fitness landscape analysis methods, that can be readily applied to new optimization problems in order to provide automatic characterization and comparison of problem instances. The focus will be on concrete solutions and implementations with immediate practical value.

1.1 History and Informal Definition

The idea of a fitness *landscape* was first described in Sewall Wright's frequently cited work about the *solution space* of evolution (Wright, 1932). In this case, the "fitness landscape", Wright was describing, was the solution space of genes and the fitness function was the actual genetic fitness of a group of individuals. Natural species are actually performing a more difficult kind of optimization in a changing environment. However, the terminology stuck and has been carried over to the vocabulary of optimization as many other terms from genetics.

Nowadays, many examples of this notion can be found throughout the literature about optimization. Informally, it describes the gut feeling of a researcher about the structure of the solution space. For this reason, the connectivity is often implicitly the most natural distance definition on the solution space.

Figure 1.1 shows a depiction of the typical landscape metaphor. Here, one of Schwefel's test functions as described in (Affenzeller et al., 2009) was used to model the topology of a landscape with the functional value as the height of a two dimensional coordinate grid.

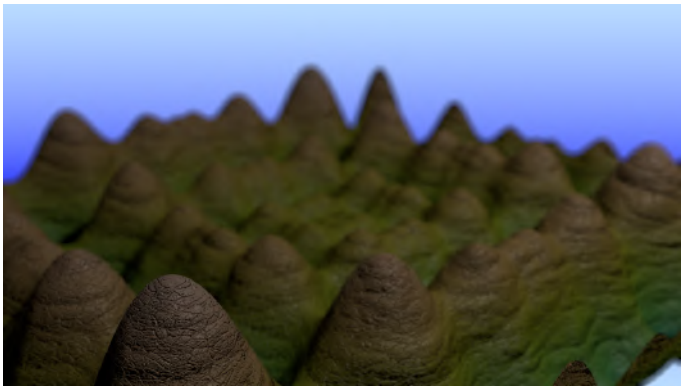


Figure 1.1. The Landscape Metaphor: Two dimensional functions, such as Schwefel's test function shown here, provide a prime example for the fitness landscape metaphor.

1.2 Motivation

The main motivation for stepping back and allowing only the very fundamental concepts of an optimization problem without exploiting domain specific knowledge might at first seem counter-intuitive. However, several powerful applications become possible through these restrictions. One important application is the selection of appropriate stochastic optimization methods with the help of fitness landscape analysis. Another application is a more detailed and fundamental problem understanding that is comparable across application domains and, finally, the mere effort to formalize the description and characterization of optimization problems yields a common vocabulary for an overarching understanding. These motivations are discussed in the following in more detail.

1.2.1 Metaheuristics and the No Free Lunch Theorem

For many practical optimization problems the complexity of an exact solution is prohibitively large. This means that approximation methods have to be employed. In particular stochastic optimization methods are very popular for these large problems, where numerical approximation is not feasible. While the convergence to the best possible solution cannot be guaranteed, many times a very good solution is sufficient. However, due to their stochasticity, one could perceive algorithms of this class all as different incarnations of guided random search. Each of which has advantages in certain circumstances. This idea is captured in the *No Free Lunch Theorem*, in (Wolpert and Macready, 1997). Theorem 1 is their first No Free Lunch Theorem, where $P(d_m^y|f,m,a)$ is the performance, or in other words the best value d_m^y, found for a cost function f using m samples with algorithm a. The theorem, therefore, states, that the average performance for any pair of algorithms a_1 and a_2 is equal over all optimization problems f given the same sample size m.

Theorem 1 $\forall(a_1,a_2)\sum_f P(d_m^y|f,m,a_1) = \sum_f P(d_m^y|f,m,a_2)$.

Accordingly, the No Free Lunch Theorem is often paraphrased as follows: For any pair of optimization algorithms there is one problem instance for which the first one is superior and a second instance for which the second algorithm is superior.

As a consequence, there is no best stochastic optimization algorithm that can outperform any other algorithm in general. However, through the use of fitness landscape analysis it seems possible to get an advantage by selecting a better suited optimization strategy beforehand. This can be achieved by extracting characteristics of the problem instance and comparing it to the previously extracted characteristics of other instances and the performances of different algorithms on these instances.

1.2.2 Problem Understanding

The purpose of studying fitness landscape analysis is most often a comprehensive understanding of an optimization problem. Often this desire stems from the solution trials of a single problem class, which has led to many problem-specific analysis results. However, once researchers started to successfully apply the same optimization methodologies to different problem classes a more general description was necessary. This is where the notion of a fitness landscape is most suitable, abstracting away the structure of the particular problem classes, leaving only the most general elements, which can then be uniformly analyzed and compared.

Therefore, of course, fitness landscape analysis can be used to obtain insights into problem classes. It has to be noted, however, that problem class-specific characteristics are often easy to obtain and can provide great insight. Fitness landscape analysis, on the other hand, is readily applicable to any new problem class without much additional work and is usually directly targeted at optimization performance-critical aspects. Therefore, once a frame of reference in the realm of fitness landscape analysis has been established for a particular problem class it can provide helpful insights.

1.2.3 Formalization and Vocabulary

Another important purpose of fitness landscape analysis is the formalization of a joint vocabulary for describing optimization problem characteristics that are universally applicable and pertain specifically to the performance-critical intrinsics of problem instances.

While practitioners of one domain might categorize their problems using one set of rules, different rules will likely exist for other domains. Stepping back to the general core aspects of optimization problems by using the appropriate terminology allows a much easier exchange of information and experience between problem domains and, therefore, faster understanding of optimization methods in general and other problem domains in particular.

1.2.4 Algorithm and Parameter Tuning

In addition to the previously mentioned more qualitative uses of fitness landscape analysis, these techniques can also be used for a more direct application. The measures and results obtained by fitness landscape analysis and the resulting problem instance characterizations can be used to find similar problem instances and, therefore, facilitate the selection of suitable combinations of algorithms and parameters for unseen problem instances. While this requires a large database of problem instances and their fitness landscape analysis results as well as their optimization results when subjected to different optimization algorithms, it has shown great potential to simplify the currently tedious process of selecting algorithms and tuning parameters for new problem classes. Additionally, as this algorithm selection and parameter tuning usually requires a great deal of expertise, it can bring the topic of efficient optimization to a broader audience as more support for selection and tuning can be provided automatically.

1.2.5 Hyper Heuristics

Another related topic is hyper heuristics. While traditional optimization methods usually act similarly when used with the same parameter settings, hyper heuristics can adapt to the problem instance at hand. Hyper-heuristics were first introduced in (Denzinger et al., 1997) and are described in detail in (Ozcan et al., 2008). Usually, they employ systems of heuristics that use or generate other heuristics in order to solve a particular problem or generate a solver for a problem instance or class. During the course of finding or constructing such a solver, however, the problem instances themselves are usually solved countless times until a good solution method has been

found. This will eventually provide a good solution method for a problem. The methodology described in this work, however, refrains from solving the problem multiple times. Instead good initial algorithm and parameter choices are derived a-priori by utilizing easily obtainable problem intrinsics and a large reference database of previous successful applications to similar problems.

1.2.6 Self-Adaptive Optimization Methods

Finally, another big potential for fitness landscape analysis is the direct incorporation into optimization algorithms. By continuously monitoring and measuring certain fitness landscape characteristics during the optimization process itself, a great wealth of self-adaptive algorithm variations is possible. Instead of "post-mortem" analysis of a failed algorithm run through a human expert, the incorporation of a-priori and "on-line" fitness landscape monitoring can open the doors for many new strategies of automatic algorithm adaptation.

1.3 Taxonomy

1.3.1 Distinction and Contrast to Other Work

The goal of this work is to provide a practical tool set of analysis methods and prediction systems to help compare and categorize problem instances and, with the help of previous solution approaches, choose algorithms and parameters for new optimization problems. Particular focus is put on methods with a high potential of re-usability that allow the application to new and never studied problem classes in a short time. Therefore, certain analysis methods are not used here, even though, they would have a high potential.

Problem-Specific Analysis is often the method of choice to discern between problem instance specific parameter settings for optimization algorithms. However, these methods cannot, if at all, be applied to other problem classes and, therefore, require additional up-front effort before they can be used on a new problem instance of a new problem class.

Theoretical analysis of fitness landscapes has provided very interesting insights into many optimization problems. While the obtained insights and derived properties might be very well comparable across different problem classes, most of the results are again, problem class-specific, and a considerable effort has to be expended before the methodology can be re-applied to different classes.

1.3.2 Connection and Differentiation to Heuristic Algorithms

An interesting connection between fitness landscape analysis and optimization algorithms exists on a different level: While optimization algorithms try to find a *good solution* within minimal time, with varying choices of "good" and "minimal", fitness landscape analysis methods are trying to find *good insights* into problem instances within minimal time, again with varying choices of "good" and "minimal". Therefore, development of analysis methods, despite a different final outcome, parallels very much the development of optimization algorithms. The major difference is that optimization algorithms usually have much more tangible results with a difference to some best known prior results or an established known optimum. Fitness landscape analysis does not have such an easily quantifiable performance indicator. Only through indirect use of fitness landscape measures can the performance be estimated.

1.4 Overview

This work is structured as follows. After this introductory text, Chapter 2 states and explains a formal definition of fitness landscapes and discusses their basic properties. It is followed by an extensive review of existing perspectives, formalisms and methodologies to analyze fitness landscapes in Chapter 3.

The heart of this work is constituted by Chapter 4, where new analysis methods are introduced and combined. It contains new views of basic properties such as the basin analysis in Section 4.1, or new analysis methods such as the investigation and a first attempt of quantification of isotropy in fitness landscapes in Section 4.2 culminating in the formalization of a new test landscape with tunable isotropy. Additionally, new and existing analysis methods are combined in Sections 4.4 and 4.5 to yield more powerful insights that can provide the information necessary for automatic algorithm selection and parameter tuning, which is described in Section 4.6. Finally, in Section 4.8, a new successful approach for the direct choice between two algorithms is proposed.

Chapter 5 starts with a detailed description of the employed methods and some notes on the developed software and tools, followed by the discussion of several large scale applications of fitness landscape analysis, for the measurement and prediction of problem similarity and problem hardness for different classes of optimization problems. Finally, Chapter 6 contains a summary and the conclusions obtained within this work, followed by a collection of open issues and directions for future work.

Several substantial parts of this work have been previously presented at conferences or as part of other publications; all of which have been peer reviewed. Large parts of the initial survey of state of the art fitness landscape analysis methods, as presented in Chapter 3, have been distilled into a book chapter in (Pitzer and Affenzeller, 2011). The results of the investigations about basins of attraction in Section 4.1 have been presented at the UK Conference on Artificial Intelligence in (Pitzer et al., 2010). The preliminary implementation and some first thoughts on fitness landscape isotropy as described in Section 4.2 have been presented at the EuroCAST 2011 in (Pitzer et al., 2011). Finally, many of the results concerning the predictions generated with fitness landscape analysis in Sections 5.3, 4.4, 4.5, 4.6, 4.7, and 4.8 have also been presented or are accepted for the presentation at notable conferences (Pitzer et al., 2012b,a, 2013a,b).

Chapter 2

Theoretical Foundations

As the base for further investigations, fitness landscapes are formally defined in this chapter. Moreover, their fundamental properties are investigated as potential leverage points for further analysis which will be introduced in the subsequent chapters.

2.1 Formal Definition

The formal definition of a fitness landscape consists of three components.

- The search space, or the set of all possible valid solution candidates for a particular problem, denoted as $\mathcal{S}$. Typical solution spaces are the set of binary vectors $\mathcal{C}(n)$ of a certain length, the set of real vectors $\mathbb{R}^n$ with a certain number of dimensions n, or the set of permutations of a certain length $\mathbb{P}_n$. However, for general fitness landscape analysis there is no restriction on the underlying set of solution candidates.
- The actual fitness function is mostly defined as $f : \mathcal{S} \to \mathbb{R}$ and gives the assignment of a fitness value to every solution candidate.
- The last component of a fitness landscape definition is its connectivity. This is the most interesting and most controversial part. Very often this connectivity is implicitly included as the most natural neighborhood or distance relation between solution candidates. For example in a real vector, Euclidian distance or in a permutation a swap operation. However, the choice of neighborhood structure or connectivity can have a drastic influence on the shape of the landscape. In general it is denoted with $\mathcal{X}$. Other frequent notations are $N : x \to \mathcal{P}(x)$, when restricted to a neighborhood-based connectivity or $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ if a distance-based connectivity is desired.

In summary, the triple in Equation 2.1 fully defines a fitness landscape.

$$\mathcal{F} := (\mathcal{S}, f, \mathcal{X}) \tag{2.1}$$

2.1.1 Search Space

The first important question when formulating an optimization problem is the selection of the search space often called encoding. This represents the theoretical set of all possible solution candidates. These candidates need not be good solutions to the given problem but at least represent a feasible solution. The choice of this set can already have a drastic influence on the performance of an optimization algorithm. One example is the choice of whether to make a component of the

search space discrete or continuous. Staying with a discrete set of choices can limit the search space to an enumerable set of choices. However, it can create difficult non-linear jumps that are hard to overcome. If the same variable was encoded continuously, the optimization algorithm might be able to "squeeze" past such a sticking point and the fitness landscape might be perceived as "smoother".

2.1.2 Fitness Function

The fitness function is probably the most obvious constituent of a fitness landscape. Typically, the fitness is given as a real value and the rest of this work will only look at this case, however, this is only the most pragmatic definition of a fitness function. In reality several variations are possible. In certain cases, the fitness value itself can be integer and hence present a discrete set of different fitness values. Most optimization methods also use only a single fitness value. In multi-object optimization scenarios, however, the "fitness value" might actually be a vector. Multi-object fitness landscape analysis is not discussed in this work and has only recently been started to undergo serious investigations (see Garrett and Dasgupta, 2008).

2.1.3 Connectivity

The last component, the *connectivity*, of a fitness landscape defines the structure and coherence between solution candidates. The solution space itself is just an unordered set of items. It is the connectivity that creates a structure. The simplest connectivity structures are unconnected and fully connected neighborhoods which can be defined over any set. Slightly more interesting are collections of subsets which create the first interesting structure or, in fact, the first landscape-like structure.

The most prevalent and well-studied structure, however, is the *neighborhood connectivity*. In this case for every solution candidate $x \in \mathcal{S}$ there is a set of neighbors $N(x)$ with $N : \mathcal{S} \to \mathcal{P}(\mathcal{S})$, where $\mathcal{P}$ is the power set of $\mathcal{S}$. Neighborhood connectivities are often used for discrete solution spaces such as permutation vectors, integer or binary vectors. In these cases, the neighborhood is induced by the employed mutation or manipulation operator and the underlying encoding or interpretation of the solution candidates' representations. A simple example for a neighborhood connectivity is the bit flip operator applied to binary vectors. In this case, every pair of binary vectors that can be reached via a single bit flip is directly connected. A different mutation operator, such as 2-bit flip operator would create a tighter coupling or a larger neighborhood for any binary vector and, therefore, create a different connectivity structure and, hence, a different fitness landscape.

Another kind of connectivity, which is typically found in continuous spaces, is distance based. In continuous spaces, especially in uncountably infinite spaces, the "neighborhood" of every solution candidate is often infinite itself. One option, to stay with the notion of a neighborhood, is to define a certain ε radius that defines the neighborhood of a solution candidate:

$$\begin{array}{rccl} N: & \mathbb{R}^n & \to & \mathcal{P}(\mathbb{R}^n) \\ & x & \mapsto & \{y \mid y \in \mathbb{R}^n, d(x,y) < \varepsilon\} \end{array} \tag{2.2}$$

In other words, the *distance* becomes the defining property of the neighborhood structure of a continuous fitness landscape. Moreover, if the choice of ε is left free, a more complex coherence structure is obtained that is purely distance based. However, most analysis methods currently rely on a discrete neighborhood relation which can be obtained using the ε neighbors as shown in Equation 2.2. While neighborhood-based connectivity is straightforward to understand and analyze, distance-based neighborhoods are more complex as they involve an additional distance function. Again, these distance function can be deduced or coupled with manipulation operators of the solution candidates.

In fact, even more complex coherence structures have been proposed, such as coherence structures based on recombination instead of mutation in (Gitchoff and Wagner, 1996; Stadler and Wagner, 1998; Wagner and Stadler, 1998). This connectivity type creates a hyper graph of solution candidates. A hyper graph is a generalization of a graph. Instead of edges between only two nodes in the graph, in a hyper graph, a single edge can have more than two nodes. Again, these edges can be directed having a set of originating nodes and a set of destination nodes. In particular, the so-called *P-structures* introduced by Stadler and Wagner have a pair of originating nodes, i.e. the parents, and an arbitrary set of destination nodes, i.e. the offspring. This opens the door to analysis of population based optimization methods as their recombinational "view" of the fitness landscape is different to purely manipulative strategies.

While analysis based on distance or even recombination coherence is possible, this work concentrates on simple manipulative or neighborhood based analysis methods. Because of the difficulties that arise when deviating for this simple assumption, a certain neighborhood structure is often implicitly assumed. For example, in binary vector spaces the bit flip neighborhood is often implicitly assumed, or in a permutation based encoding, a swap operation is usually the underlying generator of the fitness landscape. It should be remembered, however, that the choice of connectivity can have a large impact on the analysis results and, therefore, the perception of the fitness landscape.

2.1.4 Encoding and Representation

While the interpretation or phenotype of a solution candidate is not formally a part of the fitness landscape, it should be noted that it constitutes an important concept. The interplay of solution encoding which is part of the fitness evaluation and the mutation operation has been highlighted in (Reeves and Rowe, 2003) with a nice example. In one fitness landscape the fitness of a binary vector is the real value using a simple binary-coded decimal. In the second fitness landscape the fitness is the gray-encoded real value of the binary vector. Now, two different mutation operators are thinkable: The first one is the simple bit flip operator, the other one modifies the bit string according to its encoded integer value. While in the first landscape variant the two mutation operators behave quite differently, in the second landscape variant they are have the same effect, as can be seen in Table 2.1.

Table 2.1. Comparison of different binary encodings

Binary	Gray	Decimal
000	000	0
001	001	1
010	011	2
011	010	3
100	110	4
101	111	5
110	101	6
111	100	7

2.2 Fitness Landscape Analysis Fundamentals

After the formal definition and delineation of what a fitness landscape is, the formalism can be used to arrive at some fundamental notions, some of which are very well known and often even analyzed outside the scope of fitness landscape analysis.

2.2.1 Modality: Local and Global Optima

In terms of optimization, probably the most interesting features of a fitness landscape are its optima. The definition of the global optimum (or global optima in general) can be directly derived from the definition of a fitness landscape as shown in Equation 2.3.

$$\mathcal{O}(\mathcal{F}) := \{x \mid x \in \mathcal{S}, \forall\, (y \in \mathcal{S})\ f(x) \geq f(y)\} \tag{2.3}$$

It should be noted that, usually, when speaking in terms of fitness, maximization of fitness is desired, therefore, the global optimum is a maximum unless noted otherwise. One notable exception is Section 4.1, where optima are generally seen as minimums.

This is the set of values (or the single value) that is sought when performing optimization. While symbolic optimization and theoretical analysis are often able to deliver these values, complex large-scale problems can be more difficult so solve to optimality. In these cases, solution quality is often sacrificed for solution speed and the obtained result will typically be a good *local optimum*. This specimen of an optimum is more difficult to define as it only has to outperform other solution candidates locally. Therefore, in addition to the solution space and fitness function required in Equation 2.3 of the global optimum, this time the connectivity structure is incorporated as well. Therefore, technically different definitions are obtained for local optima depending on the connectivity structure of the fitness landscape. However, in essence, these definitions are conceptually equal and capture the notion of a solution candidate being better than other "near-by" solution candidates.

In case of a neighborhood-based fitness landscape, the definition is straightforward. Every solution candidate with higher or equal fitness than all its neighbors is a local optimum as shown in Equation 2.4.

$$\widehat{\mathcal{O}_N}(\mathcal{F}) := \{x \mid x \in \mathcal{S}, \forall (y \in N(x))\, f(x) \geq f(y)\} \tag{2.4}$$

It has to be noted that a local optimum can either be part of a plateau or saddle point or, on the other hand, it can be a strict local optimum with superior fitness than all its neighbors. This minor point, whether a local optimum is a strict local optimum can cause some confusion. However, when recalling the definition of a local optimum: "A solution that cannot locally be improved", it becomes immediately clear that a local optimum does not have to be strict in general.

It gets slightly more complicated for distance-based connectivities, where, analogously to continuous spaces, an *ε-neighborhood* can be used for the definition of the set of local optima as shown in Equation 2.5 using the definition of an ε-neighborhood in Equation 2.2.

$$\widehat{\mathcal{O}_{N_\varepsilon}}(\mathcal{F}) := \{x \mid x \in \mathcal{S}, \forall (y \in N_\varepsilon(x))\, x \geq y\} \tag{2.5}$$

Even for recombination spaces, it is relatively straightforward to define local optima: Here again, a direct neighborhood relation is given, that can be used to define a locally optimal solution candidate by making sure all possible recombinations with a different solution candidate have an inferior fitness value. Equation 2.6 shows a possible definition for a local optimum in the recombination space using the recombination operator $\mathcal{R}$ that defines the set of all possible offspring of two parent solution candidates as shown in Equation 2.7.

$$\widehat{\mathcal{O}_P}(\mathcal{F}) := \{x \mid x \in \mathcal{S},\ \forall (y \in \mathcal{S}, c \in \mathcal{R}(x,y))\, f(x) \geq f(c)\} \tag{2.6}$$

$$\mathcal{R} : \mathcal{S} \times \mathcal{S} \to \mathcal{P}(S) \tag{2.7}$$

Local optima are among the first and simplest characteristics that have been studied in fitness landscape analysis. Despite their elementariness, interesting features can be extracted and used for further analysis as further elaborated in Section 3.3.1

2.2.2 Plateaus and Neutral Areas

Continuing from local optima, other basic structures of fitness landscapes are plateaus and neutral areas. While a simple local optimum constitutes a single solution candidate that is superior to its neighbors other interesting solution candidates might happen to have neighbors with equally good solutions. In this case, a plateau has been found.

The general definition of plateaus and neutral areas is a bit more complicated than the definitions of local optima as this time we have to deal with a *connected* set of solution candidates. Moreover, depending on the type of *connectivity* of a fitness landscape, the formalization of such a connected subset will be different:

- For a discrete space with a neighborhood-based connectivity, a connected set of solution candidates is a set $S' \subseteq \mathcal{S}$ of which all elements are transitively connected. Such a set can be defined constructively as follows:
  - A singleton set with one solution candidate x is a connected set,
    i.e. $\text{connected}(\{x\})$.
  - The union of a connected set S and a solution candidate x which is a neighbor of any element of the connected set is also a connected set,
    i.e. $\text{connected}(\{x\} \cup S) :\Leftrightarrow \exists(y \in S')\, N(x, y) \wedge \text{connected}(S')$.
- For continuous spaces, we have to resort to topology to obtain a satisfactory definition of connectedness as described in (Pitzer and Affenzeller, 2011): A topologically connected set is an open set that cannot be represented as the union of two or more disjoint non-empty open sets (see Munkres, 2000).
- Also for recombination spaces, such as P-structures, a connected set can be reduced to the definition of neighborhood-based connectivities.

With the definition of a connected set as substrate, the notions of neutral areas in general and plateaus in particular can new be formalized.

$$\text{neutral}(S') :\Leftrightarrow \text{connected}(S') \ \wedge \ \forall(x, y \in S')\ f(x) = f(y) \tag{2.8}$$

Equation 2.8 shows the simple definition of a neutral area which is a connected subset of the solution space with equal fitness. To define a proper plateau, the definition of an *outer border* of a connected set is needed. Intuitively, these are the points surrounding the connected set, which are different to the *boundary* of a subset which might still be part of the subset itself. Equation 2.9 gives the simple definition of the border of a connected set for neighborhood-based connectivities.

$$\text{border}(S') := \{x \mid x \in \mathcal{S},\ x \notin \mathrm{S}',\ \exists(y \in S)\ x \in N(y)\} \tag{2.9}$$

The definition of the outer border for continuous neighborhoods is, again, more complicated as there exist different cases depending on whether the subset is open or closed. However, taking the special case of our previous definition of a connected open set from the last paragraph, the outer border, or in this case equivalently, the boundary of an open set is the difference of its closure and itself as shown in Equation 2.10. If the set is not open, the more general definition of a boundary ($\delta S = \overline{S} \setminus S^O$) does not represent the outer border as is required for our case.

$$\delta S = \overline{S} \setminus S \tag{2.10}$$

Finally, given the definitions of a neutral area and its border, plateaus can be formalized as neutral areas with no superior fitness values in its border as shown in Equation 2.11, where $f(S)$ is the single fitness value throughout the neutral area.

$$\text{plateau}(S) :\Leftrightarrow \text{neutral}(S) \wedge \forall(x \in \text{border}(S))\ f(x) < f(S) \tag{2.11}$$

These subsets of the solution space are of particular interest as the plateaus exhibit similar properties as local optima and the neutral areas can easily interfere with optimization strategies. Due to its lack of "slope" an optimization method can easily become trapped in a large neutral area. On the other hand, a carefully designed optimization scheme could even exploit neutral areas to spread out and access a larger range of interesting solution candidates on its outer border without loss of quality.

These neutral areas can be used to derive interesting properties when measured and, hence, constitute an important aspect of fitness landscapes. More information on the analysis of neutral areas is contained in Sections 3.8, where neutrality is measured and compared with other features to characterize solution spaces.

2.2.3 Basins

Another fundamental property of fitness landscapes are the basins of attraction of all optima. In a previous publication, this seemingly trivial component was investigated in great depth and interesting conclusions have been drawn (see Pitzer et al., 2011).

Before giving a formal definition of a basins of attraction, we will look at an informal description: A basin of attraction is the area around an optimum, where immediate "attraction" to that particular optimum is "felt". Typically, this is the *slope* of the landscape going downward—or upward, depending on the optimization objective—guiding the direction towards the next optimum. In one-dimensional space, this gives a very intuitive and easily comprehensible image which is sketched in Figure 2.1. However, once the complexity of the search space increases, in particular if the number of dimensions increases, the immediate understanding of a basin of attraction is not so clear anymore as shown in Figure 2.2.

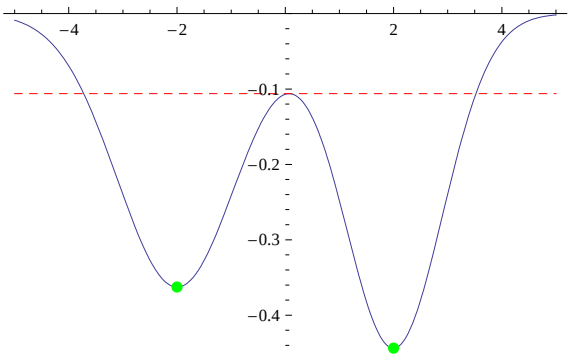


Figure 2.1. 2D Basin of Attraction: The local optima shown as dots are surrounded by their basins of attraction. The dashed line is the "energy barrier" between these two optima.

Therefore, in Section 4.1 these seemingly simple constructs are examined in detail. Using an alternative definition for local optima, makes basins of attraction more tangible and allows us to derive interesting characteristics of solution spaces.

2.2.4 Barriers

After the discussion of basins, the next fundamental property of fitness landscape are *barriers* between local optima. For an informal understanding we can go back to Figure 2.1, where we have a two-dimensional depiction of local optima. The area immediately surrounding these optima, the basins, are discussed in the previous section. Between these basins we find the barriers which stems from the same terminology of a minimization problem: The "energy" required to overcome one

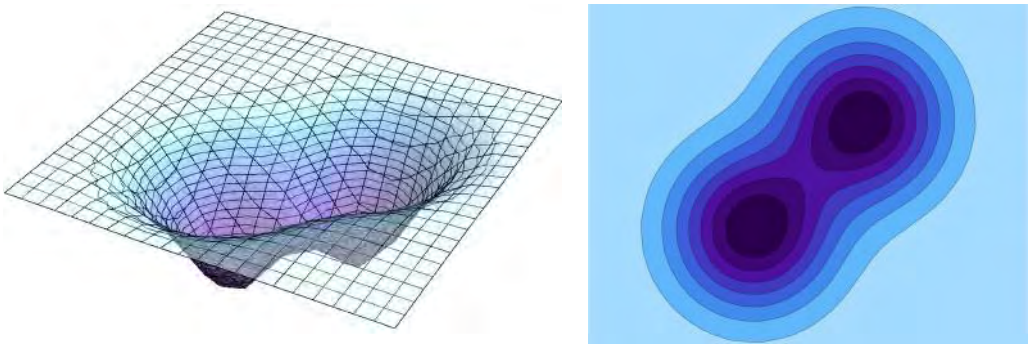


Figure 2.2. 2D Basin of Attraction: While the central areas directly surrounding the optima have an obvious convergence preference, the lighter areas, are still inside a compound basin, although it is ambiguous which optimum these areas belong to.

basin and drop into another is called a barrier. In Figure 2.1 it is highlighted as dashed line. This definition was first presented in (Stadler, 2002) and can be formalized as shown in Equation 2.12, where a simple definition of the fitness barrier between two arbitrary points is given. To obtain this value, the maximum fitness of any path between the two points has to be calculated and then the minimum of them has to be used, which gives the most energy efficient path between two points. Usually, the fitness barrier will be most interesting between local optima and the resulting distribution of barrier levels can be analyzed. Further details of barrier analysis methods are presented in Section 3.3.1.

$$B_f(x, y) := \min \{\max \{f(x) \mid x \in p\} \mid p \text{ path from } x \text{ to } y\} \tag{2.12}$$

As can be seen from this definition, and should also be clear from the definition of basins of attraction, it can quickly become very tedious and computationally expensive to perform an analysis of basic properties of fitness landscapes. For this reasons many alternative approaches have been developed in the past to elucidate the internal structure of fitness landscapes. The following chapter contains a comprehensive compilation of existing approaches to fitness landscape analysis and highlights the different perspectives that have been tried in the past to uncover the internal structure of optimization problems.

2.2.5 Landscape Variants

One important issue arising especially in the light of heuristic optimization algorithms is the strong dependence on the neighborhood relation. As stated in (Jones, 1995), in fact, "one operator defines one landscape". This statement can be immediately verified as shown in Figure 2.3, where the different neighborhood definitions can have a large influence on the obtained fitness landscape measures.

On the one hand, this allows a comparative analysis of different neighborhoods or the employed mutation or move operators used in iterative optimization algorithms. On the other hand, it can provide an obstacle for the objective comparison of problem instances. Since problem instances cannot be directly compared with each other any more, only with a detour over a certain neighborhood definition.

However, if these different neighborhood definitions are used to create their own frame of reference, inference results obtained within that frame are very well comparable to other neighborhoods as discussed in further detail in Sections 4.4.1 and Section 6.2.4.

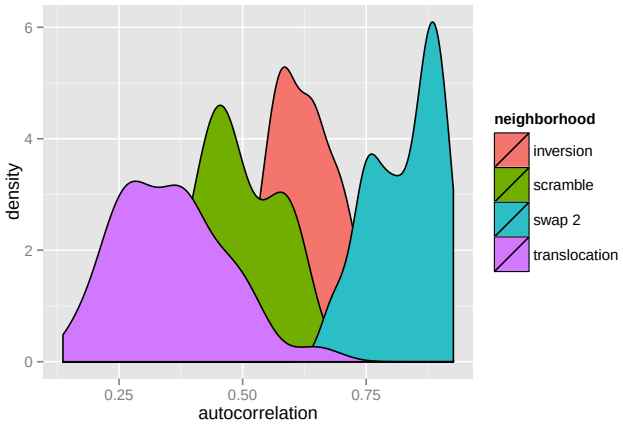


Figure 2.3. Landscape Variants: The distributions of fitness landscape measures change drastically depending on the considered neighborhood. Shown here is the distribution of auto correlation values throughout the quadratic assignment problem library. In this figure, also the increasing smoothness of the increasing smaller neighborhoods, from translocation, over scramble to inversion and finally, the smoothest, swap-2 neighborhoods can be observed.

Chapter 3

State of the Art

Many different types of analysis have been proposed for fitness landscapes in the past. This chapter contains a comprehensive compilation of techniques that can be found in the literature. Initially, researchers often tried to find "the" method for measuring, characterizing and comparing fitness landscapes with each other. However, it quickly became apparent that some techniques provide insights that are only relevant for certain types of optimization problems or only for some types of optimization algorithms. So, over time, it was realized that all of these techniques should be seen as complimentary to each other and should be used in concert to achieve the most comprehensive elucidation of a problem's characteristics (Pitzer and Affenzeller, 2011; Jones and Forrest, 1995). Parts of this chapter have been published in (Pitzer and Affenzeller, 2011), where many existing methods of fitness landscape analysis have been summarized.

3.1 Categorization of Analysis Methods

The existing analysis methods can be loosely grouped into three categories:

- The first group comprises *exhaustive* analyses of the fitness landscape and provide the most complete picture. However, they are only applicable to problems of "academic" size. While the obtained results can be educational, they are difficult to apply in practice and it might be even dangerous to estimate larger problem's characteristics using the features extracted from smaller, exhaustive analyses.
- The second group of analysis methods are *theoretical* techniques. These methods are typically applied at a problem class level and require substantial human interaction, in contrast to other analysis methods that rely solely on the general definition and possibly an interchangeable implementation of the formalizations given in Chapter 2.
- Finally, the third group of analysis methods are stochastic, sampling-based techniques which have the advantage of increased speed in comparison to exhaustive analysis but have the risk of being biased and incomplete. In a certain sense they are comparable to what metaheuristics accomplish in finding "good" solutions, stochastic methods can be used to find "good" insights into problem characteristics.

In this work a focus is put on stochastic analysis methods as they are the most practically applicable. Most stochastic analysis methods can be generically implemented to allow easily interchangeable sampling and evaluation components, which can greatly facilitate the analysis of new problem domains.

The major advantage of exhaustive analysis methods is the associated completeness. While the cost is high, the resulting complete characterization of a solution space give definite and complete information. However, this analysis reflects only the solution space of a single problem instance. For the survey of many problem instances of a certain problem class this can be very educational, however, the computational cost is mostly prohibitive. One particularly instructive example can be found in Section 4.1, where a more or less exhaustive analysis of smooth continuous solution spaces yields interesting insights into basins of attraction and their role in different metaheuristic algorithms.

The second class, theoretical analyses, have the main disadvantage that they often require human effort and are difficult to conduct automatically. If many different problem classes need to be analyzed, this can easily become a severe shortcoming. On the other hand, if problems of the same class have to be studied repeatedly, the generality of a theoretical analysis of a whole problem class can often be readily applied to new instances with little effort. Therefore, these analysis techniques are most suitable for in-depth analysis of important problem classes that are repeatedly used. Most examples of theoretical analysis are based on Stadler's and Weinberger's decomposition of fitness landscapes into elementary landscapes (see Weinberger, 1996; Stadler, 1995). Very interesting is also the subsequent derivation of exact measures for certain characteristics that have previously only been available through sampling (see Chicano et al., 2011, 2010).

Finally, stochastic fitness landscape analysis seems to be the most practical method. The major advantages are the short runtime and the minimal effort to apply them to new problem classes. This combination makes it good candidate for the exploration of new solution spaces and neighborhoods. As long as the formulation of the analysis relies only on the general components of a fitness landscape any kind of stochastic measurement can be used. This can also give a great variety of different perspectives. The major downside in comparison to the other methods is the stochasticity itself. While exhaustive and theoretical analysis methods can often yield exact and definite answers, stochastic methods deliver merely an estimate.

3.2 Sampling Methods

A fundamental but general problem of stochastic analysis methods is the selection of a representative sample. This task becomes particularly difficult if hardly any knowledge about the universe is known a-priori and the resulting sample should be representative for certain types of universe explorations.

In the past, mostly two classes of sample generation techniques have been used:

- The first class of sampling techniques mimics the path of many *trajectory-based* optimization methods and creates a continuous string of adjacent solution candidates.
- The second class is also based on exploration strategies of optimization methods and creates a dispersed sample which can then be extended by either recombination operations or again local neighborhoods.

In general, it cannot be determined which of these sampling strategies will yield the better insight. However, most subsequent analysis methods rest upon trajectory-based samples.

3.2.1 Trajectories

Sampling trajectories for fitness landscape analysis resemble very much the path of typical optimization algorithms and create a path inside a solution space or, in other words, a sequence of adjacent solution candidates. The method of choice to derive these sequences is usually a so-called *"walk"*. Formally, the notation of a sequence $\{x\}_{i=0}^{n}, x_i \in \mathcal{S}$ is used to describe a sequence of

solution candidates and $f(\{x\}_{i=0}^{n}) = \{f(x)\}_{i=0}^{n} = \{v\}_{i=0}^{n}, v_i \in \mathbb{R}$ to described the sequence of resulting fitness values.

For the analysis of fitness landscapes, many different types of walks are possible. A walk typically starts at a random solution candidate and uses the definition of the fitness landscape's neighborhood to "walk" to adjacent solution candidates. Depending on the scheme to choose a certain neighboring solution candidate, several different types of walks are possible. Formally, for linear trajectories, the difference between walk variants is the neighbor selection scheme. A very general definition of this neighbor selection function is shown in Equation 3.1, which takes the definition of the fitness landscape and a single solution candidate to select a neighboring solution candidate.

$$\text{Succ} : \mathcal{F} \times \mathcal{S} \to \mathcal{S} \tag{3.1}$$

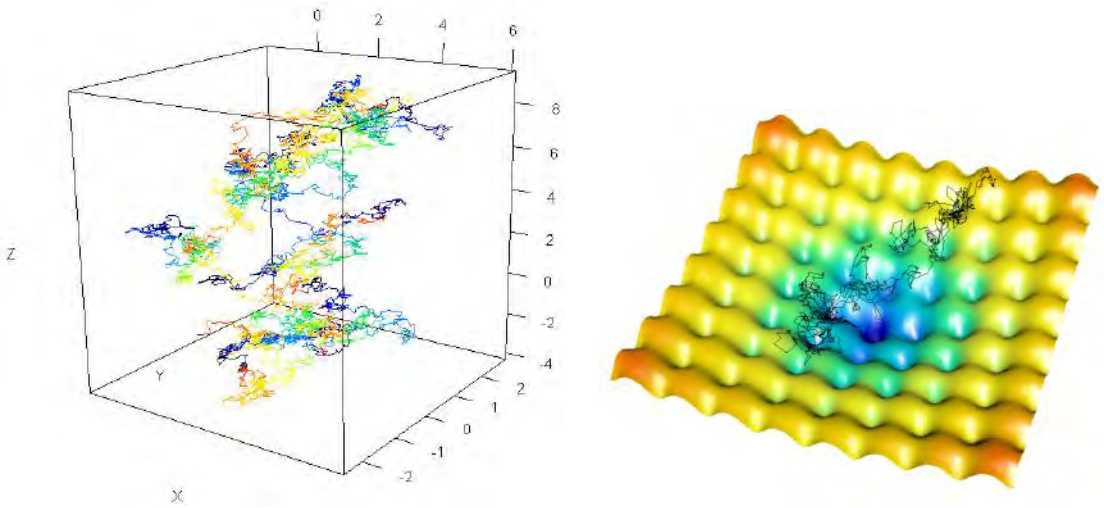
A simplified version of this walk neighborhood function, as shown in Equation 3.2 will use only the direct expansion of a solution candidate's neighborhood and select an appropriate successor using the landscape's fitness evaluation function.

$$\text{Succ} : \mathcal{S} \times \mathcal{X} \times f \to \mathcal{S} \tag{3.2}$$

3.2.1.1 Random Walk

A very popular type of trajectory-based analysis is the random walk. Figure 3.1 shows two examples of a random walk. Beginning at a random starting point, an adjacent or nearby solution is chosen at random.

Figure 3.1a shows a random walk in a three dimensional solution space, while Figure 3.1b shows a random walk in the solution space of a two-dimensional Ackley function (Ackley, 1987) which is defined in Equation 5.1 Section 5.1. In this example the solution space has only two dimensions but the fitness value of each solution candidate is added as a third dimension.



(a) Unrestricted 3D Real Vector Space **(b)** On the Ackley Test Function

Figure 3.1. Examples of Random Walks

As can be seen from these examples, a random walk is chaotic and does not follow any pattern or any "shapes" of the fitness function. Therefore, it is a perfect tool to obtain an unbiased sample and is used frequently for the analysis of fitness landscapes (see Jones, 1995; Merz, 2004).

At the same time, the indifference of solution quality, previously praised as unbiasedness, can also become a problem as a random walk does not tend to select "interesting" areas of a fitness

landscape. Therefore, while the sample is representative for the fitness landscape as a whole, it typically contains only few solution candidates that would be seen during an optimization attempt. Therefore, it has been proposed e.g. in (Jones, 1995) to include samples along other trajectories for a more complete analysis.

3.2.1.2 Adaptive Walk

To obtain a sample that is more reminiscent of an optimization algorithm, an adaptive walk can be performed that is very similar to a local search algorithm (see Battiti et al., 2008; Hoos and Stützle, 2004). From a given set of neighbors, one of the better ones is chosen to continue the trajectory. For exploratory purposes, it is often sufficient to examine only a subset of the actual neighborhood of a solution candidate and arrive at a "mildly" adaptive walk that does not converge too quickly.

While this type of walk provides a sample of solution candidates that better resembles the set of solution candidates seen during an optimization process it suffers from the same problem: premature convergence. This means that only a short trajectory will typically be produced.

One way to overcome this shortcoming is to intelligently restart an adaptive walk and merge the results of these repeated analyses. This approach is presented in Section 4.2.5. Another approach is described in the next section, where an adaptive walk is alternated with an anti-adaptive walk.

3.2.1.3 Up-Down Walk

The simple alteration of an adaptive walk in the "right direction" followed by an adaptive walk in the "wrong direction" creates what was coined as an *up-down* walk in (Jones, 1995). This sampling trajectory exhibits very favorable properties:

- Due to the adaptive nature of the sampling, the visited solution candidates and their neighborhood relation are similar to trajectories obtained through simple optimization schemes. This can provide a favorable bias towards relevant areas of the solution space and relevant relationships of neighboring solution candidates.
- Due to the repeated reversal of optimization direction, the area of analysis tends to be much greater in a single run because the trajectory does not become trapped in a local optimum. One alternative is to have fixed cycle lengths for the *up* and *down* phases, the other alternative is to simply continue in one direction until a local optimum or a local "anti"-optimum is reached at which time the optimization direction is changed.
- Another advantage is the continuous connectedness of a long adaptive trajectory that is difficult to obtain otherwise. Of course, some sections have their direction reversed which can complicate further analysis.

In summary, the up-down walk provides a valuable addition to the arsenal of sampling strategies and can usually provide better and easier insights than multiple adaptive samples. In previous experiments in Chapter 5, this sampling strategy is often as good as or even better than the ubiquitous random walk in terms of information gathering, when used in isolation. It should be noted, however, that it is not advocated here to use a single walk type. On the contrary, as many different "perspectives" as possible should be used to obtain a rich view of the fitness landscape.

3.2.1.4 Neutral Walk

Another worthwhile addition to the library of trajectory sample generators is the neutral walk first described in (Reidys and Stadler, 1998). Neutrality is briefly introduced in Section 2.2.2 and

discussed in depth in Section 3.8. To understand neutral walks, it is instructive to imagine a fitness landscape being composed of flat layers of different sizes as shown in Figure 3.2. In some circumstances, these flat areas might be infinitesimally small, or in discrete spaces comprise a single point. In many real problem instances, however, these cases do occur and are worth being explored.

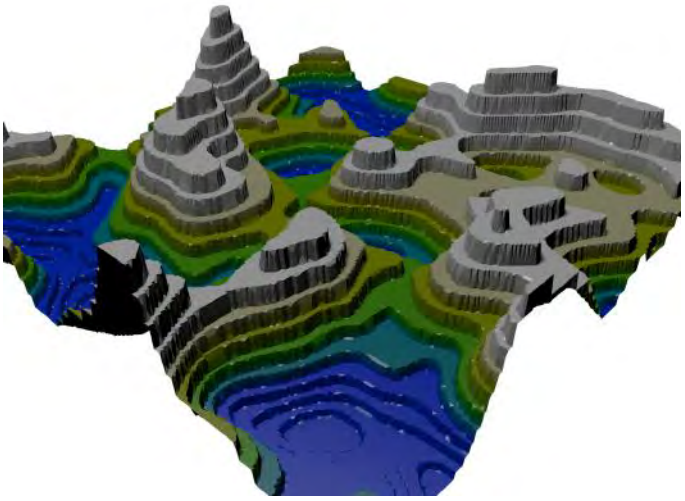


Figure 3.2. Visualization of Neutrality: Every fitness landscape can be seen as composition of many neutral layers, each of which is completely flat.

Assuming this structure, a neutral walk starts at one of these flat layers and continues to explore only this "level" of the landscape. While, in terms of fitness, obviously nothing new can be discovered, the shape and extent of this neutral area itself can be characterized. If the random starting point is not part of a neutral area, the neutral walk cannot continue and has to be aborted. Therefore, to effectively characterize the neutrality of a fitness landscape additional measures have to be taken:

- One possibility is to simply restart at different locations and attempt a neutral walk. The advantage of this approach is its simplicity and proper characterization of only neutral areas without considering a single non-neutral step.
- Another possibility is to step through non-neutral areas as part of the overall trajectory. This approach is even simpler and allows the characterization of the ratio between neutral and non-neutral areas in relation. However, the analysis of neutral areas is "polluted" by the non-neutral parts and has to be separated analytically.

As with most of the analysis methods, there is not yet a clear answer as to which measure is more useful in general, and probably, one could even apply a variant of the No Free Lunch Theorem (see Wolpert and Macready, 1997) to conclude that there is no analysis method that is superior under all circumstances. In fact, different sampling strategies can highlight different aspects and should, therefore, be considered in concert.

3.2.1.5 Summary

While the random walk was specifically designed to explore different areas that are encountered during a trajectory-based optimization run, it is usually only able to explore rather uninteresting parts of the landscape. On the other hand, it provides the only unbiased sample that still has the same coherence structure as trajectory-based optimization methods. As a complement, adaptive, or up-down walks will usually traverse more "interesting" or higher quality regions while on the

other hand being biased and might not provide such a full overview of the landscape. These techniques are further complemented by neutral walks which attempt to tackle yet another perspective and concentrate on flat—or more scientifically, neutral—areas of a landscape. Together these different perspective will yield the greatest insight as they elucidate different aspects of a fitness landscape.

In practice, another important choice has to be made. This is the *length* of the sampling trajectory. While no general figure can be given, a figure similar to the typical number of evaluations in an optimization method seems reasonable. In our experience, however, for many practical problems with a more or less coherent structure many aspects can be revealed after 100,000 steps or even after 10,000 steps already (see Pitzer et al., 2012a; Jones, 1995).

It has to be noted, however, that stochastic sampling as the method of choice for landscape exploration bears the danger of missing the important parts as already sated in Section 3.1.

3.2.2 Population-Based Methods

A complementary sampling approach diverges from a continuous string of neighboring or nearby samples and resembles population-based optimization algorithms such as genetic algorithms (see Holland, 1975) or evolution strategies (see Rechenberg, 1973).

In its simplest form a population-based sampling strategy is simply a parallel execution of several trajectories. However, given the added notion of a population, many more possibilities arise. On the one hand instead of indifferently continuing every parallel "trajectory" one can chose to stop progressing some trajectories and bifurcate other, more interesting ones like in an Evolution Strategy. Even more interestingly and contrasting to trajectory based exploration, using a recombination operator a totally different kind of sample can be obtained. Instead of relying on neighbors, existing individuals are *recombined* into new ones, that resemble its predecessors. This typically happens in genetic algorithms which provide a rather different set of advantages and shortcomings than trajectory-based optimization methods and should, therefore, also provide a different perspective of the fitness landscape.

One difficulty of sampling methods that rely on recombination is the possible bias and loss of genetic diversity which is more difficult to maintain than with purely random, trajectory-based methods. On the other hand, this tendency can be exploited to give a different, and a possibly more interesting, sample of the fitness landscape.

In the past, in (Pitzer and Affenzeller, 2011), very slowly converging population-based optimization methods have been used to obtain samples of fitness landscapes that are relatively diverse but with a strong bias towards higher quality individuals. This can help to better understand those parts of a fitness landscape that will later be explored by optimization methods, since most of these methods quickly move away from very low quality random solution candidates. In this respect, a slightly biased sampling method might be advantageous as features which are relevant to optimization methods can be perceived more easily.

Finally, it has to be noted, however, that even though, a more practically relevant sample of the fitness landscape might be obtained, it still bears the danger of oversight. Therefore, we cannot be reminded too often that a comprehensive fitness landscape analysis, especially of an unknown problem structure, should always also include an unbiased sample.

3.3 Measures

In this section, preexisting measures and quantification methods of fitness landscape characteristics are reviewed. To better highlight the contribution of this thesis, newly developed methods by the

author have been extracted into Chapter 4 even though some are based on or tightly coupled with some of the measures described in this chapter. However, appropriate cross references have been made to guide between the sections.

3.3.1 Fundamental Measures

When it comes to fundamental measures, usually, an exhaustive analysis has to be performed. However, some of these methods can be adapted to obtain stochastic results for larger solution spaces.

3.3.1.1 Solution Space Size

Probably one of the simplest measures of a solution space is its size. For finite and countable solution spaces, such as for combinatorial optimization problems, the size of the solution space is actually one of the most important factors for predicting problem difficulty, as shown in (Pitzer et al., 2012b). For infinite, especially for uncountably infinite solution spaces, still, the number of dimensions can serve as good indicator of problem difficulty. Therefore, problem size constitutes the most fundamental and most important quantifiable characteristic of a problem instance's difficulty as is shown multiple times in Chapter 4.

Based on the neighborhood structure or an inherent distance relationship between solution candidates $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$, the solution space can be measured. Similar to the raw size of a solution space, based on the number of possible solution candidates or the number of dimensions, the diameter of a search space can be defined as shown in Equation 3.3. This can give a first estimate of the traversal time using a specific neighborhood or solution manipulation operator, given that the distance measure is based on this neighborhood or manipulation operator.

$$\operatorname{diam}(\mathcal{S}) := \max \{d(x, y) \mid x, y \in \mathcal{S}\} \tag{3.3}$$

Using the distance measure, a first insight into the connectivity of the solution space can be obtained by analyzing the distribution of the set of normalized distances in the solution space as shown in Equation 3.4. This distribution will reveal the connectedness or "tightness" of a solution space. Usually, a tighter solution space is accompanied by a larger direct neighborhood structure so that every solution candidate can be reached from every other solution candidate with fewer steps. Whether this choice is beneficial for optimization purposes, however, cannot be answered in general, of course.

$$\left\{ \frac{d(x, y)}{diam(\mathcal{S})} \;\middle|\; x, y \in \mathcal{S} \right\} \tag{3.4}$$

Finally, these measures can serve as a first approach towards measures that can help in the normalization of other fitness landscape measures, that inherent measures either the neighborhood size, the solution space size or a combination thereof.

3.3.1.2 Local Optima

From an optimization standpoint, the first subject of interest is the analysis of local and global optima $\mathcal{O}(\mathcal{F})$, which is often abbreviated to just $\mathcal{O}$. This gives rise to another collection of quantifiable properties:

- One interesting first question is simply the *number of local optima*. If a problem is unimodal, i.e. has only one local optimum it will probably be easier to solve than a problem with several local optima, which in turn might still be easier than a problem with many local optima (see Altenberg, 1997).

- However, not only their number but also the *distribution of local optima* is a quantifiable characteristic of a solution space (see Talbi, 2009). If all optima are close together, for example, a problem might sill be easier than one with widely distributed optima. This property can be measured by analyzing the distance distribution of the set of local optima, shown in Equation 3.5.

$$\left\{ \frac{d(x,y)}{diam(\mathcal{S})} \;\middle|\; x,y \in \mathcal{O} \right\} \tag{3.5}$$

- Of equal interest is the fitness distribution of local optima, shown in Equation 3.6. For practical purposes it can play an important factor to be able to consider achievable solution quality given only a certain available computational effort. Knowing that many local optima have very good quality in comparison to the global optimum, for example, can help to make a more educated choice in this regard.

$$\{f(o) \mid o \in \mathcal{O}\} \tag{3.6}$$

- The combination of both sets, shown in Equation 3.7, allows for an even deeper analysis, such as whether good local optima are close to each other. One interesting question, that can be answered positively for some problem classes as shown in (Merz and Freisleben, 2000), is the analysis of whether local optima form a structure that can be exploited or figuratively "lead the way" to the global optimum. One popular analysis method, based on this idea is the fitness distance correlation analysis described in Section 3.7.

$$\left\{ \Big(d(x,y),\, |f(x) - f(y)| \Big) \;\middle|\; x,y \in \mathcal{O} \right\} \tag{3.7}$$

3.3.1.3 Basins

While in Section 2.2.3 the formalization of basins of attractions is given, these important structural components of fitness landscapes can also be used to characterize and quantify fitness landscapes. In the field of fitness landscape analysis, however, basins of attraction have not yet received their fair deal of attention, therefore, a new analysis method is presented in Section 4.1.

3.3.1.4 Barriers

Interestingly, the analysis of barriers between basins of attraction has received more attention than the analysis of basins of attraction. More specifically, in (Stadler, 2002), an interesting method to explore and characterize the connection between different fitness barriers is proposed.

In Table 3.1 an example of a combined basin and barrier analysis is shown. The underlying fitness landscape is depicted in Figure 3.3a. The table contains both the definition of the test function giving the optima, and their fitness value as well as the relative volume of the optima's basins of attraction. The detailed explanation of these values is given in Section 4.1. For now, we are only interested in the analysis of the barriers which have also been calculated and are shown in the lower region of the table. Every pair of local optima is separated by a certain fitness barrier, i.e. the worst fitness that has to be accepted on the best path from one optimum to the other, refer to Equation 2.12 in Section 2.2.4 to review the definition.

These distances form an ultrametric (Gouvêa, 1993), i.e. a special kind of metric space, where the triangle inequality (Eq. 3.8) is replaced with a stronger inequality shown in Equation 3.9.

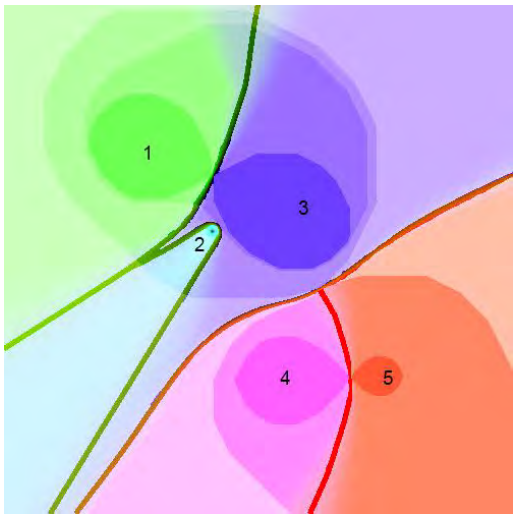
$$\forall (x,y,z \in \mathcal{S})\; d(x,z) \leq d(x,y) + d(y,z) \tag{3.8}$$

$$\forall (x,y,z \in \mathcal{S})\; d(x,z) \leq \max\{d(x,y), d(y,z)\} \tag{3.9}$$

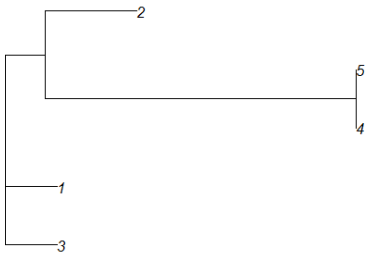
Table 3.1. Combined Basin and Barrier Analysis: The first few rows describe the fitness function itself. Each column contains another optimum. The first three lines show the location, fitness and attraction volume of each of the optima. The second section contains the fitness barriers between all pairs of optima.

	1	2	3	4	5
Optimum	(-4.04, 4.32)	(-1.84, 1.08)	(1.08, 1.88)	(1.36, -4.76)	(4.56, -4.6)
f(x)	-,14	-50,55	-,21	-,21	-,07
Volume	23%	9%	33%	15%	20%
2	-,0350				
3	-,0462	-,0350			
4	-,0039	-,0039	-,0039		
5	-,00389	-,0039	-,0039	-,0609	

This can be explained as follows: When looking at the barrier between x and z, any detour through another local optimum y would have at least the same barrier if not more. Using these values and the fact that the set of barriers form an ultrametric space, a phylogenetic inference method, which is also used in biology, can be used to build a tree that shows the mutual relationship between all local optima with regard to the barriers separating them. Figure 3.3b shows such an example: The distances have been used to infer a phylogenetic relationship (Nei, 1972) using the neighbor joining method introduced in (Saitou and Nei, 1987). By visual inspection, the tree obtained through neighbor joining can be verified to relate to the valley structure in this simple two-dimensional example. While it will become much more difficult to visually determine this structure for more complex solution spaces, the computation does not change and can still be applied, allowing an improved understanding of the coherence and separation of local optima by studying their basins and the barriers between them.



(a) Multi Normal Test Function with Highlighted Barriers



(b) Barrier Tree: The result of phylogenetic inference of the barriers between local optima can reveal a structure representing the relation between local optima.

Figure 3.3. Barrier Analysis

This analysis can be combined with the analysis of direct location and quality differences of local optima, as described in Section 3.3.1.2, to determine whether one optimum is not only difficult to reach in terms of neighborhood distance but also in terms of intermediate fitness barriers.

An efficient algorithm for calculating fitness barriers is introduced in Section 4.1 alongside the

determination of basins of attraction.

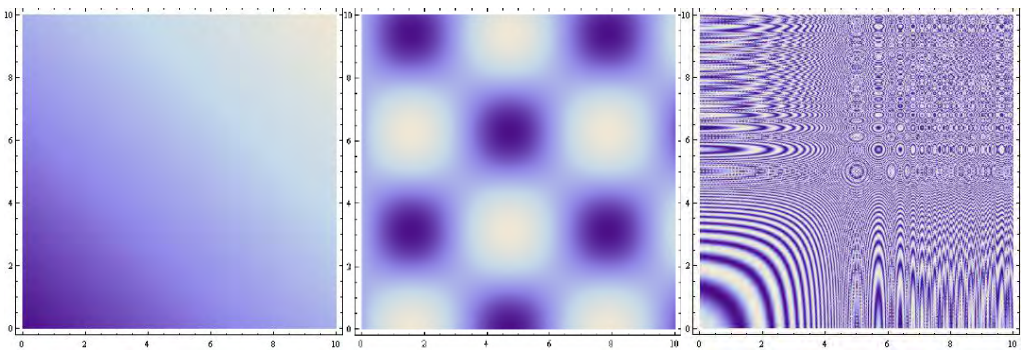
3.3.1.5 Fundamental Trajectory Measures

Based on the different trajectories introduced in Section 3.2.1, several other fundamental measures can be derived. These are sub-walk lengths and distances. While for a random walk no useful intermediate way points are usually measured, neutral walks and up-down walks are subdivided into intermediate sections, i.e. the flat sections in a neutral walk or the upwards and downward sections of an up-down walk. These sub-walks have some very fundamental properties such as their relative lengths or the traversed distances. These fundamental properties of sub-walks are explained in more detail in Section 5.3.2.3.

3.3.2 Auto Correlation

Before we delve into the topic of autocorrelation, a note on the underlying idea or visual metaphor is in order. When optimization strategies were developed for multimodal functions, the performance of optimization algorithms on different fitness landscapes was imagined to correlate to the "ruggedness" of the solution space. Many optimization functions have been initially proposed and tested on real vector spaces, where the two-dimensional variants are susceptible to visual inspection and interpretation. For example, the sphere function was an important benchmark for the design of evolution strategies (see Rechenberg, 1973) or genetic algorithms have been evaluated with various other real vector functions (see de Jong, 1975; Schwefel, 1995).

In Figure 3.4 different levels of ruggedness are shown in a density plot of several two-dimensional functions. A unimodal function shows no ruggedness (Fig. 3.4a), while a composition of smooth trigonometric functions gives a highly multimodal function that still appears visually smooth (Fig. 3.4b). Further increasing the oscillation frequency, the function appears more and more rugged (Fig. 3.4c), however, depending on the optimization method this might still be interpreted as low ruggedness, as the function is still differentiable and smooth on a smaller scale. To arrive at a truly rugged landscape, a fitness function cannot have any correlation between neighboring solution candidates and becomes completely random.



(a) None **(b)** Low **(c)** High

Figure 3.4. Different Levels of Perceivable Ruggedness on a Two-Dimensional Test Function

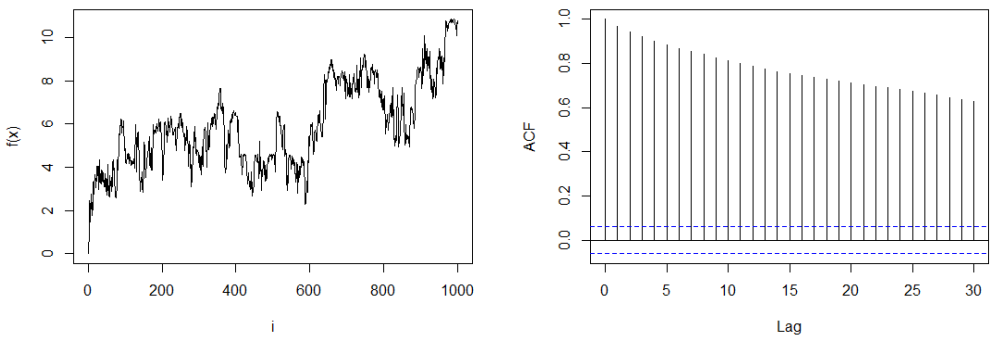
This visual analysis of function smoothness or ruggedness, therefore, needs a framework for objective and comparable quantification. In the following subsections several measures that have been more or less directly derived from the idea of this ruggedness are introduced.

3.3.2.1 Auto Correlation Function

As stated in the previous paragraph, the notion of ruggedness is connected to the correlation between neighboring solution candidates. For this reason, the auto correlation of a fitness function with itself was one of the first measures for a quantification of the ruggedness of a fitness landscape (see Weinberger, 1990; Stadler and Schnabl, 1992).

In general, the computation of the auto correlation of a whole fitness function is very expensive as all possible neighborhood relationships would have to be evaluated. A theoretic approach to model this universally measured neighborhood correlation is introduced in Section 3.4. To measure autocorrelation stochastically, first, a trajectory is sampled in the fitness landscape. For an unbiased sample, usually a random walk is used. This gives a sample of neighboring solution candidates whose neighborhood correlation can then be evaluated: The sequence of neighboring solution candidates $\{x_i\}_{i=0}^n$ is used to create a sequence of fitness values $\{f(x_i)\}_{i=0}^n$ which is then used to calculate the auto correlation.

Figure 3.5a shows the fitness values of the random walk on the 2D Ackley Function in Figure 3.1b. Figure 3.5b shows the resulting auto correlation function (ACF). This function was created by calculating the correlation between one fitness value and subsequent fitness values with increasing steps.



(a) Random Walk Fitness Values **(b)** Corresponding Auto Correlation Function

Figure 3.5. Calculation of Auto Correlation

The calculation of auto correlation is given in Equation 3.10, where $\mathrm{E}[x]$ and $\mathrm{Var}[x]$ are the expected value and the variance of the sequence $\{f(x_i)\}_{i=0}^n$ which is abbreviated as f_i. It is simply the cross-correlation of the fitness values of the random walk with itself.

$$\rho(\tau) := \frac{\mathrm{E}[f_i \cdot f_{i+\tau}] - \mathrm{E}[f_i]\mathrm{E}[f_{i+\tau}]}{\mathrm{Var}[f_i]} \tag{3.10}$$

This function gives the average correlation for the sample trajectory between points that are a certain number of steps τ away from each other. Except for pathological cases this correlation will decrease with increasing distance. As discussed in Section 3.3.2.2, often, this will be a simple exponential decay. The shape of this function can give interesting insights, especially in cases for problem classes, where this exponential decay does not occur.

One important precondition for the stochastic estimation of the auto correlation is the assumption of isotropy, which means that the fitness landscape is the result of a stationary random process (see Weinberger, 1990). More on the topic of isotropy is given in Section 3.11 and a method to verify this assumption and measure isotropy is developed in this work and described in Section 4.2.

3.3.2.2 Auto Correlation Coefficient

Often, the whole auto correlation function (ACF) is not used for further analysis (see Weinberger, 1990; Stadler, 1995, 1996). In practice, one particularly interesting piece of information is the auto correlation *coefficient*, which is simply the auto correlation after one step. This measure gives the average correlation between neighboring solution candidates over the whole landscape and is denoted as $\rho(1)$. This can arguably be a gross oversimplification but has been quite successfully employed in the past to predict hardness for problems of certain classes (see Manderick et al., 1991; Merz and Freisleben, 1998, 2000; Czech, 2008; Tavares et al., 2008; Wang et al., 2006; Mattfeld et al., 1999). For large scale analysis, a single number can be much more convenient than the comparison of a whole series. Moreover, it was shown in (Weinberger, 1990) that in isotropic landscapes, the auto correlation function is an exponential function and, therefore, the rest of the function does not contain any more information than the initial step.

The actual auto correlation coefficient λ shown in Equation 3.11 was introduced much later in (Angel and Zissimopoulos, 1998a).

$$\lambda := \frac{1}{1 - \rho(1)} \tag{3.11}$$

This transformation puts typical auto correlation values on a more linear scale and is, therefore, interesting for practical analysis. However, its relative ordering remains the same as the underlying auto correlation function value $\rho(1)$.

3.3.3 Correlation Length

Another measure that can be derived from the auto correlation function is the correlation length, which can be seen as the complementary singular measure to the auto correlation coefficient. It has to be noted that in the past there have been several different notions of correlation length (Hordijk, 1996; Weinberger, 1990; Manderick et al., 1991; Lipsitch, 1991).

Instead of taking the value of the first step, the distance of the last significant step is used. Using the assumption of a stationary stochastic process, and, hence, statistical isotropy of the fitness landscape, the correlation length can directly be obtained from the auto correlation coefficient as $1/\rho(1)$ which is shown in (Weinberger, 1990; Stadler, 1996).

3.3.3.1 Statistical Correlation Length

Another, more general, way of measuring correlation length was proposed in (Hordijk, 1996). There, instead of stopping right before the value where the correlation has dropped below zero, the correlation length is measured only up to statistical significance. Given the stochastic nature of the underlying random walk that was used to generate the sample, this seems to be the soundest approach.

As noted in (Hordijk, 1996), statistically significant difference from zero is usually measured using a two-standard-error bound (see e.g Judge et al., 1988). Therefore, a correlation in the range of $(-2/\sqrt{n}, 2/\sqrt{n})$, where n is the length of the random walk should already be considered as not significant and, therefore, the last step before the auto correlation falls within this range gives the statistical correlation length.

3.3.4 Box-Jenkins Time Series Analysis

An interesting addition to the plain correlation analysis of (Weinberger, 1996) was proposed in (Hordijk and Stadler, 1998), where a sequence of trajectory samples is taken as time series and sub-

jected to an analysis using the Box-Jenkins approach (Box et al., 1970) to obtain an autoregressive-moving average model (ARMA) (see Whittle, 1951). There, the correlations at different time-lags are used to create a predictive model using two components.

- The first component is an *autoregressive model* that uses past values to create a regression model for the following value. Equation 3.12 shows the formulation of an autoregressive model of order p, AR(p), in which it is tried to model a future value using the past p values, where φ_i are the parameters, c is a constant and the remaining stochastic variable ε_t is white noise, i.e. it has $\mathrm{E}[\varepsilon_t] = 0$ and $\mathrm{Var}[\varepsilon_t] < \infty$.

$$y_t = c + \sum_{i=0}^{p} \varphi_i y_{t-i} + \varepsilon_t \tag{3.12}$$

- The second component is a *moving average model* of order q, MA(q), where θ_i are the parameters, μ is the expected value of y_t, and the ε_i are again white noise errors terms.

$$y_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{3.13}$$

Combining these two components, AR(p) and MA(q), gives an autoregressive-moving average model ARMA(p, q) as shown in Equation 3.14.

$$y_t = c + \varepsilon_t + \sum_{i=0}^{p} \varphi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{3.14}$$

Using the approach described in (Box et al., 1970), appropriate values for p and q can be determined as follows:

1. The first step is *model identification* in which the autocorrelation and partial autocorrelation function are examined to chose values for p and q or leave one of the components out completely. The partial autocorrelation function of a time series can be informally described as the "remaining autocorrelation" between elements y_t and y_{t-i} after removing intermittent correlations between y_{t-1} to y_{t-i+1}.
2. The second step is *parameter estimation* for φ and θ using either non-linear least-squares or maximum likelihood estimators.
3. Finally, in a *model checking* step, the fitted model is calculated and it is verified that the autocorrelation of the residuals is only white noise with, for example, the Ljung-Box test (see Ljung and Box, 1978). Alternatively, a higher order model can be built and it can be checked whether the additional parameters are insignificant.

The series of fitted parameters can then be used as a more complete characterization of the correlation structure of a fitness landscape. However, it is also more difficult to directly compare it with other landscapes. On the one hand, the size of the models required i.e. p and q can be a good indicator of the complexity of the correlation structure, while the parameters φ and θ can be used to directly compare the correlation structure at a more detailed level than with autocorrelation alone.

3.3.5 Information Analysis

Similar to the analysis of auto correlation, in (Vassilev et al., 2000), a different view on the correlation structure and, hence, on ruggedness is proposed. The basic idea is to analyze slopes. Any solution candidate in the search space is surrounded by a set of neighbors. It might, therefore, be

interesting to analyze the different relationships to all neighbors. While autocorrelation measures the fitness value similarity between neighbors, information analysis tries to quantify the frequency of relative orderings between neighbors from an information-theoretical standpoint. The quantitative neighborhood value comparison is explained in Section 3.5, where several ways to measure evolvability are reviewed.

A straightforward way to measure the information content of a system was discovered in (Shannon, 1948). It can be described as the uncertainty of the state of a system. The higher the information content, the more difficult a prediction becomes. Analogously, when analyzing a sequence of fitness values, one could perceive it as the inverse of compressibility. The more information is contained in this sequence the more space would be needed to compress it.

Besides the information-theoretic analysis of landscape slopes, this method includes another interesting perspective by using different quantization factors ε to view or suppress slope changes of varying magnitudes.

3.3.5.1 Fundamentals

Starting from a series of fitness values obtained, for example, from a random walk $\{f(x_i)\}_{i=0}^{n}$, the first step is to focus only on the relation between neighbors and calculate the relative fitness difference as shown in Equation 3.15.

$$\{d_i\}_{i=1}^{n} = \{f(x_i) - f(x_{i-1})\}_{i=1}^{n} \tag{3.15}$$

Subsequently, these fitness differences are quantized to either an increasing slope ' ↗ ', encoded as 1, a downward slope ' ↘ ', encoded as -1 or a straight slope ' → ', encoded as 0, using a relaxed sign function and a variable threshold ε for quantization as shown in Equation 3.16.

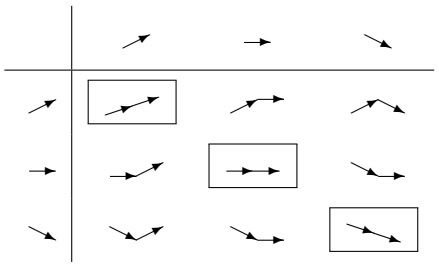
$$\widetilde{\operatorname{sgn}}_{\varepsilon}(x) := \begin{cases} x < \varepsilon & -1 & (\text{i.e. } \searrow) \\ x > \varepsilon & 1 & (\text{i.e. } \nearrow) \\ \text{otherwise} & 0 & (\text{i.e. } \rightarrow) \end{cases} \tag{3.16}$$

The resulting sequence $\{S_i\}_{i=1}^{n} = \{\widetilde{\operatorname{sgn}}_{\varepsilon}(d_i)\}_{i=1}^{n}$ is then used to construct the multiset of consecutive symbol pairs P as shown in Equation 3.17.

$$P := \left\{ [S_i S_{i+1}] \;\middle|\; i \in \{1, \ldots, n-1\} \right\} \tag{3.17}$$

This multiset of symbol pairs, which represent the slope changes going through all intermediate points of the trajectory, are then the basis of information analysis. In Table 3.2, all possible combinations of consecutive slopes are shown.

Table 3.2. Possible Slopes Analyzed within Information Analysis: The boxed slope changes are used for basin analysis while all other slopes are included in the entropy measurement of ruggedness.



3.3.5.2 Partial Information Content

One of the simplest information measures derived in this analysis is the *partial information content*. It is the count of slope changes along the trajectory or the modality of the random walk. Given the sequence of slopes $S = \{S_i\}_{i=1}^n = \{\widetilde{\text{sgn}}_\varepsilon(d_i)\}_{i=1}^n$, a new sequence is constructed using a function σ that first removes all straight slopes ' → ' and then ensures that consecutive slopes are different. In the end, a sequence with alternating slopes of the form ' ↗↘ ⋯ ↘↗ ' or ' ↘↗ ⋯ ↘↗ ' or similar is obtained. However, the sequence itself is not of prime interest but rather its relative length to the original sequence. The higher its relative length, the more rugged the trajectory has been. It should be noted that this sequence depends not only on the trajectory sample but also on the employed value of ε to categorize straight vs. non-straight consecutive steps. Therefore, the resulting *partial information content* is actually a function dependent upon this threshold as shown in Equation 3.18, where $|S|$ is abused to denote the length of sequence S.

$$M(\varepsilon) := \frac{|\sigma(S)|}{|S|} \tag{3.18}$$

3.3.5.3 Information Stability

The next measure, *information stability* is an even simpler one. It tries to capture the maximum magnitude of the information contained in the random walk. Therefore, it is defined as the smallest ε for which the resulting sequence of symbols S contains only zero. In other words, it is the maximum difference between neighboring solution candidates in the trajectory and can hence be defined as shown in Equation 3.19.

$$\varepsilon^* := \max\{f(x_i) - f(x_{i-1})\}_{i=1}^n \tag{3.19}$$

3.3.5.4 Information Content

After discussing the simplest measures, we return to the original intent of information analysis. For this, the underlying set of consecutive symbols P defined in Equation 3.17 are used. A comparable entropy measure can be found in statistical thermodynamics, where it is also referred to as "mixedupness" of a system (see Gibbs, 1906). It can be described as the number of micro states a system is composed of. The typical formula to calculate the entropy, given the number of these micro states, is shown in Equation 3.20 as the logarithmic density of states, where the different values of i represent the different states a system can be in, and P_i is the probability of being in that state i.

$$-\sum_i P_i \ln P_i \tag{3.20}$$

The entropy reaches a maximum for a certain number of states, if all states have equal probability. In our case, the information content reaches, therefore, a maximum, when all possible shapes are equally likely. On the other hand, a reduction of the information content can be observed if certain patterns prevail. Finally, the formula is directly applied to our "micro states" which are the different combinations of slopes of Table 3.2 as shown in Equation 3.21, where $p, q \in \{S_i\}_{i=1}^n$ are both slopes and $[pq] \in P$ are symbol pairs. It should be noted that both the sequence S and the set P are dependent upon the previous choice of ε. Therefore, varying the choice of ε allows a whole series for different information analyses. In Section 4.3, a simple method to select appropriate values for ε based on the distribution of relative quality differences in the sampling trajectory is described.

$$H(\varepsilon) := -\sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]} \tag{3.21}$$

As can be seen from the index of the summation in Equation 3.21, not all "micro states" of the system are considered. Instead, for the information content, only non-equal successors are considered. This gives the entropy or information content just for rugged steps in the trajectory.

3.3.5.5 Density Basin Information

The information content analysis for rugged areas is complemented by a similar entropic measure for flat areas as shown in Equation 3.22 and is called *density basin information* in (Vassilev et al., 2000).

$$h(\varepsilon) := -\sum_{p=q} P_{[pq]} \log_3 P_{[pq]} \tag{3.22}$$

In contrast to Equation 3.21, where 6 possible micro states are considered, in Equation 3.22 only 3 different micro states represent consecutive equal slopes. Therefore, the base of the logarithm differs in these cases.

3.3.5.6 Regularity

Another measure, introduced en passant in (Vassilev et al., 2000), is the degree of regularity of a sampling trajectory or, assuming isotropy, of the whole landscape. It is simply the number of distinct fitness differences between neighboring solution candidates in the sampling trajectory, or, in other words, the size of the sequence of differences after transforming it into a set as shown in Equation 3.23.

$$r(\varepsilon) := \left| \left\{ d_i \;\middle|\; d_i \in \{d_i\}_{i=1}^{n} \right\} \right| \tag{3.23}$$

3.3.5.7 Algorithm Entropy: The Information Landscape

A related concept to trajectory entropy was explored in (Borenstein and Poli, 2005b,a, 2007), where algorithmic entropy is analyzed and an *information landscape* is described. This concept is very similar to the stochastic exploration of neighborhood structures as explored in the beginning of Section 3.3.5. The information landscape is formally defined in (Borenstein and Poli, 2005a) and has been adapted to our notation here, as the triple $(\mathcal{S}, \mathcal{X}, t)$, where $\mathcal{S}$ is a set of solution candidates, $\mathcal{X}$ is a neighborhood structure and $t : \mathcal{S} \times \mathcal{S} \to [0, 1]$ is an information function that gives the probability of one solution candidate being superior than another. For many fitness landscapes, the value of $t(x_i, x_j)$ will be either 1 or 0 or 0.5, for solution candidates with better, worse and equal fitness, respectively. However, in stochastic fitness landscapes this probability can assume other values too.

An "ideal" landscape would always directly lead to the optimum, with successive fitness improvements. However, as most interesting problems also have local optima this is mostly not the case. Using the notion of an information landscape, the distance between any given landscape and the ideal landscape can be defined as the number of differences between any information landscape and its ideal counterpart. In (Borenstein and Poli, 2005a), this leads to the definition of the performance landscape which gives the number of necessary steps to reach the optimum.

This framework provides an interesting theoretical analysis method for hardness prediction of algorithm applications. However, it cannot be used for the analysis of problems with practically relevant sizes as the computational complexity is currently insurmountable as it equals an exhaustive analysis of the whole search space.

3.4 Spectral Analysis

A very interesting thread of analysis is based on the idea of auto correlation applied to the whole graph of a fitness landscape. While auto correlation is usually measured empirically for a certain sample of the solution space, another approach is to theoretically derive the average measure for the whole landscape at once.

The roots of this work in (Stadler, 1994, 1995, 1996; Stadler et al., 2000; Stadler, 2002; Reidys and Stadler, 2002) are a "Fourier Analysis" of the solution space and are also based on Weinberger's initial idea in (Weinberger, 1991) of Fourier and Taylor series of fitness landscapes. As shown in Equation 3.24, the idea is to reformulate the fitness function as a composition of several component functions φ_i with corresponding weights a_i. Ideally, these component functions have a simple structure and are orthogonal to each other. Therefore, a re-projection of the solution space to a different base is sought, similar to a Fourier decomposition as explained in e.g. (Katznelson, 2004).

$$f(x) := \sum_i a_i \varphi_i(x) \tag{3.24}$$

Moreover, when sorting these functions according to their coefficients a_i, the most significant contributing component functions can later be extracted and analyzed.

After defining the goal of the composition, the first step, however, is to find such a decomposition. This can be done with spectral analysis of the fitness landscape's graph. Using their connectivity structure, many fitness landscapes, such as for example all combinatorial optimization problems, can be described as a graph.

Instead of the adjacency matrix $\mathbf{A}$ of the fitness graph, usually, the graph Laplacian $-\Delta$, as shown in Equation 3.25, is used, where $\mathbf{I}$ is the identity matrix and d is the number of edges of each node. In this case, it is assumed that each vertex in the graph, i.e. each solution candidate, has the same number of neighbors, which holds for many practical problem classes and has nicer mathematical properties when solving for Eigenvalues.

$$-\Delta := d\mathbf{I} - \mathbf{A} \tag{3.25}$$

A cornerstone to understanding spectral landscape decomposition is the meaning of a graph's Eigenvalue decomposition as shown in Equation 3.26, where e are the eigenvectors of $-\Delta$.

$$-\Delta e = \lambda e \tag{3.26}$$

While this ansatz provides the necessary means to derive a decomposition, as required from Equation 3.24 for a Fourier decomposition, it also has interesting graph theoretic properties: Equation 3.26 can be rewritten in terms of neighborhoods as shown in Equation 3.27, where $N(x)$ is the set of all neighbors of x, i.e. all those entries in the adjacency matrix greater than zero, and $I : \mathcal{S} \to \{1, \ldots, |\mathcal{S}|\}$ is an index function that enumerates the solution space and gives the corresponding index of an element in the matrix representation.

$$\forall (x \in \mathcal{S}) \sum_{n \in N(x)} e_{I(n)} = \lambda e_{I(x)} \tag{3.27}$$

In this notation, it becomes clear what the eigenvalues and eigenvectors actually are. For an undirected regular graph, the adjacency matrix is symmetric and the eigenvectors form an orthogonal base. Therefore, we have arrived at a suitable re-projection of the solution space into *elementary landscapes* which have a very nice property: The fitness of every solution candidate can be directly calculated as the sum of the fitness values of all its neighbors as summarized in Equation 3.28, where an elementary component landscape ϕ_i is shown with its corresponding scalars ϕ_i^* and λ.

$$\Delta \phi_i = \lambda \left(\phi_i - \phi_i^* \mathbf{1} \right) \tag{3.28}$$

The difficulty of this method's application comes from the theoretical work necessary to solve eigensystems of large or infinite solution spaces. However, this method allows the decomposition of fitness landscapes into easier component landscapes with very simple properties which facilitates analysis. Several problem classes have been analyzed already in (Hordijk and Stadler, 1998; Chicano et al., 2010; Chicano and Alba, 2011). Based on these works, a general protocol for the analysis of arbitrary other problem classes has been published in (Chicano et al., 2011).

Based on this decomposition, even crossover landscapes have been analyzed using so-called P-structures in (Stadler and Wagner, 1998; Wagner and Altenberg, 1996). elaborated in Section 3.12.

3.5 Evolvability

The ability to evolve is one of the most important properties of natural systems. The most important driving force in nature, next to survival, is called *evolution*. As many metaheuristics have been inspired by evolutionary mechanics, the property of evolvability is important for both natural systems and optimization methods. In (Altenberg, 1994a), the author describes it as "the ability of a population to produce variants fitter than any yet existing" in the field of genetic programming.

In nature, evolvability has long been discussed and seems to be a countering force to natural selection. It is still not quite clear why it is such an important property for natural evolution (see Marrow, 1999). However, in (Turney, 1999) it is shown as a favorable long term property of an evolutionary algorithm. An important factor to ensure continued evolvability is through variability (see Wagner and Altenberg, 1996). Many different optimization schemes, therefore, include measure to increase diversification and, hence, variability. Some examples are the importance of diversity of a population in many recombinatory heuristics, such as random restarts in local search or the acceptance of inferior solutions in simulated annealing.

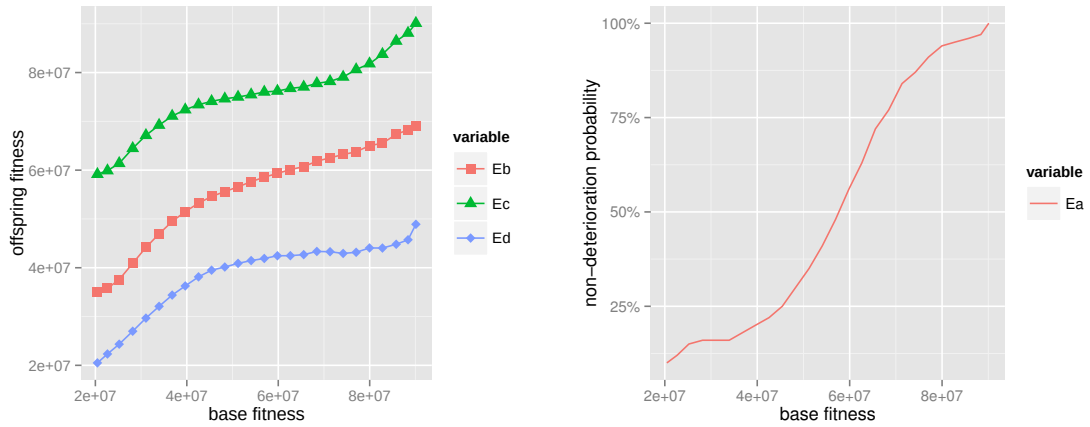
In (Grefenstette, 1999), dynamic fitness landscapes and their evolvability are discussed. There, the focus is not only on the evolution of solution candidates themselves but an evolution of the whole evolutionary machinery, analogously to biological organisms carrying and evolving their own reproductive systems, the evolution of genetic operators is explored. Therefore, this notion is not directly reusable in the generic setting of pure non-dynamic fitness landscape analysis and is not explored in further detail in this work.

In a certain sense evolvability is not a property of the fitness landscape itself but of the application of an optimization algorithm to a fitness landscape. Even though, the neighborhood structure defines the substrate for further evolutionary progress, the actual progress made by the optimization procedure lies not within the realm of fitness landscape analysis as it is tightly coupled with the process of making an improvement and, hence, a motion *on* the fitness landscape rather then a structure *in* the fitness landscape.

In the following sections several attempts have been made to analyze the "landscape portion" of evolvability or, more specifically, the neighborhood structure which can then be used by an optimization algorithm. Concrete measures of evolvability will be presented that have been proposed in the past. Finally, these efforts can be summarized into a new, more general, definition of evolvability that can serve as a basis for further analysis.

3.5.1 Evolvability Portraits

A comprehensive overview of evolvability can be obtained with a method described in (Smith et al., 2002), called evolvability portraits. In their work, evolvability is analyzed over different ranges of both ruggedness and neutrality (see Section 3.8). Figure 3.6 shows an example of a full evolvability portrait over a single landscape variant.



(a) Fitness Dependent Values

(b) Probability and Count

Figure 3.6. Evolvability Portrait: In (a), the parallel bands are the maximum, average, and minimum offspring fitness over a range of different base fitness values. In (b), the value E_a is shown that represents the probability of non-deleterious mutations at different base levels.

The underlying idea for the formulation of evolvability is the probability of finding a fitter neighbor, which is detailed in (Smith et al., 2002). Similar to the notation there, evolvability can be written either as an integral for continuous spaces as shown in Equation 3.29, or simply as a quotient and as counts of neighbors, as shown in Equation 3.30, where the integral over all non-deleterious neighbors of the probability of choosing a certain neighbor n of x is used or simply the ratio of superior neighbors in a uniform discrete space such as for many combinatorial optimization problems.

$$e(x) := \int_{f(n)\geq f(x)} \mathrm{p}(n|x)\mathrm{d}n, n \in N(x) \tag{3.29}$$

$$e(x) := \frac{|\{n \in N(x) \mid f(n) \geq f(x)\}|}{|N(x)|} \tag{3.30}$$

The probability of choosing a certain neighbor $\mathrm{p}(n|x)$ usually depends on the choice of algorithm. Therefore, for a general analysis, a uniform distribution is often appropriate. On the other hand, different choices for $\mathrm{p}(n|x)$ can provide different perspectives. As the total probability of choosing any neighbor $\int \mathrm{p}(n|x)\mathrm{d}n = 1$, the integral in Equation 3.29 is a probability density function.

Based on this quantification of evolvability, several different measures have been proposed in (Smith et al., 2002). These measures are initially defined for a single solution candidate x. However, they can then be evaluated for different partitions of the solution space to better understand its inherent structure.

The *evolvability portraits* proposed in (Smith et al., 2002) consist of the following measures:

- $E_a(x) := e(x)$, or simply the probability of a non-deleterious mutation.
- $E_b(x) := \int \mathrm{p}(n|x)f(n)\mathrm{d}n$, which is the average, or expected offspring or mutation fitness.
- $E_c(x) := \frac{100}{C} \int_{f(n)\geq f_c} \mathrm{p}(n|x)f(n)\mathrm{d}n$, where f_c is defined implicitly through $\int_{f(n)\geq f_c} \mathrm{p}(n|x)\mathrm{d}n = \frac{C}{100}$, or in other words, the top C-th percentile of the offspring fitness.
- $E_d(x) := \frac{100}{D} \int_{f(n)\leq f_d} \mathrm{p}(n|x)f(n)\mathrm{d}n$, where f_d is defined analogously to f_c to give the bottom D-th percentile of the expected mutation fitness.

These measures provide a nice summary of important aspects of the evolvability of a single solution candidate. In the original paper, these measures have then been collected for different base fitness values, where solution candidates with equal or similar fitness are grouped together and the averages of these four values are calculated. An example of this is shown in Figure 3.6.

From these charts, the mutation strength of different operators can be directly compared over different ranges of base fitness. However, the general trend is often the same: For very bad solution candidates, larger mutations have a higher chance of bringing them to average quality, while more conservative operators have a higher chance of not destroying good solutions further on. However, close to the optimum, or in a local optimum, less conservative operators have a higher chance of escaping again, and, therefore, have a higher chance of fitter offspring.

3.5.2 The Fitness Cloud

Another way of visualizing evolvability was introduced in (Collard et al., 2004). There, a simple scatter plot containing base fitness f and *bordering fitness* $\widetilde{f}$ (f' in the plot), which is the fitness of a neighboring solution candidate according to some neighborhood and assumed to be unique.

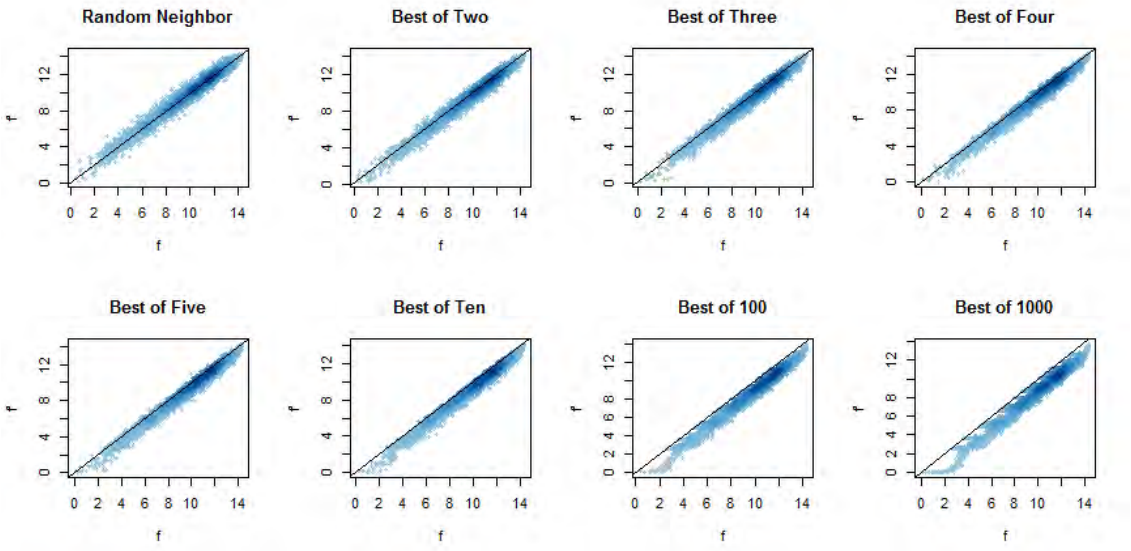


Figure 3.7. Fitness Clouds of Different Neighborhoods of Ackley's Function: As the quality of the neighbors increases, more insights into the underlying structure of this function become apparent. (Compare to Figure 3.1b for Ackley's function itself.)

In Figure 3.7, several examples of different fitness clouds are shown. These are all for the same fitness function, albeit for different fitness landscapes as the neighborhood relation has been changed according to a "best selection" scheme with increasing sample size. As discussed in Section 2.2.5 it remains a matter of taste whether these landscapes constitute something fundamentally different. However, gauging from the plot, different properties become more or less apparent in the different variants: As the number of neighbors increases, the path towards local optima can be revealed as seen in the last plot with 1000 samples.

3.5.2.1 Interpretation

To explore the shape of a fitness cloud, in (Collard et al., 2004), several subsets of the overall cloud are defined as shown in Equation 3.31.

$$FC_{\min} := \left\{ (\phi, \widetilde{\phi}) \;\middle|\; \phi \in f(S), \widetilde{\phi} = \min \widetilde{f}(x) \right\} \tag{3.31}$$

$$FC_{\text{mean}} := \left\{ (\phi, \widetilde{\phi}) \;\middle|\; \phi \in f(S), \widetilde{\phi} = \text{mean } \widetilde{f}(x) \right\} \tag{3.32}$$

$$FC_{\max} := \left\{ (\phi, \widetilde{\phi}) \;\middle|\; \phi \in f(S), \widetilde{\phi} = \max \widetilde{f}(x) \right\} \tag{3.33}$$

According to (Collard et al., 2004), the intersections of these boundaries with the diagonal ($f = \widetilde{f}$) mark important fitness levels which are denoted as α, β and γ for $FC_{\min}$, FC_{mean} and $FC_{\max}$ respectively.

- Base fitness values below α will always have better bordering fitness, and are, therefore, called *strictly advantageous*.
- Values between α and β will be on average *advantageous* as their bordering fitness is higher than their own.
- Values between β and γ, on the other hand, are *deleterious*, on average as their bordering fitness is lower.
- Finally, values above γ are *strictly deleterious* as their bordering fitness is always lower than their own.

All points at the top (or at the bottom for minimization problems) are local optima, as they have no neighbors with a bordering fitness which is higher (or lower). The fitness cloud allows a relatively easily understandable visualization of the neighborhood fitness structure. The fitness level β is assumed to be a "barrier of fitness" in (Collard et al., 2004) as it describes a level that is easier to fall below than overcome. It should be noted, that, as with all evolvability analysis methods, this method is typically specific to a certain neighbor selection scheme as already shown in Figure 3.7. However, it is a worthwhile addition to our fitness landscape analysis tool belt.

3.5.2.2 Negative Slope Coefficient

An extension to the fitness cloud introduced in (Collard et al., 2004) is described in (Vanneschi et al., 2004), the *negative slope coefficient*. Fitness pairs (f, g) of the fitness cloud are divided into groups based on their first value f, or in other words, the fitness cloud is sliced vertically into groups. For each of these groups, the average fitness and average offspring fitness are calculated, denoted as $\overline{f_i}$, and $\overline{g_i}$. The resulting line, which can be described as the spine of the fitness cloud, is then further analyzed. In particular the slopes S_i between consecutive points are calculated as shown in Equation 3.34.

$$S_i = \frac{g_{i+1} - g_i}{f_{i+1} - f_i} \tag{3.34}$$

Finally, the negative slope coefficient is defined as the sum over all negative slopes as shown in Equation 3.35.

$$nsc = \sum_i \min(S_i, 0) \tag{3.35}$$

The negative slope coefficient can be thought of as a broad overview of the fitness landscape, where misleading structures can be identified. Whenever the optimization process has to go downward on average to reach a higher fitness (or the other way around for a minimization problem) this can hinder the optimization process as is could become stuck. This idea is captured nicely in the negative slope coefficient, where all such downward slopes are summed up to represent the

difficulty of a problem. It should be noted, that this method measures just one type of difficulty that can occur, therefore, it should be used in combination with other measures to obtain a more complete picture.

One of the weaknesses of this method, however, is that the split points for the slices are not standardized and can actually have a large impact on the usefulness or predictive power, which was pointed out in (Vanneschi et al., 2006). They also noted, that a more elaborate segmentation scheme would improve the performance

It should be further noted, that this method lacks a normalizing term that would make it comparable across different problem instances. While easy problems can be characterized as problems with zero or low negative slope coefficient, depending on the scale and size of the fitness landscape itself, the number and choice of segments as well as the extent of fitness values, the concrete numerical value of the negative slope coefficient can vary greatly.

The negative slope coefficient has been originally developed for genetic programming. However, its generality makes it applicable to any fitness landscape as it requires only the direct components of a fitness landscape.

Later in (Vanneschi et al., 2007) and (Vanneschi et al., 2009) this method has been used to investigate neutrality and was applied to NK landscapes (see Section 3.14.3), where it was successful in correlating with increased ruggedness and, hence, increased problem difficulty of the NK landscapes. In addition, they confirmed that an analysis of only a small sample of an NK landscape showed comparable results to a full analysis. This fact is probably explainable by the high isotropy of NK landscapes for which many other measures will probably also show a high degree of generalizability over smaller samples.

3.5.2.3 Fitness Correlation Analysis

A very similar concept to the evolvability analysis methods introduced in the previous sections was introduced in (Brandt and Dieckmann, 1999). However, in that study it is called "correlation analysis" between classes of solution candidates with different base fitness. By analyzing the convergence probabilities from one class to the next, they gain insights into the fitness landscape that are very similar to the evolvability analysis of the previous sections. Despite the name, it is less similar to typical correlation analysis, like the autocorrelation that analyzes the similarity of fitness values regardless of the underlying base fitness than to evolvability analysis described here.

3.5.3 Generalized Evolvability

Combining all ideas from the literature, a simple general definition of evolvability can be derived that can serve as a base for new analysis techniques.

The underlying structure for performing different evolvability analyses formally is now characterized. This can be used to easily derive new measures related to evolvability. First, $\phi_R(w)$ is defined in Equation 3.36 as the set of all solution candidates in the sample $R \subseteq \mathcal{S}$ that have a fitness of w.

$$\phi_R(w) := \{x \mid x \in R, f(x) = w\} \tag{3.36}$$

Evolvability $\mathcal{E}(x, \delta)$ is defined for a single solution candidate x to increase fitness by δ as shown in Equation 3.37. Now, the evolvability can be calculated for different samples or different regions with equal fitness, as shown in Equation 3.38.

$$\mathcal{E}(x, \delta) := \int_{f(n)=f(x)+\delta} p(n|x) \, \mathrm{d}n \tag{3.37}$$

$$\mathcal{E}_R(w, \delta) := \frac{1}{|\phi_R(w)|} \sum_{x \in \phi_R(w)} \mathcal{E}(x, \delta) \tag{3.38}$$

The function $\mathcal{E}_R(w, \delta)$ can be used as the basic tool for a whole array of different evolvability analyses by choosing different samples, looking at bins of similar fitness improvements or bins of parent fitness values. If the most probable δ for a certain fitness value w are examined, a similar analysis as described in Section 3.5.1 can be obtained. Alternatively, the most probable combinations of w and δ can be plotted to arrive at an analysis similar to Section 3.5.2. Evolvability analysis is yet another viewpoint of the fitness landscape that enables a better understanding of problem difficulty and suitability of a certain heuristic.

3.6 Global Versus Local Analysis

In the previous sections, the focus was on local analysis methods that successively explore the neighborhood structure of a fitness landscape. These methods are intended to yield information about the progression of trajectory-based optimization methods as these methods "perceive" the landscape from a similar perspective. Once other optimization methods are employed that have a different notion of "progress" such as variable neighborhood search (VNS) (see Hansen et al., 2010), or population based methods such as evolution strategies (ES) (see Rechenberg, 1973) or genetic algorithms (see Holland, 1975), other fundamentally different views become increasingly important.

In the following sections, measures that focus more on a larger scale of the fitness landscape are introduced. However, sometimes, the distinction between global and local is not clear cut as some of the methods will analyze local properties over a wider area or in different contexts. While there is no clear better view of fitness landscapes there are some advantages and disadvantages of the different scopes: Global methods tend to be more computationally expensive as more care has to be taken for sample selection and evaluation. On the other hand, global methods also tend to reveal results more readily, while local methods need more effort for interpretation to arrive at the big picture.

3.7 Fitness Distance Correlation

In contrast to other trajectory based methods, evolvability analysis takes on a more global view. In the analysis of *fitness distance correlation* (FDC) as introduced in (Jones, 1995) and discussed in detail in (Jones and Forrest, 1995), another take on more global aspects is made. In this case, knowledge of the global optimum is also a prerequisite. While this makes the analysis slightly less general as an exhaustive landscape analysis is required beforehand to find the global optimum, it can provide a very good overview of the landscape structure. As most optimization methods are striving to find the global optimum, a detailed analysis of the whole landscape with this respect can be revealing, however, this advantage is at the same time the greatest disadvantage as the global optima have to be obtained beforehand.

The basic idea of fitness distance correlation (FDC) is to analyze the correlation between the *fitness difference* between the global optimum and any point in the landscape and the *location difference*, or in other words the *distance*, between the global optimum and any point in the landscape. Therefore, it is named fitness distance correlation. Two versions of this analysis have been presented in (Jones, 1995) and (Jones and Forrest, 1995):

- On the one hand, a plot of fitness and distance pairs can be created. Let $\widehat{o} \in \mathcal{O}(\mathcal{F})$ a global optimum of $\mathcal{F}$, then a plot of the set in Equation 3.39 will show the correspondence between

the difference in the distance domain and in the fitness domain.

$$\mathrm{FDS}(\mathcal{F}) := \left\{(d_i, f_i) \;\middle|\; x \in \mathcal{S}, d_i = d(x, \widehat{o}), f_i = |f(x) - f(\widehat{o})|)\right\} \tag{3.39}$$

- On the other hand, the correlation coefficient between these two distributions can also be calculated to obtain a single number that represents one aspect of problem difficulty. Using the set of Equation 3.39, the correlation coefficient can be calculated, as shown in Equation 3.40, where $\mathrm{E}[x]$ and $\mathrm{Var}[x]$ are the expected value and variance of x, respectively.

$$\mathrm{FDC}(\mathcal{F}) := \frac{\mathrm{E}[f_i \cdot d_i] - E[f_i] \cdot E[d_i]}{\mathrm{Var}[f_i] \cdot \mathrm{Var}[d_i]} \tag{3.40}$$

The FDC has been used in many studies already. Especially in situations, where a deeper problem understanding is important, it can provide help in the categorization of problem instances (Fonlupt et al., 1997; Naudts, 1998; Merz, 2004; Wang et al., 2006; Czech, 2008; Tavares et al., 2008; Ochoa et al., 2009; Uludağ and Uyar, 2009; Le et al., 2009).

While the FDC plot as a whole can nicely reveal the relationship between fitness and distance distributions, the mere correlation coefficient alone, as a single numerical result, bears some problems. As with any linear correlation coefficient, it is only able to capture linear relationships. While, indeed, an optimization landscape should ideally show some correlation of fitness and distance to be able to lead an optimization algorithm towards the optimum, this is neither necessary nor sufficient for an optimization scheme to be successful. One the one hand, a very narrow path alone could be correlated, while the rest of the landscape is arbitrarily distributed. In this case, the problem could be quite easy for some algorithms but still show no correlation. On the other hand, the general tendency of most solution candidates could have a good correlation with fitness and distance while in the direct neighborhood of the global optimum the trend is reversed, which could lead to the conclusion of an easy problem that actually requires some extra effort to be solved to optimality. This property is already discussed in the original works in (Jones, 1995; Jones and Forrest, 1995). Later on, several concrete example cases were documented in (Naudts and Kallel, 2000). Probably, the simplification of a large multi dimensional relationship to a single number is too much of a simplification. Nevertheless, the underlying distribution has great potential for further exploration and the correlation coefficient contributes an important insight that still works for many problem classes. Again, it captures one distinct aspect of a fitness landscape and should be combined with other measures that provide a different view.

In Figure 3.8, several examples of fitness distance correlations are shown. First, the area of Ackley's function (see Ackley, 1987) is shown. Here, the structure surrounding the global optimum can almost be seen. When this is compared to Figure 3.1b, the prominent "bumps" close to the global optimum can be clearly seen in the FDC plot. Rastrigin's test function (see Törn and Zilinskas, 1989; H. Mühlenbein and Born, 1991), shown in Equation 3.41 is somewhat similar to Ackley's function which can be seen from the third FDC plot in Figure 3.8. Other examples, included in the figure are Rosenbrock's function (see Rosenbrock, 1960) and Beale's function (see Moré et al., 1981). Finally, the last plot in Figure 3.8 shows a multi normal test function which is introduced in (Pitzer et al., 2010), where the deceptive properties of this function are clearly visible and large basins can lead the optimization procedures into the shallow optima.

$$f(x) := 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right) \qquad \text{where } x \in [-5.12, 5.12]^n \tag{3.41}$$

Also combinatorial problems can be analyzed with the help of fitness distance correlation. Figure 3.9 shows a small instance of the traveling salesman problem library (TSPLIB) (see Reinelt, 1991, and Section 3.14.1.1), where the nature of combinatorial optimization problems can be observed. While there is a correlation between fitness and distance at the extreme values, most

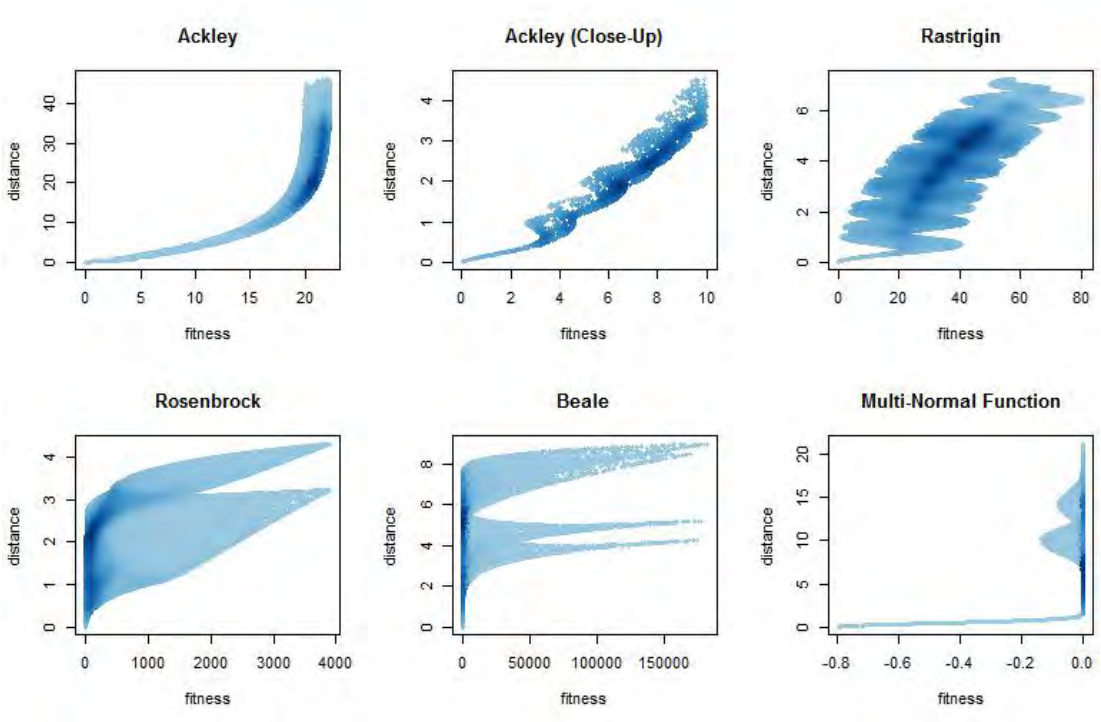


Figure 3.8. Several Examples of Fitness Distances Correlations

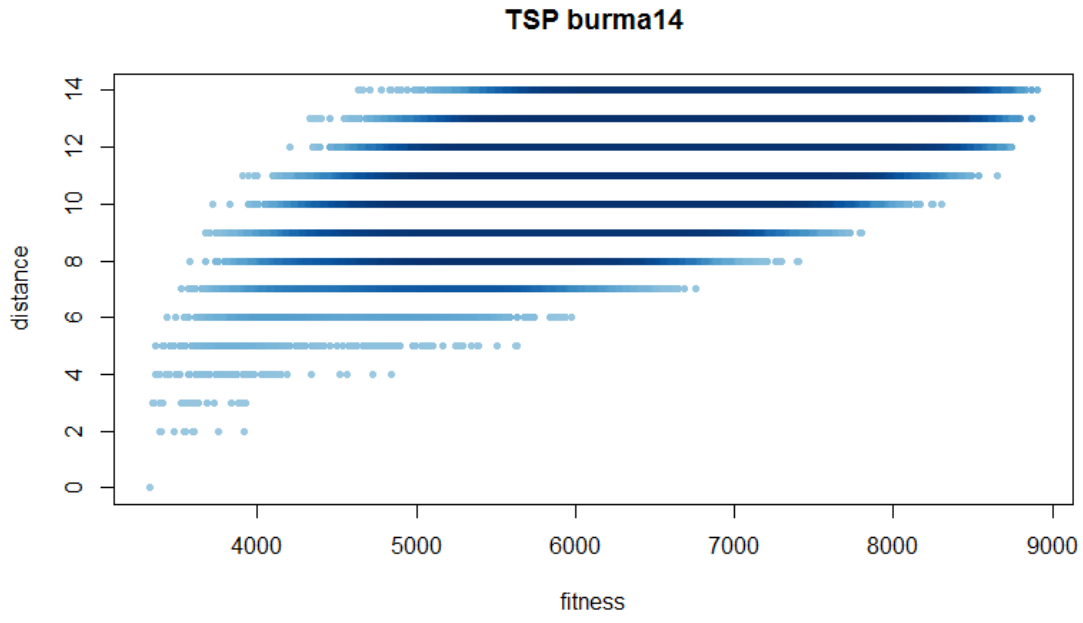


Figure 3.9. Fitness Distance Correlation of Traveling Salesmen Problem "burma14"

points in the solution space are evenly distributed. As a final note, the weak correlation could also be due to the employed distance calculation. While the optimization methods will usually use a swap operator that exchanges two cities in the permutation, the distances in Figure 3.9 have been calculated with a simpler distance function as the actual swap distance is computationally very expensive to compute.

The sampling points for the plots in Figures 3.8 and 3.9 have been obtained using an *optimized sample* as described in Sections 3.2.2 and 4.2.5.1 so that areas with higher fitness can be included more easily.

While the fitness distance correlation marks an important cornerstone in problem understanding it has two important limitations. On the one hand, knowledge of the global optimum can prevent its application for new problem instances, and for a predictive algorithm selection it is therefore of little use. On the other hand, the information compression into a single correlation coefficient, while successful in some cases (see Jones, 1995; Merz, 2004; Tavares et al., 2008), has also been found to be misleading in others (see Altenberg, 1997; Quick et al., 1998; Kallel et al., 1999).

3.8 Neutrality

Neutrality has already been discussed as a very basic property in Section 2.2.2. It surfaced first, however, in the discussion of natural evolution (see Kimura, 1968), where it caused strong debate as to whether Darwinian selection or neutrality was the prevalent force in evolution. While neutral evolution allows the exploration of a much greater gene pool, Darwinian selection allows much faster adaption.

After the introduction of the notion of nearly neutral mutations in (Ohta, 1992, 1996, 1998), it was accepted as an important factor in the success of natural evolution. While slightly disruptive mutations can have negligible effects on current fitness, they can provide a great potential for future mutations, when the environment has changed and provide increased diversity. Finally, in (Wagner, 2008) a balance between the two mechanisms of improving selection and nearly neutral mutations was proposed, where a neutral mutation does not *currently* have an effect but might have later on. The main problem to arrive at this conclusion was the difficulty of measuring fitness and, hence, of measuring neutrality (Wagner, 2005).

In (Huynen et al., 1996), high-dimensional spaces are characterized as neutral networks penetrating each other, so that any solution candidates can be reached quickly from any point in the solution space by using neutral networks as "tunnels" throughout the landscape. The relevance of neutral areas in high-dimensional spaces is also discussed in (Provine, 1989), where it is argued that it is extremely unlikely that a solution candidate is optimal in every single dimension, and hence, local optima may not even exist there anymore. Also in (Gavrilets, 1999), the importance of neutrality for the proliferation of natural evolution is emphasized: Without neutral mutations and redundancy in the encoding, intermediate steps in evolution would likely lead to non-functional proteins which would create individuals that are not able to survive.

The formal definition of a neutral area or neutral network was already given in Section 2.2.2. Another variation, as already informally introduced in the previous paragraphs are *almost* neutral areas. In this case instead of a strict equality of solution candidates small variations between neighbors or inside the whole connected set can be allowed:

- This can be done either be relaxing the neighboring fitness constraint to a certain maximum difference ε as shown in Equation 3.42.

$$(\forall y \in N(x) \cap \widetilde{S_n}) \; |f(x) - f(y)| < \varepsilon \tag{3.42}$$

- Alternatively, the fitness difference between all members of the neutral area can be confined into a certain ε-band as shown in Equation 3.43.

$$\max_{x,y \in \widetilde{S_n}} |x - y| \leq \varepsilon \tag{3.43}$$

Inside a neutral area an optimization algorithm can become stuck or "confused" as there is seemingly no progress to make. In this situation it is important to be able to perform *constant innovation* as described in (Huynen, 1996) and (Smith et al., 2002), where an algorithm further explores the neutral area without becoming stuck.

As described in Section 3.2.1.4, one possibility to measure and characterize neutral areas are neutral walks. The distance between entry and exit point of a neutral walk, when the walk continues diametrically from the entry point to the exit point, can be a good indicator of the landscape's neutrality. Other measures for neutrality are proposed in (Katada et al., 2004; Katada, 2006) using the Nei standard genetic distance (see Nei, 1972). Finally, in (Barnett, 1997) and (Barnett, 1998), neutrality in fitness landscapes is characterized and several measures are defined:

- Volume or number of solution candidates inside neutral areas.
- The number of neighboring fitness values of the neutral area.
- The maximum distance between two members of a neutral area in (Reidys and Stadler, 1998).

It has been noted that neutrality has a very profound impact on the success of many algorithms (see Barnett, 1997, 1998; Reidys and Stadler, 1998; Vassilev et al., 2000). Also, for the popular family of NK landscapes, described in Section 3.14.3 special variants, the NKp and NKq landscapes, have been created that artificially introduce neutrality to model and measure this important property.

Again, neutrality is another characteristic property of fitness landscapes and provides a complementary measure for the analysis and characterization that has proven to be quite insightful also in our own tests as shown in Chapter 4.

3.8.1 Neutrality and Large Neighborhoods

As the number of dimensions increases, the possibility of neutrality also increases, as it becomes more and more likely to find at least one dimension, where an encoding changes but the fitness remains the same. The same is true for large neighborhoods, where, as the number of possible neighbors rises, the chance of finding a neighbor with equal fitness increases. This is, again, of particular interest to natural evolution (see Huynen et al., 1996; Huynen, 1996) but can have interesting effects on the measurement of ruggedness: As the size of the problem class or the size of the neighborhood increases, even for problem classes with known low neutrality, it becomes more and more likely to observe neutrality simply by chance.

3.8.2 Quasispecies and Near Equilibrium

In natural evolution, neutrality was discovered as an important mechanism to enable phenotype coherence alongside genetic diversity. This is the concept of a quasispecies, where a group of individuals, despite coherent appearance and behavior do not consist of identical incarnations of the same genotype. This is obviously beneficial for evolution, especially with respect to a changing environment (see Wright, 1932).

Therefore, many quasispecies remain in a state of near equilibrium for a long time, where all genotypes are centered around a few very fit master sequences. However, in a changing environment as well as through discovery of new beneficial random mutations these quasispecies can evolve.

Moreover, through the higher genetic diversity of the population, chances of discovering such a beneficial mutation are much higher than that of a species with a stationary genotype.

3.8.3 Artificial Neutrality

Also for metaheuristics, neutrality was discovered as an important concept. One issue is the detection and escape of neutral areas to ensure further progress, as described in the paragraph on constant innovation in the previous section. On the other hand, the beneficial aspects of quasispecies have been exploited by specially designed metaheuristics or deliberately redundant encodings such as messy genetic algorithms in (Goldberg et al., 1993, 1989) to enlarge the *surface area* of mutation and create a potentially more robust scheme of evolutionary progress.

Along the same lines, offspring selection (see Affenzeller and Wagner, 2005; Affenzeller et al., 2009), can be seen to create artificial neutrality when individuals are re-used to create further offspring from the same population and, hence explore the "surrounding" individuals more thoroughly than direct crossover would.

3.8.4 Percolation

Another important term that is strongly connected to neutrality is percolation (see Kesten, 2006). The neutral networks or neutral areas of a fitness landscape can be seen as its pores that allow contention-free traversal by an appropriate algorithm. If the surrounding solution candidates are explored, one can ask the question of how likely it is to "trickle through" or percolate to the optimum. Therefore, the percolation index in (Barnett, 1997) of a fitness landscape can be a measure which can give an estimate of the amount of penetration of neutral networks inside a fitness landscape.

3.9 Epistasis

A very interesting concept of fitness landscapes is *epistasis*. Conceptually, it is the interaction of different "components" of the solution candidate. The terminology stems actually from biology, where it denotes the effects of gene interference as described in (Bateson, 1909). Even 100 years later, as mentioned in (Cordell, 2002) it is still difficult to exactly define this phenomenon even in a biological sense. Even more difficulties have been observed when it was tried to measure epistasis in fitness landscapes. Several methods have been proposed in the past, however, all of them have been heavily criticized.

Some simple biological examples of epistasis are the expression of one gene inhibiting the functioning of another gene such as the *albino* gene which prevents the expression of any color pigments. Therefore, the allele, or "value", of the color gene itself, which would, for example, decide between brown or black fur is irrelevant.

To study epistasis, a fitness function is first defined in terms of its *components*. A linear fitness function without any epistatic interactions could be reformulated as a shown in Equation 3.44, where a solution candidate $x \in \mathcal{S}$ is decomposed into its "components" $x = \{x_i\}_{i=1}^{n}$ and the fitness function is decomposed analogously.

$$f(x) := \sum_{i=0}^{n} f_i(x_i) \tag{3.44}$$

As epistasis increases, the fitness function does not only contain linear function components, such as $f_1(x_1)$ based on a single component, but more and more epistatic components, such as $f_{1,2,3}(x_1, x_2, x_3)$.

3.9.1 Epistasis Variance

A first attempt to measure epistasis was proposed in (Davidor, 1990). It is based on the assumption that the fitness function can be modeled as a linear function if it contains no epistasis and, hence, any resulting error that cannot be explained by the linear function must be due to epistasis. This was called *epistasis variance* as, actually, the variance between actual fitness function and linear approximation were compared.

In detail, first a linear model $\widetilde{f}(x) = \sum f_i(x_i)$ using parameter learning has to be estimated. Then, the difference of variance of the original fitness function and the linear approximation is compared as shown in 3.45, where $\mathrm{Var}[f(x)]$ denotes the variance of $f(x)$ over the sample points x.

$$\mathrm{EV}(f) = \frac{\mathrm{Var}[f(x)] - \mathrm{Var}[\widetilde{f}(x)]}{\mathrm{Var}[f(x)]} \tag{3.45}$$

In some cases, this method can be used to measure epistasis, but it actually only measures the absence of epistasis as shown in (Naudts, 1998) and has been heavily criticized in (Reeves and Wright, 1995). However, in (Naudts, 1998) and (Naudts and Verschoren, 1999), it is used successfully for the prediction of problem hardness and correlated to deceptiveness.

3.9.2 Graded Epistasis Variance and Epistasis Correlation

One severe shortcoming of epistasis variance is the lack of distinction between different orders of epistasis. While certainly a linear model is easiest, there should arguably be a difference between a fitness function that houses interactions of just two components at most, such as $f_{1,2}(x_1, x_2)$, or more such as $f_{1,2,3,4}(x_1, x_2, x_3, x_4)$. This has been noted already during the original introduction in (Davidor, 1990).

A first attempt to include more orders of epistatic interaction is proposed in (Rochet, 1997), where more orders of non-linearity are modeled. This approach is called *graded* epistasis variance. Moreover, in this work, the use of direct correlation between original and simplified fitness function is proposed instead of the comparison of variances.

This approach mitigates the problem of different orders of epistasis but introduces the additional problem that many more parameters have to be approximated. In fact, the number of possible interactions increases exponentially with the considered order of epistasis.

3.9.3 Criticisms and Perspectives

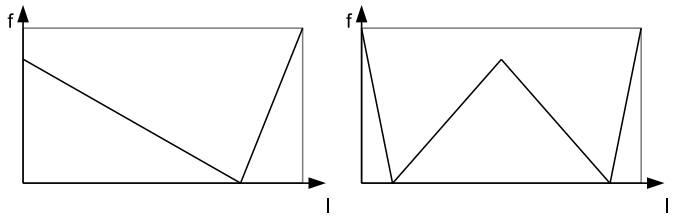
The concept of epistasis is very important and it would be nice to have a reliable measurement. In (Rochet et al., 1998), an interesting experiment is tried to deceive the epistasis measurements: A different encoding is used which reduces the measurable epistasis, however, the problem does not become easier to solve. As pointed out in (Naudts, 1998; Naudts and Kallel, 2000), all measures, actually measure the absence of epistasis and not epistasis itself. And in (Reeves and Rowe, 2003) an elaborate collection of arguments against these measurements can be found. Nevertheless, these measurements, even though unable to measure full epistasis can provide a valuable insight into the problem structure. The only drawback for practical purposes is the relatively high computational effort.

In summary, epistasis is a fascinating concept that is and should be used conceptually. The contribution of several, albeit relatively superficial, measurements of epistasis are valuable despite their high computation effort and should be kept at hand for a comprehensive problem instance characterization.

3.10 Deceptiveness

The term of deceptiveness is more of a conceptual description of a fitness landscape than a real measure as it requires human interpretation and knowledge about not only the search space itself but also the type of algorithm that will be used to optimize it. In fact, deceptive landscapes are specially designed fitness landscapes with no real application other than to be hard for a particular class of algorithms, such as, for example, the famous trap function in (Deb and Goldberg, 1993).

The term was introduced in (Goldberg, 1987) and has often been used to describe the difficulty of functions for genetic algorithms. Figure 3.10 shows two popular examples which are defined as functions, where the attractor and the global optimum have maximum distance to each other (see Deb et al., 1993). These function are defined for a binary encoding, the primary encoding for theoretical analyses of genetic algorithms and therefore, the values along the axes have to be seen as only partially ordered, according to their number of ones and zeros.



(a) Single Optimum **(b)** Two Optima

Figure 3.10. Examples of Simple Trap Functions: The number of ones in the binary representation increases from left to right, therefore, the axis is only partially ordered.

In (Liepins and Vose, 1991) a collection of deceptive and easy problems most notably the fully deceptive problem as described in (Deb and Goldberg, 1994) is given. However, as the usefulness of deceptiveness itself, especially outside the realm of genetic algorithms has been questioned in (Mitchell and Forrest, 1991; Grefenstette, 1992), no concrete and general formulation is available that can easily be applied to the general formulation of a fitness landscape using just solution space, fitness and neighborhood.

3.11 Isotropy

The word isotropy comes from Greek “isos” meaning equal and “tropos” for manner, style or way. In other words “equal in every way”. Different disciplines have slightly different interpretations. For fitness landscape analysis, it is the property of a landscape to “look the same” everywhere. In other words, if a landscape is fully isotropic, any local analysis will be the same no matter in what direction or which starting point is used. This assumption is often made to justify the application of “the powerful results of the theory of stationary random processes to the study of random walks on fitness landscapes” (see Weinberger, 1990) or other actually local analyses that are only generalizable under the assumption of isotropy.

There are several levels of isotropy, the most restrictive one is implying complete independence of direction and location for any kind of analysis. Obviously, this is generally too restrictive. Therefore, it is mostly applied to local correlation measures (see Stadler and Happel, 1999). The weaker definition of isotropy is satisfied by the popular NK landscapes (see Section 3.14.3), however, many real-world examples are not completely isotropic (see Vassilev et al., 2000; Czech, 2008). Therefore, care should be taken when generalizing results of local analysis to ensure isotropy or use more sophisticated methods to deal with anisotropy instead.

Up to now, only few studies have tried to quantify isotropy and have attached little value to the meaning of isotropy in practical fitness landscapes. One attempt in (Stadler and Grüner, 1993) was to exhaustively enumerate the free energy landscape of RNA, where significant anisotropy was discovered in a real-world practice-oriented problem.

3.11.1 Formal Definition in Previous Works

While conceptually, we can talk about an isotropic landscape, there is no way of directly measuring it. That might be the reason, isotropy has received little attention from the field of fitness landscape analysis in the past. However, looking at the previous sections, plenty of measurements are available that *can* be used on fitness landscapes. Most of these measures are defined over a single large sample, representative of the whole landscapes. As isotropy is the "measure" of homogeneity of this sample, repeated measurements of different parts of the landscape can be taken and compared to each other.

This first formulation of isotropy can be found in (Stadler and Grüner, 1993) and has been reproduced in original notation in Equation 3.46, where $\langle f(x,y)\rangle_d^A$ is the average of $f(x,y)$ over all pairs x, y of solution candidates with a distance of d and $A \subset \mathcal{S}$ is a connected subgraph of the solution space $\mathcal{S}$. Furthermore, A has to be an element of a partition $\mathcal{B}$ of the solution space. This requirement was added to remove any bias to particular areas of the fitness landscape and to obtain an even distribution of sample regions. In (Stadler and Grüner, 1993) it is shown that Equation 3.46 has to hold if a landscape is *empirically isotropic*.

$$\left\langle (f(x) - f(y))^2 \right\rangle_d^A \cong \left\langle (f(x) - f(y))^2 \right\rangle_d \tag{3.46}$$

Moreover, also Equation 3.47 has to hold for empirically isotropic landscapes. The notation has been slightly adapted to the notation used in this work, so that $\mathcal{B}$ is, again, a partition of the fitness landscape, f and $\mathcal{S}$ are the fitness function and the solution space, respectively and $\overline{\rho}_{\mathcal{B}}$ is the average correlation between all pairs of the partition $\mathcal{B}$.

$$\mathrm{var}_A[f] \cong \mathrm{var}_{\mathcal{S}}[f](1 - \overline{\rho}_{\mathcal{B}}) \tag{3.47}$$

Finally, from Equation 3.47, the *coefficient of anisotropy* is derived in (Stadler and Grüner, 1993), as shown in Equation 3.48 and captures the amount to which isotropy is violated. Here, the variance of the average in each individual partition is divided by the overall variance minus the average correlation. Or even simpler, the relative variance of the average fitness in different areas of the solution space.

$$\alpha = \frac{\mathrm{var}_{\mathcal{B}}[\langle f \rangle_A]}{\mathrm{var}_{\mathcal{S}}[f]} - \overline{\rho}_{\mathcal{B}} \tag{3.48}$$

Finally, it is noted that this notion of isotropy pertains to the choice of the partition. Therefore, for certain choices of the partition $\mathcal{B}$ a landscape $\mathcal{F}$ can have different levels of measured anisotropy.

Similar to this idea, a sampling-based algorithm was developed in this work that analyses the distribution of different base measures over the landscape as a very practical way to infer and quantify isotropy in a fitness landscape. The details are developed in Section 4.2.

3.12 Crossover Analysis

Another area of fitness landscape analysis that has received little attention is the analysis of more complex neighborhood structures such as crossover. While, of course, within the realm of these optimization strategies, crossover, and recombination have been studied as demonstrated in

e.g. (Holland, 1975, 1988; Syswerda, 1989), most of the time, this happens within the framework of a concrete algorithm and a concrete encoding and not in a general setting such as fitness landscape analysis, where all we have is the solution space, the fitness function and a neighborhood structure.

3.12.1 Optimum Relinking

A first shot at crossover analysis is introduced in (Merz, 2004), where NK landscapes and the binary quadratic programming problem (see Beasley, 1998; Wegener and Witt, 2005) are analyzed with respect to the applicability of memetic algorithms (see Moscato, 1989; Moscato and Cotta, 1991; Moscato, 1999).

The idea is to compare two trajectories, both starting at a local optimum. The first trajectory leads from one local optimum to another local optimum, while the other trajectory constitutes a random walk. By comparing whether worthwhile intermediate solutions lie between two optima it can be examined whether crossover would be beneficial for the optimization of a particular problem instance. If intermediate points provide no advantage compared to points reached by a random walk, trajectory based optimization methods should be sufficient. On the other hand, this test will only work if the trajectory between local optima actually describes points that are reachable by the employed crossover operator as a crossover operator does not necessarily have to produce offspring that are "between" their parents. However, it can be assumed that the offspring of a crossover operator is similar to both parents.

3.12.2 Spectral Crossover Analysis: P-Structures

An interesting formulation has been presented as an extension to elementary landscape analysis (see Section 3.4) over a crossover neighborhood, where the neighborhood structure has been expanded to a hypergraph in (Stadler and Wagner, 1998). In this particular case, edges are not drawn from one vertex to another but from two vertexes together to a third. Figure 3.11 shows an example of a very simple hypergraph.

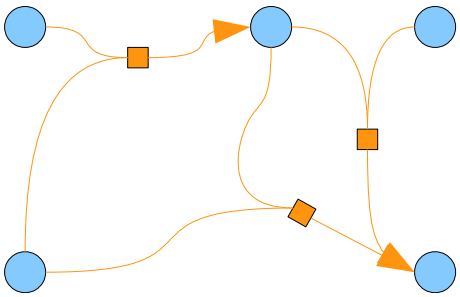


Figure 3.11. Example of a Directed Hypergraph: In contrast to a conventional directed graph, every edge has more than a single start vertex and a single end vertex. In this case the edges can be seen themselves as vertexes, shown as small boxes, therefore, a hypergraph is a special case of a bipartite graph with one vertex set representing the vertexes and the other one representing the edges.

Many landscapes of combinatorial optimization problems are elementary or easily decomposable into a small set of elementary landscapes (see Hordijk, 1996) and crossover landscapes have been successfully analyzed with this method in (Stadler and Wagner, 1998; Wagner and Stadler, 1998) with the help of so-called *P-structures*: A graph is defined through an incidence matrix $\mathbf{H}$ over the crossover "neighborhood" in Equation 3.49, where $R : \mathcal{S} \times \mathcal{S} \to \mathcal{P}(\mathcal{S})$ maps a pair of parents

into a set of possible offspring ($\mathcal{P}(x)$ is the power set of x).

$$\mathbf{H}_{x,(y,z)} = \begin{cases} 1 & \text{if } x \in R(y,z) \\ 0 & \text{otherwise} \end{cases} \tag{3.49}$$

This matrix is then used to define an $|\mathcal{S}| \times |\mathcal{S}|$ square matrix as given in Equation 3.50 that can then be used to derive a graph Laplacian analogous to Equation 3.25 in Section 3.4.

$$\mathbf{S}_{xy} = \frac{1}{|V|} \sum_{z \in \mathcal{S}} \mathbf{H}_{x,(y,z)} \eta(y,z)^{-1} \tag{3.50}$$

where $|V|$ is the number of vertexes and $\eta(y,z)$ is the number of different possible offspring of the parents y and z as given in Equation 3.51.

$$\eta(y,z) = |R(y,z)| = \sum_{x \in \mathcal{S}} \mathbf{H}_{x,(y,z)} \tag{3.51}$$

However, in this approach, mating is simulated with a uniform distribution of a second parent which is usually not the case in a population-based heuristic. Specifically, the second parent is usually drawn from a similar distribution than the first parent and, therefore, has a high chance of having similar fitness. It is more like the "headless chicken" test proposed in (Jones, 1995), which is significantly different to how crossover actually works in most algorithms, where it creates directed macro-mutation, whereas in this approach, only undirected macro-mutation is examined.

3.13 Problem-Specific Analysis

Finally, as a complement to fitness landscape analysis and a look outside the "box" to alternative and competing methods, two examples of problem-specific analysis methods are briefly mentioned here. This section should only give an idea of the many possible theoretical analysis results that have been preformed in the past and are not directly within the scope of this work. However, from our own experience, they pose an important complement to pure fitness landscape analysis and demarcate their borders.

Once, we can look deeper into the problem itself, by allowing specific knowledge into the components of the fitness function and the neighborhood structure many more possibilities become available. Here, Two examples are given for problem-specific measures of the quadratic assignment problem (QAP) which is explained in detail in Section 3.14.1.3.

In short, a quadratic assignment problem consists of two matrices, the flows and distances between facilities and locations. These two matrices, which are not directly accessible in fitness landscape analysis, can now be subjected to direct, problem-specific analysis.

The first, and probably best known, example are the so-called *flow dominance* and *distance dominance* introduced in (Vollmann and Buffa, 1699) and later extended in (Knowles and Corne, 2002). Equations 3.52 and 3.53 show the flow and distance dominance, respectively, for the weight matrix $\mathbf{F}$ and distance matrix $\mathbf{D}$, where $\mathrm{Var}[x]$ and $\mathrm{E}[x]$ are the variance and expected value of x, respectively.

$$\mathrm{dom}(\mathbf{F}) := 100 \cdot \frac{\mathrm{Var}[\mathbf{F}]}{\mathrm{E}[\mathbf{F}]} \tag{3.52}$$

$$\mathrm{dom}(\mathbf{D}) := 100 \cdot \frac{\mathrm{Var}[\mathbf{D}]}{\mathrm{E}[\mathbf{D}]} \tag{3.53}$$

Therefore, these dominance measures describe the normalized variance or variability of the problem-constituent matrices. These measures have been used for the determination of problem hardness characterizations in (Merz and Freisleben, 2000).

Another interesting class of problem-dependent measures has been described in (de Abreu et al., 2002), where theoretical deduction of the fitness function mean and variance are based on the flow and distance matrices as shown in Equations 3.54 and 3.55. The mean fitness of a quadratic assignment problem is relatively easy to deduce (Angel and Zissimopoulos, 1998b), where it is shown that when iterating over all possible permutation each product occurs exactly the same number of times, and, therefore, can be reduced to the simple expression shown in Equation 3.54, where $c(\mathbf{F}, \mathbf{D})$ is the cost function using the flow matrix $\mathbf{F}$ and the distance matrix $\mathbf{D}$.

$$\mathrm{E}[c(\mathbf{F}, \mathbf{D})] = \frac{1}{2n(n-1)} \sum_{ij} f_{ij} \sum_{ij} d_{ij} \tag{3.54}$$

To obtain the variance of the overall fitness of a quadratic assignment problem, a little more work is needed. However, in (de Abreu et al., 2002), a relatively straightforward expression is obtained as shown in Equation 3.55.

$$\begin{aligned}
\mathrm{Var}[c(\mathbf{F}, \mathbf{D})] &:= [S_0 + S_1 + S_2]/n! - \mu^2 \\
S_0 &= 4(n-4)! \sum_{\cap_0} f_{ij} f_{rs} \sum_{\cap_0} d_{ij} d_{rs} \\
S_1 &= (n-3)! \sum_{\cap_1} f_{ij} f_{rs} \sum_{\cap_1} d_{ij} d_{rs} \\
S_2 &= 2(n-2)! \sum_{\cap_2} f_{ij} f_{rs} \sum_{\cap_2} d_{ij} d_{rs} \\
\cap_k &= \left\{ ((i,j),(r,s)) \;\middle|\; 1 \leq i,j,r,s \leq n, |\{(i,j) \cap (r,s)\}| = k \right\} \\
\{(i,j) \cap (r,s)\} &= \{i,j\} \cap \{r,s\}
\end{aligned} \tag{3.55}$$

Here, the variance is decomposed into three different cases, based on the number of coinciding indexes, accessed via the index duplication set, which is defined as $\cap_k$ (note, the diagonal is not considered): First, all products with non-duplicate indexes are calculated and multiplied with the correct number of occurrence in S_0, next, all products with one overlapping index are summed up and finally, all same-index products are considered. This calculation can be implemented efficiently by iterating over all pairs of products once and consecutively adding to either S_0, S_1, or S_2 depending on the overlap of indexes.

Finally, these measures can be used to predict the problem hardness as is shown in Section 4.7. In the original paper in (de Abreu et al., 2002), however, the flow dominance itself is used and divided by the problem size to give a problem hardness estimate.

In the past, problem specific measures have also been examined for the vehicle routing problem which are described in (Pitzer et al., 2012b) and are elaborated in Section 4.3.

These two examples could just scratch the surface of the vast area of problem-specific analysis methods. As the concrete problem formulation and encoding become tangible, many more leverage points are possible and, therefore, make room for many more possibilities of analysis than the general but fundamental formulation of fitness landscapes has available.

3.14 Test Landscapes and Previous Results

Some problem classes have been analyzed using fitness landscape or problem specific analysis methods in the past. This section provides a few examples. Most prominently, the NK landscapes have been created as an artificial test problem for the investigation of fitness landscape analysis methods.

3.14.1 Combinatorial Optimization Problems

A very prominent example of problems that are solved with heuristic methods are combinatorial optimization problems. While real-valued problems often are susceptible to exact analytic or

approximative numeric methods, interesting combinatorial optimization problems are often NP-complete and leave little choice other than stochastic optimization. In the following, a short introduction is given about the problem classes that have been examined using fitness landscape analysis, either in this thesis or in previous works.

3.14.1.1 The Traveling Salesman Problem

The traveling salesman problem (TSP) (Lawler et al., 1985) is a popular combinatorial optimization problem. It is relatively easy to describe but NP-complete, as shown in (Karp, 1972). It is often used as a benchmark problem for heuristic algorithms. One of the reasons might be that exact methods are available that are relatively efficient, such as branch-and-bound or branch-and-cut methods. Therefore, for many problem instances global optima are known and provide a good basis for the comparison of metaheuristics. On the other hand, because of the efficiency of exact algorithms it is actually not such a good example for the application of metaheuristics.

Formally, the TSP is defined as finding the minimum length of a cycle that visits all nodes in a fully connected graph. Very often in combinatorial optimization, the path or the solution candidate is described as a permutation π of cities which defines the order in which they are visited. Moreover, an $n \times n$ distance matrix $\mathbf{D} = (d_{ij})$ gives the mutual distances between all pairs of the n cities. Equation 3.56 shows the simple fitness function of the TSP, which is the sum over all distances d_{ij} between the cities in the order defined by the permutation.

$$f(\pi) = d_{\pi(n)\pi(1)} + \sum_{i=1}^{n} d_{\pi(i)\pi(i+1)} \tag{3.56}$$

In the past, because of its popularity in metaheuristics, it has been formally analyzed also using fitness landscape approaches:

- Using problem-dependent measures together with a correlation analysis, in (Mathias and Whitley, 1992) it was tried to predict operator performances on the TSP.
- Another ruggedness analysis in (Stadler and Schnabl, 1992) discovered that TSP instances from real city maps are smooth and that ruggedness increases with problem size.
- In (Fonlupt et al., 1997), different landscape variants were compared and it was found that operators which make only small changes, such as a 2-opt, are the smoothest and, therefore, the easiest for optimization.
- Later, in (Merz and Freisleben, 2001), a very interesting discovery about the structure of the TSP landscape was made. They postulated that the global optimum is surrounded by other local optima which are all concentrated in a certain region of the search space, forming a mountainous region similar to the "massif central" in France.
- In (Reidys and Stadler, 2002), the correspondence between optimization algorithm and mutation operator is analyzed and it is shown that depending on the algorithm, different mutation operators will have better performances. This is similar to the correspondence between encoding and mutation operators as already discussed in Section 2.1.4.

Finally, it should be noted that many instance have been collected into a central repository, the traveling salesman problem library (TSPLIB) as described in (Reinelt, 1991) which can serve as base for large scale analysis of diverse problem instances.

3.14.1.2 The Vehicle Routing Problem

The vehicle routing problem (VRP) has been introduced in (Dantzig and Ramser, 1959) and describes a problem somewhat similar to the traveling salesman problem albeit with a central

depot. Typically, also some form of capacity or time constraints are added which make this problem more complex than the traveling salesman problem.

A simple formulation can be found in (Laporte, 1992), where it is defined as follows: The VRP consists of a graph $G = (V, A)$ with vertexes $V = \{1, \ldots, n\}$ representing the cities and a set of arcs A connecting the vertexes. The central depot is at vertex 1. The distances between the cities along each arc are contained in an asymmetric distances matrix $\mathbf{C} = (c_{ij})$. Moreover, m identical vehicles are available each with a capacity of D. The goal of the VRP is then to find minimal cost tours such that:

- All cities in $V \backslash \{1\}$ are visited exactly once.
- All tours start and end at the depot.
- Some constraints are satisfied. Popular constraints are:
  - A limit of capacities of the vehicles.
  - A limit on the number of stops a vehicle can visit.
  - A limit on the total time for each vehicle.
  - Certain cities must be visited at certain times.

Many different extensions have been proposed as summarized in (Eksioglu et al., 2009) and (Golden et al., 2008) as well as many methods for solving it (see Cordeau et al., 2005).

A first analysis of the fitness landscape was given in (Jones, 1995), where it has been subjected to a fitness distance correlation analysis and a big valley structure of the local optima was postulated which could not be confirmed in (Czech, 2008). Some preliminary analysis of the vehicle routing problem were performed in (Pitzer et al., 2012b) which have been extended in this work.

Also for this problem class, several collections of instances are available. The instances described in (Augerat et al., 1995), (Christofides and Eilon, 1969), (Fisher, 1994) and (Christofides et al., 1979) were used in the experiments in this work as they are freely available on the web[1].

3.14.1.3 The Quadratic Assignment Problem

Another popular combinatorial optimization problem is the quadratic assignment problem. It has been studied extensively in Chapter 4 and used as a testbed for many of the fitness landscape analysis methods.

The quadratic assignment problem (QAP) is best imagined as the assignment of facilities to locations, each of which have a corresponding coherence structure. The locations have certain distances between each other, encoded in a distance matrix $\mathbf{D} = (d_{ij})$ while the facilities have associated flows between each other, encoded as flow matrix $\mathbf{F} = (f_{ij})$. It is then the objective to minimize the overall product of flows and distances as shown in Equation 3.57, where π is a permutation that assigns a location to each facility.

$$f(\pi) := \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}\, f_{\pi(i)\pi(j)} \tag{3.57}$$

Given this definition, the number of locations and facilities has to be the same. For problems with unequal number, however, it is possible to introduce dummy locations with very high distance or dummy facilities with zero flows.

In the past, many instances of the quadratic assignment problem have been analyzed (see Pitzer et al., 2013a,b, 2012a; Beham et al., 2011), which are detailed in Chapter 4. A detailed analysis

[1] `http://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm`

of the quadratic assignment problem can be found in (Merz and Freisleben, 2000), where also a combination of many measures is used to characterize the quadratic assignment problem with respect to hardness for memetic algorithms. In summary, they split the problem instances into several classes with different difficulties based on fitness landscape analysis results.

In (Rockmore et al., 2002), the quadratic assignment problem is decomposed into elementary landscapes. Similar work has been done in (Chicano et al., 2010), moreover using this techniques a direct method for the calculation of the exact auto correlation has been introduced in (Chicano et al., 2011).

Again, also for this popular problem class a comprehensive collection of problem instances is freely available in the quadartic assignment problem library (QAPLIB) which is described in (Burkard et al., 1997) and was used as the primary source for test instance for the implementation of fitness landscape analysis methods in this work.

3.14.2 Random Fields

In (Reidys and Stadler, 1998) additive random fields are described as a very general class of optimizaiton problems with several variations that have been frequently studied using fitness landscape analysis methods. The basic definition, shown in Equation 3.58, is very simple and consists of several fitness component functions $f_i(x)$ each with a different weight c_i. In contrast to the component functions in the analysis of epistasis in Section 3.9, however, each of the components has access to all components of the solution candidate x. Therefore, random fields have an enormous potential for containing epistatic interactions.

$$f(x) = \sum_i c_i f_i(x) \tag{3.58}$$

3.14.2.1 Spin Glasses

A relatively simple incarnation of random fields are the famous spin glasses introduced in (Sherrington and Kirkpatrick, 1975; Derrida, 1980), which model the behavior of adjacent magnets. Each magnet interacts with the spins of its neighbors either positively or negatively. In this case, all fitness component functions are similar and connect the spins of those components that are adjacent. These systems have many *metastable* states as a physicist would call them. In other words, these systems have many local optima and are slightly simpler than the NK landscapes, which are introduced in the next section.

3.14.3 NK Landscapes

NK landscapes are the preferred playground test landscapes for many fitness landscape analysis methods. They were introduced in (Kauffman, 1989) and discussed further in (Kauffman, 1993). This model is closely related to the idea of binary DNA strings with epistatic interactions.

The name NK derives from the two parameter of the landscape. While 'N' denotes the number of elements in the binary vector, 'K' stands for the number of interactions of each gene. In essence, the NK landscapes are the prime model for directly epistatic landscapes.

There are two popular neighborhood models that define the positions of gene interactions. While in the *adjacent neighborhood* model, only adjacent genes are allowed in the interaction while the *random neighborhood* model allows interactions with any gene.

In summary, NK landscapes can be formulated as shown in Equation 3.59, where every component function operates on the corresponding gene x_i and a set of k neighboring genes. The function

$n_{i,j}(x)$, therefore, extracts the appropriate components of x which is different for every component function f_i. Therefore, every component function f_i operates actually on $k+1$ genes, x_i itself and a certain selection of k other genes.

$$f(x) := \frac{1}{n} \sum_i c_i f_i(x_i, n_{i,1}(x), n_{i,2}(x), \ldots, n_{i,k}(x)) \tag{3.59}$$

As a simple example, the neighborhood selection functions for the two adjacent bits neighborhood is shown in Equation 3.60.

$$\begin{array}{rcl} n_{i,1}(x) & := & x_{(i-1) \bmod N} \\ n_{i,2}(x) & := & x_{(i+1) \bmod N} \end{array} \tag{3.60}$$

To illustrate the complexity of this simple model, an example is shown in Table 3.3. Here an NK landscape with $N = 3$ and $K = 1$ is defined. In addition, only the first neighborhood definition of Equation 3.60 is used. For every combination of input values and component functions, one value has to be defined. This is typically done at random in an NK landscape as shown in Table 3.3.

Table 3.3. f_i's for an NK landscape with $N = 3$ and $K = 1$.

x_i	x_j	$f_1(x_i, x_j)$	$f_2(x_i, x_j)$	$f_3(x_i, x_j)$
0	0	.3	.7	.2
0	1	.8	.2	.3
1	0	.0	.7	.1
1	1	.1	.2	.2

Given these two definitions, now the solution candidate $x = 010$ is evaluated, as shown in Equation 3.61. First, it has to be determined which solution candidate components are inserted into each of the component functions, next the component functions are evaluated and finally they are averaged.

$$\begin{array}{rcl} f(010) & = & [f_1(x_1, x_3) + f_2(x_2, x_1) + f_3(x_3, x_2)]\,/3 \\ & = & [f_1(0,0) + f_2(1,0) + f_3(0,1)]/3 \\ & = & (0.3 + 0.7 + 0.3)/3 \\ & = & 1.3/3 \end{array} \tag{3.61}$$

Even in this simple example, NK landscapes have a very high epistatic interaction between genes, which can easily become very *frustrating* for any optimization scheme. Many practical optimization problems on the other hand exhibit much more structure.

By increasing the parameter K, the number of epistatic interactions can be controlled and hence the level of frustration adjusted. As the parameter K approaches N, the landscape becomes more and more random until at $K = N-1$, the table contains a unique entry for every possible solution candidate (see Jones, 1995).

Because of these extreme properties, NK landscapes have been studied extensively in the past (see Weinberger, 1990; Hordijk, 1996; Smith et al., 2002; Vassilev et al., 2000; Merz, 2004) very often not because of the interest in the structure of NK landscapes themselves but as a testbed for new fitness landscape analysis methods.

3.14.3.1 NK Landscape Variants with Artificial Neutrality

As mentioned in Section 3.8, NK landscapes have been augmented to artificially introduce neutrality. Again to create a testbed for analysis methods that measure neutrality.

In the first proposal in (Barnett, 1998), an additional parameter p is added giving rise to NKp landscapes with tunable neutrality. This neutrality is achieved by setting any entry in the random table with values for the component functions to zero with probability p. The higher this

probability, the more values in the table will become zero and, therefore, the higher the chance to generate equal values in the resulting function which also causes an increased chance of equal adjacent values which in turn causes an increase in neutrality.

A different approach is presented in (Newman and Engelhardt, 1998), where a similar effect is achieved through an additional parameter q and, hence, the birth of NKq landscapes. This time, the parameter q determines the number of distinct values allowed in the component function table. As fewer values are allowed for the component functions, fewer combined values are possible. Therefore, similar to the NKp landscapes, the chance of equal adjacent values increases, which gives again a higher chance of neutrality.

In (Geared et al., 2002), these two variants of NK landscapes with neutrality have been compared. While the effects of both variants are very similar, through the different approaches the characteristics are quite different. Therefore, both methods produce interesting new landscape variants with different properties and provide further playground for testing of fitness landscape analysis methods.

3.14.3.2 Previous Analysis Results

In many cases the NK landscapes or its variants have been used as first testbed for new analysis algorithms (see Hordijk, 1996; Vassilev et al., 2000; Smith et al., 2002; Merz, 2004; Vanneschi et al., 2009), which have been described already in previous sections. Moreover, as NK landscapes have been designed with natural evolution in mind in (Kauffman, 1989) it was confirmed in (Aita et al., 2007) that they are, indeed, a good model for molecular evolution.

The characteristics of NK landscapes as determined in (Kauffman, 1993) and (Fontana et al., 1993) are summarized in (Altenberg, 1997) as follows:

- For the minimum complexity case $K = 0$, the optimization is easy as there is only a single global optimum to which, on average, $N/2$ steps are necessary. Moreover, the landscape is very smooth and highly correlated.
- For low values of K, the best local optima have many alleles in common, which is not too surprising since many combinations of component functions can be re-used for other local optima as they are not destroyed by epistatic interactions.
- For maximum complexity with $K = N-1$ there are, on average, $(2^N)/(N+1)$ local optima. On the other hand, the average fitness value of these optima decreases to almost the mean fitness of the whole landscape. This is called "complexity catastrophe" in (Kauffman, 1989).

These results were obtained through an adaptive trajectory analysis. In (Altenberg, 1997), also a population-based analysis is conducted, where it is found that recombination makes no difference for high values of K when compared to trajectory-based methods. However, he finds support for Wright's argument that on landscapes with higher ruggedness, population subdivision or increased diversity plays a more important role.

The results of (Thompson and Wright, 1996) and (Weinberger, 1996) are also summarized in (Altenberg, 1997) to give some bounds for the complexity of the NK problem:

- NK problems with adjacent neighborhood are solvable in $O(2^K N)$ (Weinberger, 1996).
- NK problems with random neighborhood with $K = 2$ are NP complete (Thompson and Wright, 1996).
- NK problems with random neighborhood are NP complete for $K > 2$ (Weinberger, 1996).

As a side note, there is no difference for NK problems with $K = 1$ between random neighborhood and adjacent neighborhood. Similar results are already found in (Stadler and Happel, 1995), where

it is shown that random neighborhoods are more complex than adjacent neighborhoods as more possibilities of epistasis are possible.

In (Barnett, 1998), the properties of NKp landscapes are examined. Neutrality is found primarily in areas with lower fitness which is not very surprising since, neutrality is introduced by setting random values in the component function tables to zero. An interesting finding is that the parameter p which tunes neutrality has almost no effect on the ruggedness of the NKp landscapes. Moreover, NKp landscapes enable constant innovation along a neutral walk similar to an RNA folding landscape as studied in (Huynen, 1996).

Finally, it should be noted, that NK landscapes are much more difficult that typical other problem classes of the same size as their difficulty is close to a completely random landscape (see Borenstein and Poli, 2007).

3.14.3.3 Isotropy of NK Landscapes

In (Weinberger, 1996) and (Stadler, 1994), NK landscapes are used as examples of isotropic landscapes. Due to the typically unbiased random initialization and the even distribution of interactions, it is shown that NK landscapes are isotropic. In fact, NK landscapes are one of very few examples of perfectly isotropic landscapes. Many practical problem classes such as the TSP or the QAP exhibit some form of structure throughout the solution space. In Section 4.2.7, an attempt is made to artificially introduce anisotropy in NK landscapes as a new test landscape for isotropy measures.

3.14.3.4 Generalized NK Landscapes: The Block Model

The slightly peculiar formulation shown in Section 3.14.3, where every gene x_i always has to be part of component function f_i is relatively arbitrary and has been abandoned in (Altenberg, 1997) for the formulation of a more general and simpler block model. In this model, a component function has access to an arbitrary number of solution candidate components and the overall function consists again of an arbitrary number of fitness component functions as shown in Equation 3.62, where k_i is the number of solution candidate components in the component function f_i and $n_{i,j}(x)$ denotes the j-th component of the i-th component function, while the overall number of components is given with n.

$$f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(n_{i,1}(x), n_{i,2}(x), \ldots, n_{i,k_i}(x)) \tag{3.62}$$

Figure 3.12 shows an example of such a generalized block model, that can be used to model NK landscapes or other variants of additive random fields. The components of the input vector x are shown at the top, while the component fitness functions f_i are in the middle row. Finally all values are averaged as the result of f. In Figure 3.12 the component functions have been color coded and the respective solution components reflect this coloring to show the extent and target of epistatic interactions.

3.14.3.5 Implementation of NK Landscapes

For the implementation of an NK landscape, the naive approach is to randomly build a table with fitness values for each combination of inputs values and for each component function. The number of positions is shown in Equation 3.63, where n is the number of component function and $|f_i|$

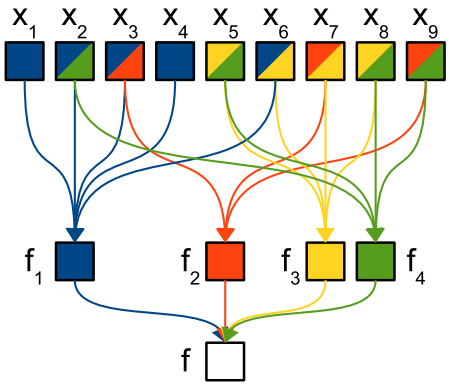


Figure 3.12. Generalized Block Model: This formulation can be used to model any of NK landscapes, spin glasses or more general, highly epistatic fitness functions.

denotes the arity, or the number arguments of f_i and m is the number of possible values of each fitness component, which is usually two in the case of binary vectors.

$$\sum_{i=1}^{n} m^{|f_i|} \tag{3.63}$$

As can be seen from this calculation, the space requirements for an NK function with higher values of K are enormous. A simple function over a binary string with $N = 100$ and $K = 50$, for example, would require $100 \cdot 2^{51} \approx 2 \cdot 10^{17}$ random table entries. Using 32 bit floating point numbers this would require more than 900 petabytes of memory and the function initialization alone would probably take quite some time.

For this reason, a clever alternative has been proposed in (Altenberg, 1994b), where the table is not created up-front but table entries are repeatedly generated as needed using a pseudo-random number generator. Equation 3.64 shows the evaluation of a single component function using a hash function h instead of a pseudo-random number generator, where $\oplus$ denotes the "bitwise exclusive or" operation and $\wedge$ is reused to denote "bitwise binary and". Moreover, g_i is the binary component interaction vector that selects which bits participate in the calculation of component function f_i and s is a random seed that determines which concrete values will be used for this incarnation of the NK landscape.

$$\begin{aligned} f_i(x) \quad = \quad & 2^{-W} \cdot h((x \wedge g_i) \oplus \\ & \quad h(g_i \oplus \\ & \quad\quad h(i \oplus s))) \end{aligned} \tag{3.64}$$

While this approach uses slightly more computation time during evaluation, it saves a lot of space and time during initialization and is much more practical as for typical experiments only a few thousand or million evaluations are made, which is actually less than the values in the initialization table.

3.14.3.6 Summary

Additive random fields are not only useful "toy" landscapes and serve as testbeds for optimization and analysis methods. Rather, they serve as a direct model landscape with tangible epistasis and have been related to several practical problems such as RNA folding (see Barnett, 1998) or DNA to protein affinity landscape (see Rowe et al., 2010).

Chapter 4

New Analysis Methods

A comprehensive review of existing theory and techniques for fitness landscape analysis is given in Chapter 3. As can be seen from the various attempts, fitness landscape analysis methods usually yield simple measurements that describe different characteristics of a problem instance. While some of these measures have been created to directly elucidate problem hardness, most of them provide only a narrow view of the characteristics. In other words, these measurements can all be seen to provide different perspectives as they are measuring different properties of a problem instance.

The first contribution of this thesis is to dig deeper into the foundations of fitness landscape analysis and explore simple properties that have not yet been explored to their full potential.

The second contribution is to create new complementary measures that try to capture characteristics that can be understood conceptually, but have not yet been measured and add them to the arsenal of different perspectives. In particular, we set out to measure the extent of the often-assumed isotropy of many practical problem classes. On the one hand, isotropy assumptions should be validated for certain problem classes and, on the other hand, their implications to optimization difficulty should be investigated. Moreover, isotropy measures contribute another characteristic of problem instances that can be added to obtain a clearer overall picture.

The third contribution of this thesis is to use the full power of previous methods and, therefore, combine them and metaphorically get a *3D full-color view* of the fitness landscape instead of the one-dimensional gray-scale images that are provided by singular measures. This comprehensive characterization of a fitness landscape can then be used to identify similar problems by comparing their characteristics, and by inferring optimization strategies based on similarity to and existing expertise in similar problems. This leads to the final goal of this thesis, to test fitness landscape analysis methods for their true generality across different problem classes, and achieves first steps towards automatic algorithm selection and parameterization.

Ideally, these efforts should lead to a guiding system, or even an automatic optimization system, that circumnavigates the No Free Lunch Theorem by providing what would usually be done by an experienced researcher of optimization methods: Finding similar but known problems that have been subjected to optimization in the past and select appropriate algorithms and parameter settings to start with.

4.1 Basin Analysis

An informal definition of basins of attraction was given in Section 2.2.3 which will be extend in this section. In a previous study (see Pitzer et al., 2010), basins of attraction were examined in

great detail and interesting properties of this seemingly simple construct were discovered. Some results of this section are similar to results previously published in (Pitzer et al., 2010).

4.1.1 Definition

As mentioned in (Stadler, 2002), a problem that becomes apparent mostly in higher dimensional spaces and with nearby optima with adjacent basins of attraction is that the assignment of solution candidates to basins of attraction is dependent on the search strategy. The notion of converging trajectories is used here to define a basin of attraction. Therefore, the formalization is started with the definition of a downward path, as shown in Equation 4.1.

$$\begin{array}{lll} \widetilde{p\downarrow}([x,y]) & :\Leftrightarrow & f(x) \geq f(y) \wedge N(x,y) \\ \widetilde{p\downarrow}([x,h|t]) & :\Leftrightarrow & f(x) \geq f(h) \wedge N(x,h) \wedge \widetilde{p\downarrow}([h|t]) \end{array} \tag{4.1}$$

In this case, discrete steps are used to explore the downward path. However, it can also be applied to continuous neighborhood connectivities, where an ε-neighborhood N_ε can be used instead of a discrete neighborhood. Moreover, a similar constructive definition as for connected sets, shown in Section 2.2.2, can be used to define a downward path $p\downarrow$ in a solution space. The notation follows the formalism used in logic programming, so a sequence of two elements x and y is denoted as $[x, y]$ and a *destructured* sequence, with head h and tail t, is denoted as $[h|t]$. Moreover, the neighborhood itself is used as a directed relation $N(x, y)$, as shown in Equation 4.2, as a shorthand for denoting that y is a neighbor of x, instead of the previous set notation. In many practical neighborhoods, this relation will be symmetric, for our considerations of trajectories, symmetry is irrelevant. Even for non-reversible manipulation operators, trajectories and basins can be defined. For these cases, the direction of the neighborhood relation in Equation 4.2 becomes relevant.

$$N(x,y) :\Leftrightarrow y \in N(x) \tag{4.2}$$

To enforce an actual downward movement, the definition in Equation 4.3 of a strict downward path is used, where at least one step in the path has to have a downward change in fitness value.

$$p\downarrow(S) :\Leftrightarrow \widetilde{p\downarrow}(S) \wedge \exists(i < j)\ f(S_i) > f(S_j) \tag{4.3}$$

At this point it should be noted that a basin of attraction is defined as an actual basin, or in other words as part of a minimization problem in which an optimum will be a minimum without loss of generality.

For simplification, the binary predicate $p\downarrow(x, y)$ is defined as a shorthand for the existence of any strict downward path from x down to y in Equation 4.4

$$p\downarrow(x,y) :\Leftrightarrow \exists(S = [x,\ldots,y])\ p\downarrow(S) \tag{4.4}$$

This gives us a way to define a local optimum in terms of downward paths as shown in Equation 4.5. In particular, any solution candidate with no downward path to another solution candidate is a local optimum. This matches exactly with the definition of a local optimum given in Section 2.2.1 and gives us a very simple definition of a generalized local optimum that could either be a singular point or part of a plateau as defined in Section 2.2.2.

$$o \in \mathcal{O} \Leftrightarrow \nexists(x \in \mathcal{S})\ p\downarrow(o,x) \tag{4.5}$$

In this section, however, basins of attraction are formalized. In fact, two different scenarios, or types of basins, can immediately be identified when the contour plot of Figure 2.2 in Section 2.2.3 is examined. On the one hand, we have what we will call a *strong* basin of attraction that immediately surrounds a local optimum and promises convergence to that optimum. On the other

hand, these strong basins are surrounded with weak basins of attraction that, while still attracting search trajectories, are not unconditionally attractive towards a single optimum.

It is easier to start with the definition of a weak basin as shown in Equation 4.6. A weak basin is actually a superset of a strong basin. For every optimum $o \in \mathcal{O}$, in fact even for any point $p \in \mathcal{S}$, its weak basin $\widetilde{b}(o)$ can be defined by subsuming all elements of a solution space that have a downward path $p{\downarrow}(x, o)$ to this optimum or point.

$$\widetilde{b}(o) := \{x \mid x \in \mathcal{S}, p{\downarrow}(x, o)\} \tag{4.6}$$

To arrive at the definition of a strong basin, those parts of a weak basin have to be removed that have ambiguous convergence, i.e. those solution candidates from which convergence to other optima is possible. For this definition, knowledge of all optima $\mathcal{O}$ and their weak basins is required.

$$b(o) := \left\{ x \;\middle|\; x \in \widetilde{b}(o), \nexists(o_2 \in \mathcal{O}, o_2 \neq o)\ x \in \widetilde{b}(o_2) \right\} \tag{4.7}$$

While convergence inside an optimum's strong basin certainly happens towards that particular optimum, convergence inside any overlapping weak basins can have non-zero convergence probabilities to other optima. These probabilities can easily be defined recursively as shown in Equation 4.8. The probability of converging from one point x to an arbitrary optimum o is the weighted sum of convergence probabilities over its neighbors $N(x)$. For every neighbor, two different aspects are considered:

- On the one hand, the probability of choosing that particular neighbor n when currently at x, denoted as $p(n|x)$,
- which is then multiplied by the probability of this particular neighbor n itself converging to the optimum o.

Finally, for the boundary cases at the optima themselves $p(o, o) = 1$, or in other words, the probability of converging to an optimum o at the optimum o itself is 100%.

$$p(x, o) := \sum_{n \in N(x)} p(n|x) \cdot p(n, o) \tag{4.8}$$

4.1.2 The Basin-Fill Algorithm

Examining Equation 4.8 seems to be the end of the story for practical analysis. On the one hand, knowledge of all optima is required, which would still be feasible for small solution spaces, but the analysis would require exploration of the whole solution space from every solution candidate to every other, which would further limit practical application. Fortunately, many of the selection probabilities are zero for many trajectories. This allows us to apply dynamic programming for solving Equation 4.8 quite efficiently without even obtaining all local optima first.

The underlying idea comes from the landscape metaphor itself. When calculating probabilities of Equation 4.8, usually all probabilities leading upward will be zero. Therefore, it just needs to be ensured that convergence probabilities for solution candidates with lower values are calculated first. While it still requires an exhaustive enumeration of the solution space this is feasible at least for smaller problems. Even for continuous and relatively smooth spaces, this analysis can be performed.

To start with, the whole solution space is sorted according to fitness values. This can be done in $\mathrm{O}(n \log n)$ time and constant space e.g. with a Heapsort algorithm (see Williams, 1964; Knuth, 1997). From here on, it is quite simple to start calculation of the recursive formula. If the calculation starts with the lowest values, there will be no downward paths and once higher values are reached, all lower fitness values and most importantly all convergence probabilities of lower

fitness solution candidates will already be available. This can be imagined as a porous landscape which is slowly submerged in water. As the solution candidates are submerged, their corresponding convergence probabilities are calculated using only convergence probabilities of solution candidates that are already under water.

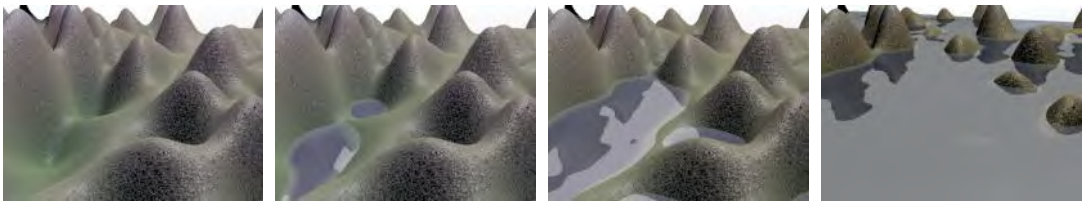


Figure 4.1. Subsequently "Submerging" the Landscape in Water: All downward paths are already under water and, therefore, have been calculated. This means the that algorithm can progress linearly through the landscape, "submerging" one point ofter the other.

In total, the complexity of this algorithm is $\mathrm{O}(n \cdot m \cdot k + m \log m)$, where n is the average size of the neighborhood, k is the number of local optima and m the size of the solution space.

Algorithm 1 Basin-Fill Algorithm

1: $V \leftarrow \left\{(x = \vec{x}, f = f(\vec{x}), \vec{p} = \vec{0}) \mid x \in \text{sample}(\mathcal{S})\right\}$
2: $V \leftarrow$ sorted-by-f(V)
3: **for all** $x \in V$ **do**
4: $\quad N_d \leftarrow \{n \mid n \in N(x), f(n) < f(x)\}$
5: $\quad$ **if** $|N_d| = 0$ **then**
6: $\quad\quad p(x, x) \leftarrow 1$
7: $\quad$ **else**
8: $\quad\quad$ **for all** $n \in N_d$ **do**
9: $\quad\quad\quad$ **for all** $o \in \mathcal{O}$ **do**
10: $\quad\quad\quad\quad p(v, o) \leftarrow p(v, o) + p(n, o)/|N_d|$
11: $\quad\quad\quad$ **end for**
12: $\quad\quad$ **end for**
13: $\quad$ **end if**
14: **end for**

Algorithm 1 shows the pseudocode for the simplest version of the basin fill algorithm. A set is initialized with records of positions, fitness value, and convergence probabilities towards all optima that are initially all zero. This set is sorted into a vector that is then iterated according to fitness values, so that local optima are traversed before their corresponding basins and lower optima are traversed first. Then, either a local optimums has been found with no downward neighbors, in which case its convergence probability is set to one, while all other probabilities remain at zero. Otherwise, for all downward neighbors and all optima, the convergence probabilities of the current point are updated.

Please note, even though all examples in this section show only two-dimensional solution spaces for illustrative purposes, the algorithm itself can be applied to solution spaces with arbitrary numbers of dimensions.

4.1.2.1 Basin Convergence Volumes

Based on the convergence probabilities of a single optimum to a certain basin, the analysis can simply summarize all convergence probabilities towards a certain optimum and hence, calculate

the overall attractiveness of a certain optimum as shown in Equation 4.9. In other words, the probability of converging to a certain optimum from a random starting point is calculated.

$$\mathrm{Vol}(o) := \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} p(x, o) = \frac{1}{|\mathcal{S}|} \sum_{x \in \widetilde{b}(o)} p(x, o) \tag{4.9}$$

Based on this analysis, the distribution of convergence volumes and especially, the convergence volume of the global optimum can give a hint on the difficulty of finding each of the optima using trajectory-based optimization methods. The smaller the attractiveness of the global optimum, the smaller the chance to find it.

In Table 4.1, the numerical results are shown for a 2D multi-normal test function which is visualized in Figure 4.2b in the following section. For each optimum, its location, fitness value and all convergence volumes are included. For visual comparison with the figure, the optimum number zero is the left one, optimum number one is the bottom one, optimum two is at the top, optimum number three is in the center extending to the lower right, and optimum number four is to the far right in Figure 4.2b.

Table 4.1. Location, Fitness and Volumes of all Optima of the Multi Normal Test Functions shown in Figure 4.2

	Optima $o \in \mathcal{O}$				
	0	1	2	3	4
(x, y)	(-3.$\dot{3}$, -1.1$\dot{3}$)	(-0.7$\dot{3}$, -4.$\dot{6}$)	(0.1$\dot{6}$, 3.2$\dot{6}$)	(0.7$\dot{6}$, -2.5)	(3.1$\dot{3}$, -0.4$\dot{3}$)
$f(o)$	-0.0798	-0.5367	-0.0570	-0.1070	-0.8131
$\mathrm{Vol}(o)$	0.3091	0.0939	0.3552	0.1968	0.0450

4.1.2.2 Visualization

While for small combinatorial problem instances this analysis can be directly performed, an encoding using real vectors can only be sampled. However, if the fitness function is smooth and the sampled points are dense, the results can provide a good insight into the fitness function.

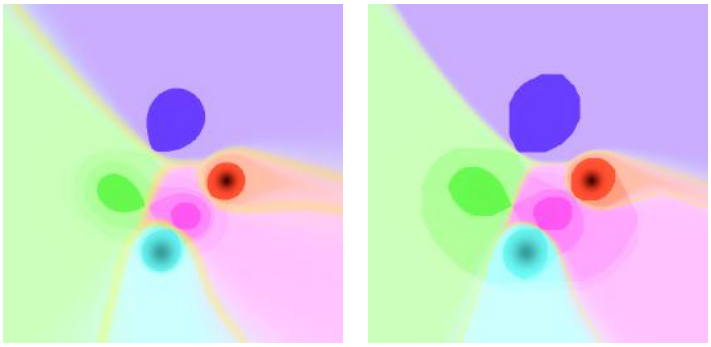
A simple method to visualize several important pieces of information concerning basins of attraction were developed:

- The first important piece of information is the fitness value itself which is encoded as the brightness of a pixel.
- For visualizing the affiliation of a pixel to a certain optimum different colors are used. One distinct color for each optimum.
- Finally, the convergence probabilities to all optima are visualized by varying intensity of the respective color according to these probabilities.
- As described in Section 4.1.3, a visualization of fitness barriers was also added by overlaying another complementary color along the barriers.

To add all this information into a single pixel, a color space transformation is needed as typical red green blue color values (RGB) used in computer displays cannot be easily mixed. To compensate for this, the *Lab* color space has been used which stands for lightness (L) and two chroma channels A and B as described in (Hunter, 1948). This colors space allows the simple averaging of colors to achieve mixing according to visual perception of humans. Therefore, it becomes easy to combine

the visualization of several features, in our case, fitness value and convergence probabilities towards different optima.

Figures 4.2 and 4.3 show several examples of the basin analysis of a multi-normal test function (see Section 4.1.4 for details). These test functions are simple superpositions of normal distributions. This provides excellent ground for studying fitness landscapes with different combinations of basins of attraction. These figures show different basins with different colors, where color intensity varies with convergence probability. Moreover, the barriers, which are explained in Section 4.1.3, are also visualized as separating lines between the differently colored basins.



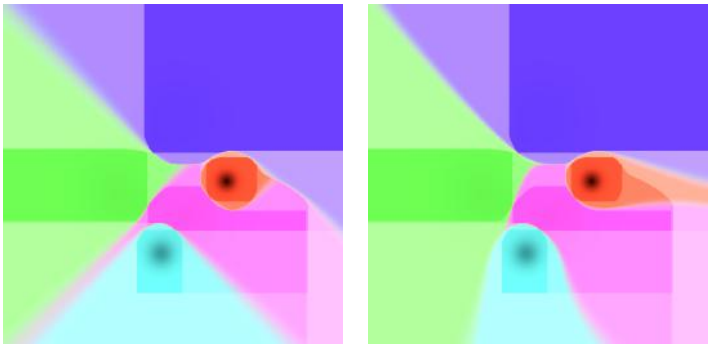
(a) Maximum Neighbor Distance 9

(b) Maximum Neighbor Distance 3

Figure 4.2. Multi Normal Test Function, Equal Weights

As can be seen in Figure 4.2, an increase in the size of the neighborhood does not significantly influence the structure of the basins of attraction. Figure 4.2a shows the visualization for an example with a very large neighborhood. Specifically, all neighbors with a distance of a maximum of nine grid steps in normalized space is allowed. Comparing this to Figure 4.2b, shows that the smaller neighborhood, gives slightly rougher edges of the different convergence levels but does not alter its shape.

The only significant change in shape can be seen when considering only a very small neighborhood as shown in Figure 4.3a, where only a maximum of one grid step away and consequently only four neighbors are considered. These neighbors lie above, below and left and right of the original point. This gives rise, then, to a more quadratic shape of the basin levels. However, the overall relation of shapes are, again, not greatly influenced.



(a) Equal Weights

(b) Slope-Dependent Weights

Figure 4.3. Multi Normal Test Function, Maximum Neighbor Distance 1

Furthermore, variations in the neighbor selection probabilities, as typically found in optimization algorithms, were tested. In all above examples, the probability of choosing each individual neighbor was uniformly distributed, or, in other words, every neighbor with better fitness had equal chance of being selected. If fitness distance of a neighbor is incorporated, for larger neighborhood sizes, hardly any change is visible. The only noticeable change has been produced by choosing a very small neighborhood size as shown in Figure 4.3b. Here, the slope change has been incorporated into the probability of choosing a certain neighbor, as shown in Equation 4.10, where $N{\downarrow}(x)$ is the set of downward neighbors as defined in Equation 4.11. As all slopes have the same direction, even for a maximization problem, the sum of slopes cancel a possible negative sign and ensure that $p_{\mathrm{s}}(n|x)$ is a probability density function over all neighbors of x.

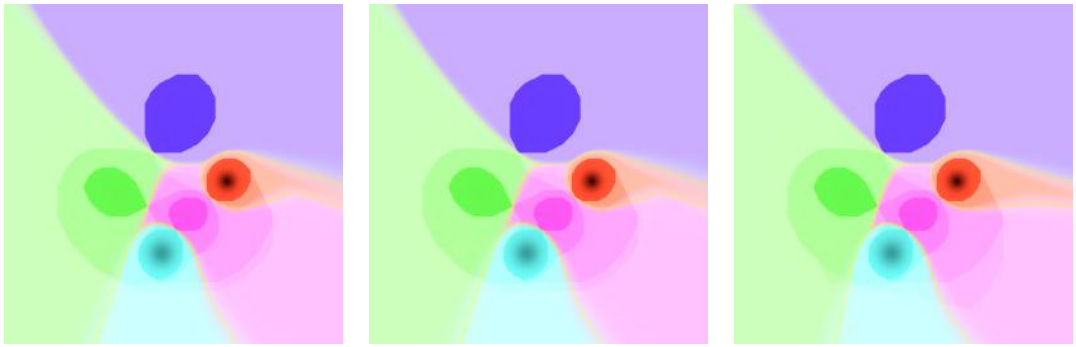
$$p_{\mathrm{s}}(n|x) := \frac{f(x) - f(n)}{\sum_{n' \in N{\downarrow}(x)} f(x) - f(n')} \tag{4.10}$$

$$N{\downarrow}(x) := \{n | n \in N(x), f(n) < f(x)\} \tag{4.11}$$

Another variation that was tried is the consideration of distances within the neighbor selection probabilities as shown in Equation 4.12.

$$p_{\mathrm{d}}(n|x) := \frac{d(n,x)}{\sum_{n' \in N{\downarrow}(x)} d(n',x)} \tag{4.12}$$

However, as can be seen, for example, in the comparison of Figures 4.4a and 4.4b, no significant effect on the size or shape of the basins is obtained.



(a) Equal Weights **(b)** Distance-Dependent Weights **(c)** Slope-Dependent Weights

Figure 4.4. Comparison of different neighborhood probabilities

This allows the assumption that basins of attraction are a relatively robust concept that do not vary too much with the choice of neighborhood, at least for real vector encoded solution spaces. In fact, the locality and, therefore, the mutual vicinity of solution candidates in a real vector has only very few meaningful choices that all allow a continuous and almost monotonous projection onto the assumed distance-based neighborhood. However, other solution spaces are much more difficult to visualize and have not been considered in this study of basins of attractions.

4.1.3 Barriers

As described in Section 2.2.4, a fitness barrier is defined as the minimum level (in a minimization problem) that has to be overcome to go from one point to another. This is shown in Equation 2.12 in Section 2.2.4. Using the samples and convergence probabilities form the basin analysis the barriers can empirically be estimated.

4.1.3.1 Barrier Calculation

The calculation scheme is based on the same ideas as the estimation of convergence probabilities for the basin analysis and is illustrated in Algorithm 2. In the beginning, the barriers between optima are initialized to positive infinity, except for barriers between an optimum and itself, where it is set to negative infinity. Starting from the lowest values towards higher values, the convergence of neighbors are analyzed and current barriers are updated by examining whether the connection between point v and neighbor n provides a traversal opportunity of lower fitness, and hence lowers the barrier, between two optima.

Algorithm 2 Barrier Calculation

$B \leftarrow \left\{(o_1, o_2, b) \;\middle|\; o_1 \in \mathcal{O}, o_2 \in \mathcal{O}, b = \pm\infty \text{ (see text)}\right\}$
$V \leftarrow \left\{(x = \vec{x}, f = f(\vec{x}), \vec{p} = \vec{0}) \mid x \in \text{sample}(\mathcal{S})\right\}$
$V \leftarrow$ sorted-by-f(V)
for all $v \in V$ **do**
  for all $n \in N(v)$ **do**
    for all $o_1 \in \{o | p(v, o) > 0\}$ **do**
      for all $o_2 \in \{o | p(n, o) > 0\}$ **do**
        $b \leftarrow \min\left\{b(o_1, o_2), b(o_2, o_1), \max\left\{f(v), f(n)\right\}\right\}$
        $b(o_1, o_2) \leftarrow b$
        $b(o_2, o_1) \leftarrow b$
      end for
    end for
  end for
end for

The worst-case run time complexity consists of the sorting of the fitness values, and a traversal of all points and their neighbors, therefore, $O(n \log n + n \cdot m \cdot k^2)$, where n is the number of considered solution candidates, m is the size of the neighborhood and k is the number of local optima. If the number of local optima is high, the calculation might become prohibitively slow. Also, as the number of local optima increases, the number of barriers increases quadratically which can exhaust the memory during calculation.

The numerical results, again, for the same multi-normal test function of Figure 4.2b as the previous examples, are given in Table 4.2. The values correspond to those points on the barrier lines with the least fitness between every pair of optima.

Table 4.2. Barrier Analysis of the Multi Normal Test Function as shown in Figure 4.2b

	0	1	2	3
1	-0.0683			
2	-0.0355	-0.0355		
3	-0.0683	-0.0900	-0.0355	
4	-0.0516	-0.0530	-0.0355	-0.0530

4.1.4 Multi-Normal Test Function

As a side product to the in-depth analysis of basins of attraction, and by studying the visualizations shown in Section 5.1, the importance and influence of not only the number but also the sizes and shapes of basins of attraction have become apparent. As can be seen in Section 5.1, where some

standard test functions are visualized, these test functions, although containing a large number of optima, have a very homogeneous and isotropic structure. Which makes it easy to prepare and tune an algorithm for the difficulties it will face.

However, when the basins of a fitness landscape are less homogeneous in size and shape, it can be postulated that an optimization would be much more difficult. Moreover, very often the size of a basin is assumed to be proportional to its depth. In other words, the slope towards an optimum is assumed to be similar throughout a landscape (see Jones, 1995).

An interesting example, therefore, that violates this assumption is the normal distribution: While the maximum height of the probability density function increases for decreasing standard deviation σ, the breadth of the basin of attraction decreases at the same time. This creates, in a certain sense, a counterintuitive or deceptive basin of attraction as the larger basins are shallower and, therefore, lead into the inferior local optima. On the other hand, deep optima will have only very narrow basins of attraction and should, therefore, be much harder to find.

In (Pitzer et al., 2010), a multi-normal test function was defined that is composed of an overlay of several normal probability density functions. Equation 4.13 gives the general definition of a multivariate normal distribution as found in e.g. (Larsen and Marx, 2011; Gut, 2009; Bishop, 2006). In the standard formulation, the covariance matrix $\mathbf{\Sigma} = E[\mathbf{X}\mathbf{X}^{\mathrm{T}}] - \mu\mu^{\mathrm{T}}$ is needed to describe all covariances between the different dimensions and should be symmetric positive definite.

$$\mathbf{X} \in N(\mu, \mathbf{\Sigma}), \mathbf{\Sigma} > 0 \quad f_{\mathbf{X}}(x) = \left(\frac{1}{2\pi}\right)^{\frac{n}{2}} \frac{1}{\sqrt{\det \mathbf{\Sigma}}} \exp\left(-\frac{1}{2}(x-\mu)^{\mathrm{T}} \mathbf{\Sigma}^{-1} (x-\mu)\right) \tag{4.13}$$

To simplify the analysis, only multivariate normal distributions with the same variance for all dimensions were used, i.e. all the covariances except auto-covariance are zero and all diagonal entries are the same. In this case, the multi normal distribution is a simple multiplication of several univariate normal distributions as shown in Equation 4.14, where the same value of σ sits at every position of the diagonal of Σ and n is the number of dimensions.

$$N(\vec{x}, \vec{\mu}, \sigma) := \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(x_i - \mu_i)^2\right) \tag{4.14}$$

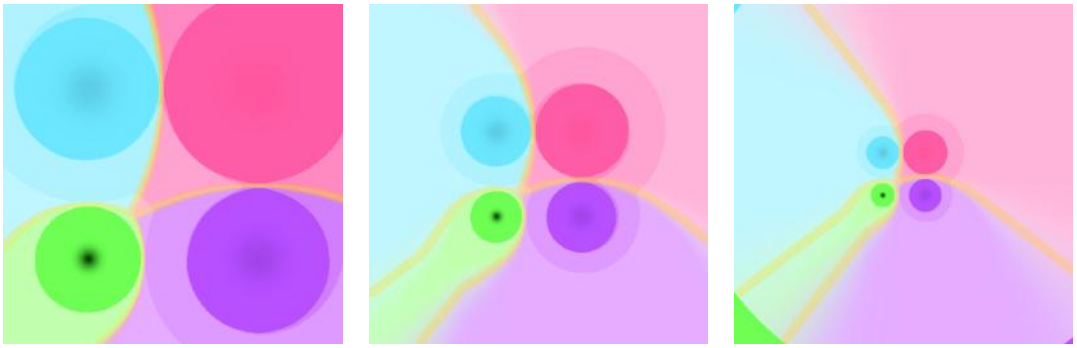
These simple multivariate normal distributions are then overlaid to build a more interesting, multi modal, landscape with different slopes and basin sizes, inversely proportional to their depth as shown in Equation 4.15. To keep with the metaphor of a basin, the fitness function is inverted, so that all normal distributions point downward.

$$f(\vec{x}) := -\sum_{i=1}^{n} N(\vec{x}, \vec{\mu_i}, \sigma_i) \tag{4.15}$$

4.1.4.1 Standardization

Finally, in (Pitzer et al., 2010), a simple scheme for the creation of a multidimensional multi normal test function was proposed to provide a test function similar to popular choices such as the Rastrigin (see Eiben and Smith, 2003), the Griewank (see Locatelli, 2003) or the Ackley test function (see Ackley, 1987; Bäck, 1996).

The idea is to have one deeper global optimum, surrounded by a set of shallower local optima throughout all dimensions and one very shallow and very broad optimum diametrically opposite the global optimum. Figure 4.5 shows the basin analysis of this standardized test function with two dimensions.



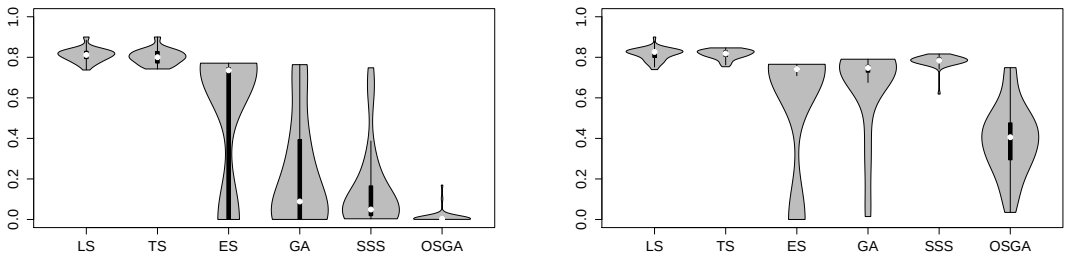
(a) Extent ±10 **(b)** Extent ±20 **(c)** Extent ±40

Figure 4.5. Standardized Multi-Normal Test Function: Only as the extent size is increased the dominance of the shallow optimum becomes clearly visible as well as the preemption of the basin of the global optimum.

4.1.5 Results

Using the basin and barrier analysis methodologies described here, a deeper understanding of the importance of a simple concept such as basins of attraction was gained. Through the analysis of different two dimensional test functions, using the efficient algorithm for the evaluation of basin volumes and barriers, a test function has been designed that, despite having only few local optima, can be remarkably difficult to optimize.

Figure 4.6 shows the results achieved previously in (Pitzer et al., 2010), where an array of algorithms was tuned to optimize the multi normal test function which seems to be quite difficult, while the performances of the same algorithms on larger problem instances of other standard test functions with many more local optima remained fairly descent as shown in Figure 4.7.

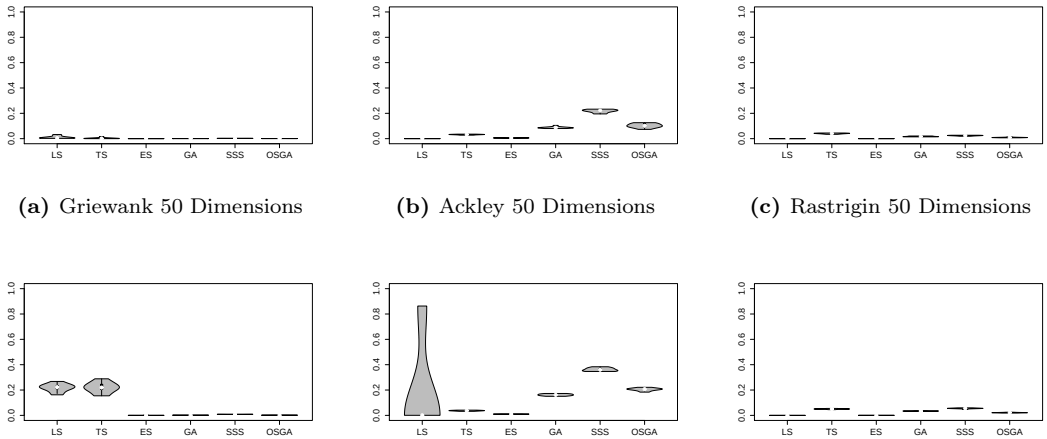


(a) 5 Dimensions **(b)** 10 Dimensions

Figure 4.6. Optimization Tests on the Multi Normal Test Function: Even though the number of local optima is relatively low and increases only quadratically with the number of dimensions, several algorithms have big difficulties coping with the deceptive nature of its basins of attraction.

While increasing the number of dimensions of the multi normal test function only slightly, causes severe problems, the same increase for standard test functions goes almost unnoticed in terms of performance for many of the algorithms. Furthermore, while for the multinormal test function the number of local optima only doubles with each increase in dimensions, for many of the other test functions, the number of local optima increases exponentially (Ackley, 1987).

An explanation to why the standard test functions are so much easier than the multinormal



(a) Griewank 50 Dimensions **(b)** Ackley 50 Dimensions **(c)** Rastrigin 50 Dimensions

(d) Griewank 100 Dimensions **(e)** Ackley 100 Dimensions **(f)** Rastrigin 100 Dimensions

Figure 4.7. Optimization Tests on other Test Functions: Even for much higher dimensions the same algorithms have no problems with the sometimes exponential increase in the number of local optima, which might come from the regularity of size, shape and slope of the basins of attraction.

function might be found by examining the two dimensional basins analysis plots in Section 5.1.

4.2 Isotropy

One of the issues that has, so far, received relatively little attention is the measurement of isotropy. As mentioned several times in Chapter 3, perfect isotropy is a convenient premise for the analysis of fitness landscapes. It would allow the generalization of any measurement over a representative sample of the solution space to the whole fitness landscape. In some cases, such as the extremely epistatic NK landscapes, the assumption of isotropy seems justified. For simpler landscapes with a higher degree of order however, this claim should be validated. In the following, a first attempt towards the empirical quantification of isotropy is made.

4.2.1 Dependence on Other Measures

One important aspect of isotropy is its dependence on other measures. If a landscape is said to be isotropic, it is "the same in every way". However, to quantify this, it has to be specified in which way it is the same. Exactly for this reason, *a way* of measuring the landscape's properties has to be used. Luckily, as comprehensively detailed in Chapter 3, there are many ways of measuring fitness landscapes. The basic idea is to take these measurements as *the way* a landscape is, however, confined to different regions of the landscape. In (Stadler and Grüner, 1993), these subspaces are non-intersecting partitions of the solution space. However, for practical problem sizes the probability of intersection even of larger samples is vanishingly small.

4.2.2 Subsample Overlap Probabilities

To illustrate the probabilities of overlaps, the same calculation scheme as in the birthday paradox (Naus, 1968) can be used. While in the birthday paradox, chances to have coinciding birthdays

of just a few people is surprisingly high, in our case, even with a substantial sample in a problem instance of practical size the probability of coincidences is surprisingly low.

For most of the measurements of Chapter 3, a connected trajectory is used to generate the underlying sample for further analysis. It can be assumed that a trajectory-based sample is more connected, or more local, than a random sample. For simplicity of the argument, a random sample will be assumed that can, therefore, serve as an upper bound of the true overlap probability between trajectories.

For a solution space of permutations of length n, the number of possible permutations is $n!$, which is generally denoted as $|\mathcal{S}| = n!$. If a random walk of length l is performed and repeated s times, a total number of $w = l \cdot s$ walk samples is obtained. So, there are $\binom{w}{2}$ pairs of points that could potentially overlap. However, it is easier to calculate the chances of no overlap as shown in Equation 4.16, where w is the number of walk samples and n is the total size of the solution space. The first sample has a 100% chance of *not* coinciding with previous samples (because there are none yet), the second sample has a chance of $(n-1)/n$, i.e. there are $n-1$ non-overlapping solution candidates out of n, and so on.

$$\begin{aligned} 1 - p(w,n) &= \frac{n}{n} \cdot \frac{n-1}{n} \cdot \frac{n-2}{n} \cdots \frac{n-w+1}{n} = \\ &= \frac{n \cdot (n-1) \cdot (n-2) \cdots (n-w+1)}{n^w} = \\ &= \frac{n!}{(n-w)! \cdot n^w} \end{aligned} \tag{4.16}$$

Equation 4.16 can become difficult to calculate for larger numbers. However, using either the Stirling formula or a Taylor expansion of the exponential function, an approximation can be derived as shown in Equation 4.17.

$$1 - p(w,n) = \exp\left(\frac{-w \cdot (w-1)}{2 \cdot n}\right) \tag{4.17}$$

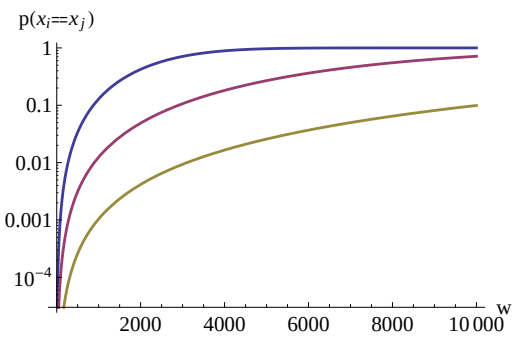
Using this formulation, the probability of any coinciding samples in typical solution spaces can be approximated. Figures 4.8 and 4.9 show the graphs for different sample sizes. These are for purely random samples. It can be assumed that the probability should not be higher for trajectory-based samples. Actually, even if the trajectories touch each other in certain cases this should not cause big problems.

Finally, Table 4.3 lists some suggested maximum sample sizes for different problem sizes to keep the chance of any overlap reasonably low. As can be seen, for typical combinatorial optimization problems in the range of 30 and upwards there is no real danger. The same holds for binary vectors of more then 100 dimensions.

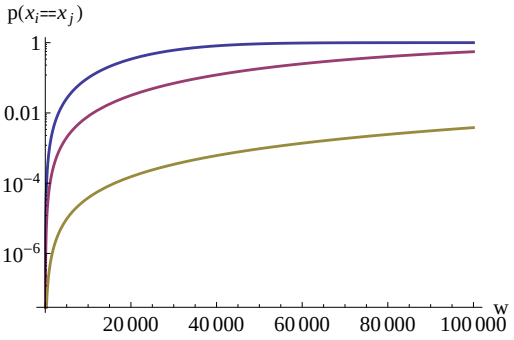
Even for the very tightly coupled combinatorial landscapes, it is therefore, extremely unlikely, that two or more continuous random samples are overlapping. For this reason, the possibility of different samples from different regions being, by chance, actually from the same region can be neglected. This yields a very simple and pragmatic approach: Some measurements are repeated over several different regions of a fitness landscape and are then compared if "they look the same".

4.2.3 Definition

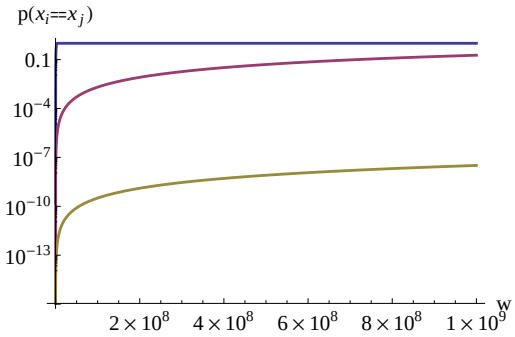
Similar to the definition of *empirical isotropy* given in Section 3.11.1, another notion can be defined that does not require a strict partitioning for reasons illustrated in the Section 4.2.2. In general, the definition of isotropy has to be relative to, or based on, another measure as discussed in Section 4.2.1. Putting all these things together we arrive at a surprisingly simple concept for describing isotropy as shown in Equation 4.18, where $\mathcal{M}$ is any fitness landscape measure from



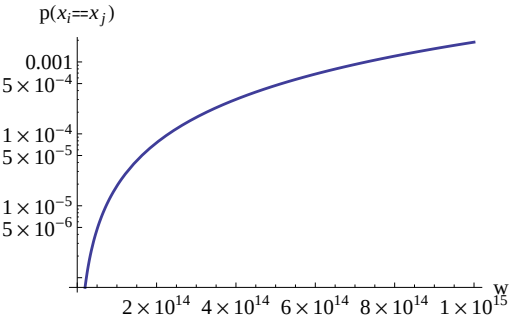
(a) Permutations of Length 10, 11, and 12 with Sample Sizes up to 10 000



(b) Permutations of Length 12, 13, and 15 with Sample Sizes up to 100 000

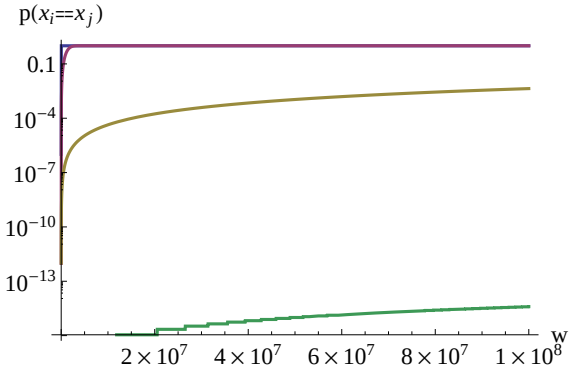


(c) Permutations of Length 15, 20, and 25 with Sample SizeS up to 10^9

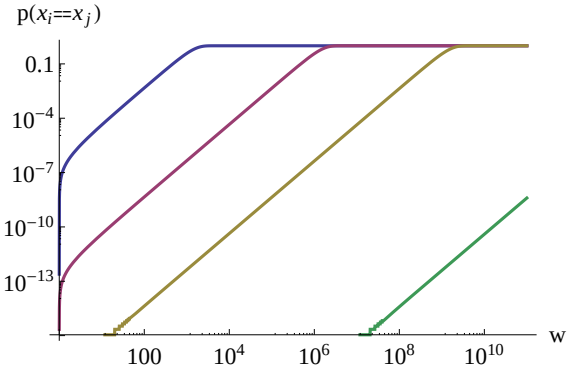


(d) Permutations of Length 30 with Sample Sizes up to 10^{15}

Figure 4.8. Log Plot of the Probabilities of Overlapping Samples in Combinatorial Solution Spaces: The x-axes show the number of samples while the y-axes contain the overlap probabilities. As the solution space sizes approach dimensions usually tackled with metaheuristics, the probability of any overlaps becomes vanishingly small.



(a) Binary Vectors of Lengths 20, 40, 60, and 100 with Samples Sizes up to 100 000 000 steps



(b) Log Log Plot for Binary Vectors of Lengths 20, 40, 60, and 100 with Sample Sizes up to 10^{10} steps

Figure 4.9. Log Plot of Probabilities of Any Samples Overlapping in Binary Solution Spaces: Here, the number of dimensions has to be somewhat larger than for combinatorial problems to achieve the same solution space size. However, again, those problem sizes that are usually considered are large enough to make overlaps extremely unlikely.

Table 4.3. Maximum Sample Sizes for Combinatorial and Binary Problems of Different Sizes with Reasonably Low Overlap Probabilities

Type	Problem Size	Max Sample Size	Overlap Probability
Permutation	10	$85{\cdot}10^{0}$	$\sim 0.1\%$
Permutation	15	$50{\cdot}10^{3}$	$\sim 0.1\%$
Permutation	20	$70{\cdot}10^{6}$	$\sim 0.1\%$
Permutation	25	$150{\cdot}10^{9}$	$\sim 0.1\%$
Permutation	30	$700{\cdot}10^{12}$	$\sim 0.1\%$
Permutation	35	$4{\cdot}10^{18}$	$\sim 0.1\%$
Permutation	40	$40{\cdot}10^{21}$	$\sim 0.1\%$
Binary Vector	10	$2{\cdot}10^{0}$	$\sim 0.1\%$
Binary Vector	20	$50{\cdot}10^{0}$	$\sim 0.1\%$
Binary Vector	30	$1.5{\cdot}10^{3}$	$\sim 0.1\%$
Binary Vector	40	$50{\cdot}10^{3}$	$\sim 0.1\%$
Binary Vector	60	$50{\cdot}10^{6}$	$\sim 0.1\%$
Binary Vector	80	$50{\cdot}10^{9}$	$\sim 0.1\%$
Binary Vector	100	$50{\cdot}10^{12}$	$\sim 0.1\%$
Binary Vector	150	$2{\cdot}10^{21}$	$\sim 0.1\%$

the previous chapter which is evaluated, once, on a subset S' of $\mathcal{S}$ and, once on the whole solution space. If these measures substantially deviate from each other, the landscape is not isotropic. It should be noted that, of course, for extremely small samples these values will deviate which can, however, be accounted for using an appropriate significance level in the comparison.

$$\forall(S' \subseteq \mathcal{S}) \quad \mathcal{M}\left(\{S', f, \mathcal{X}\}\right) \cong \mathcal{M}\left(\{\mathcal{S}, f, \mathcal{X}\}\right) \tag{4.18}$$

By extension, this can be transformed into the more practically applicable form shown in Equation 4.19, where, instead of comparing the measure of a subsample with the measure over the whole solution space, just two independent samples $S_1, S_2 \subseteq \mathcal{S}$ are compared with each other. This comparison is much more practical as a measure calculated over the whole solution space is often only practical for small and relatively uninteresting solution spaces.

$$\forall(S_1, S_2 \subseteq \mathcal{S}) \quad \mathcal{M}\left(\{S_1, f, \mathcal{X}\}\right) \cong \mathcal{M}\left(\{S_2, f, \mathcal{X}\}\right) \tag{4.19}$$

If this defines isotropy, the next question to be asked is the following: "At what point and to what extend can we talk about *an*isotropy?" A simple approach would be to just define the violation of the equality in Equations 4.18 and 4.19 as the extend of anisotropy. However, this could neglect the inherent variance of different measures over different samples. Nevertheless, it is this variance that is attempted to be captured. Therefore, taking the uncorrected variance is already a valid measurement of anisotropy. If a comparison of different isotropy measures is desired, however, it seems more natural to take inherent variance into account and focus on the variations that stem from locality. Equation 4.20 shows the simplest possible definition of anisotropy or simply the variance of a fitness landscape analysis measure of different subsets of the landscape. Of course, in practical scenarios the variance will not be evaluated over all possible subsets but only a, possibly random, selection $S' \subseteq \mathcal{P}(\mathcal{S})$, ideally, where $\forall(S_i, S_j \in S')\ |S_i| = |S_j|$.

$$\widetilde{\mathcal{I}_{\mathcal{M}}}\left(\{S', f, \mathcal{X}\}\right) := \mathrm{Var}\Big[\left\{\mathcal{M}\left(\{S_i, f, \mathcal{X}\}\right) \middle| S_i \in S'\right\}\Big] \tag{4.20}$$

This definition is similar to the previous definition in (Stadler and Grüner, 1993) insofar as the isotropy is described by the variance of another, substrate measure. However, it is less accurate but in turn easier to apply practically. In fact, any measure of the distribution over the samples at different locations can give insights into the isotropy of a fitness landscape.

As the variance is relatively sensitive to extreme outliers, which could easily be introduced with small samples, a slightly different formulation for the practical measurement of anisotropy is proposed in Equation 4.21, where the difference between the 95^{th} and the 5^{th} percentile are used instead as further defined in Equations 4.22 and 4.23.

$$\mathcal{I}_{\mathcal{M}}\left(\{S', f, \mathcal{X}\}\right) := \text{QDiff}_{.05}^{.95}\Big[\left\{\mathcal{M}\left(\{S_i, f, \mathcal{X}\}\right) \middle| S_i \in S'\right\}\Big] \tag{4.21}$$

$$\text{QDiff}_{\text{l}}^{\text{h}}(X) := Q_h(X) - Q_l(X) \tag{4.22}$$

$$\text{Q}_q(X) := \inf\left\{x \in X \;\middle|\; q \leq Pr(X \leq x)\right\} \tag{4.23}$$

4.2.4 Subsample Selection

As briefly mentioned in the last section, the choice of S' should ideally be in a way, where all subsamples have the same size. Additionally, to be able to analyze the variance of *local* properties the subsample should be locally confined and separated from other subsamples, as shown in Equation 4.24, where $\overline{d}$ is an extension to the underlying distance d between solution candidates to give the average distance inside and between two subsamples as defined in Equations 4.25 and 4.26, respectively.

$$\forall(S_i, S_j \in S')\ \overline{d}(S_i) < \overline{d}(S_i, S_j) \tag{4.24}$$

$$\overline{d}(S) := \text{E}\Big[\left\{d(x, y) \;\middle|\; x, y \in S\right\}\Big] \tag{4.25}$$

$$\overline{d}(S_1, S_2) := \text{E}\Big[\left\{d(x, y) \;\middle|\; x \in S_1, y \in S_2\right\}\Big] \tag{4.26}$$

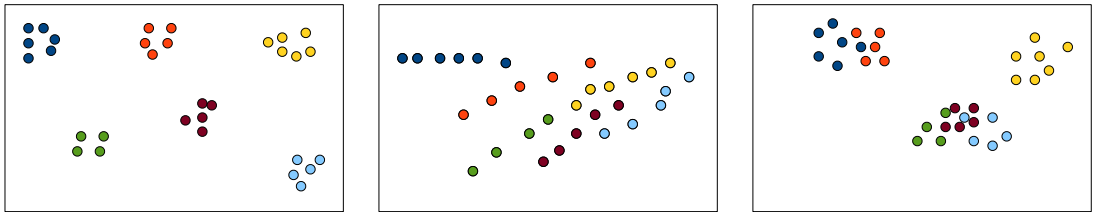
This is a very mild selection criterion, and it should not even matter too much if it is violated sometimes as long as the distribution and the variance still reflect variation over enough different locations. In other words, our samples should be local and not overlap with other samples too much. Please note that a strict separation is not required, i.e. several points of a sample might be inside the "territory" of another subsample without causing too much trouble.

Figure 4.10 shows the different scenarios. First, in Figure 4.10a, the ideal scenario, where the sample represents the local neighborhood around a single solution candidate. In Figure 4.10b, the problem is not so much that the samples overlap but that they do not represent a single location in the landscape. Finally, Figure 4.10c shows a further example of good locality, despite overlaps with neighboring subsamples. In Section 4.2.5.2, a method is described to create an unbiased random sample, centered around a single solution candidate, which has very good locality and is a true random sample without a bias towards going back to the starting point.

4.2.5 Multi-Trajectory Analysis

The basis for the analysis of isotropic features is the analysis of a distribution over several subsamples of a solution space. Therefore, the next logical step is to go from the simple singular trajectories used in the previous chapter to several, hopefully, independent, non-overlapping samples with high locality. In its simplest form, this is just the analysis of several trajectories from different starting points in the landscape.

Figure 4.11 shows several examples of an autocorrelation distribution analysis. In this case all statistically significant autocorrelations have been used to create quantiles which are visualized in Figure 4.11 with different combinations of repetitions and different walk lengths. Interestingly, a higher number of repetitions has almost no influence on the distribution of the quantiles. However,



(a) Good Locality

(b) Bad Locality, although without Overlaps

(c) Good Locality, despite Overlaps

Figure 4.10. Subsample Separation & Locality: Different scenarios with good or bad locality. The problem is not so much the clear delineation from other subsamples as the representativeness of a local point.

as the individual trajectories are made longer, the auto correlation quantiles get more similar to each other as can be seen in Figures 4.11(b), (e), and (h) and even more for Figures 4.11(c), (f), and (i).

As the lengths of the individual walks increase, the statistical accuracy of each of the walks increases and all of them start to measure more or less the same, especially for the lower step counts. This insight might seem slightly discouraging as it implies that the more significant the results become the less we are able to observe isotropy at all. Ideally it would be desirable to see measurements for very short walks as these would exhibit very local features, however, these results would be quite insignificant and mostly attributable to inherent variation instead of location dependent variation.

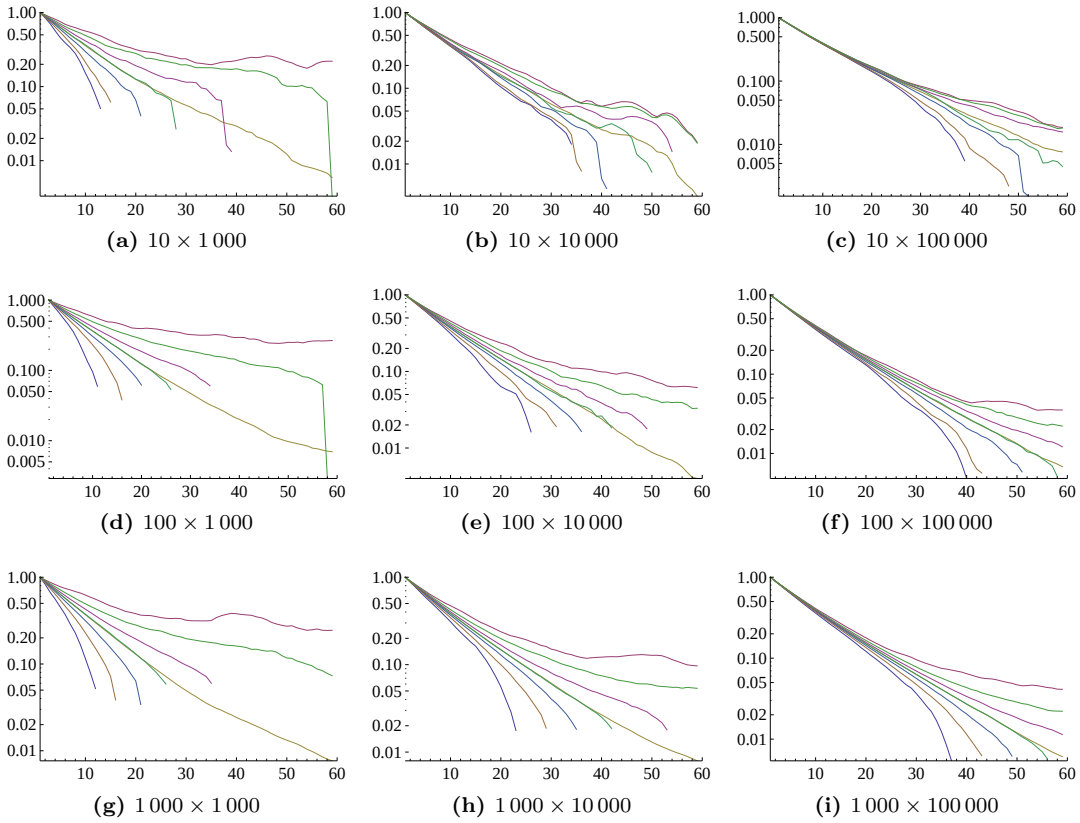
While the auto correlation analysis of Figure 4.11 shows the different quantiles over the whole series of auto correlation values, as stated in Equation 4.20, for the measurement of isotropy, however, the *variance*, or the spread between the quantiles, is of more interest. In Figure 4.12 the quantile spread or the difference between 95^{th} and the 5^{th} percentile is shown. Figure 4.12a shows the actual difference between the two quantiles, while Figure 4.12b shows the normalized differences: As the walks of different lengths have different statistical significances and, therefore, different inherent variances, this effect has to be accounted for by multiplying the differences with the corresponding standard error, which is the square root of the samples' sizes, or in this case, the square root of the walks' lengths.

Figure 4.13 shows a second example of auto correlation quantiles. The only difference between the individual curves are the significance thresholds due to different walk lengths and the different smoothnesses due to the different number of start locations throughout the landscape.

Finally, in Figure 4.14, the whole range of the auto correlation is visualized, i.e. QDiff_0^1. In this case, the influence of outliers blurs the crisp structure that can be observed in the previous examples in Figures 4.11 and 4.13.

As can be seen from these examples, for the lower step sizes the correspondence is relatively clear. Only with a low number of restarts, the inherent variance causes more severe deviations. However, with at least 100 starting locations throughout the landscape, the different analyses are quite stable and comparable. The apparent peaks in the individual curves are due to the cut-off of insignificant correlations, which are different for different walk lengths. Therefore, the individual isotropy measures can only be compared up to this point. However, most frequently, only $\rho(1)$, or the auto correlation after one step is used.

The apparent match of different walk lengths, especially for the first few steps of the auto correlation functions in combination to the quantile spread or confidence interval size, gives solid repeatable results in these cases.

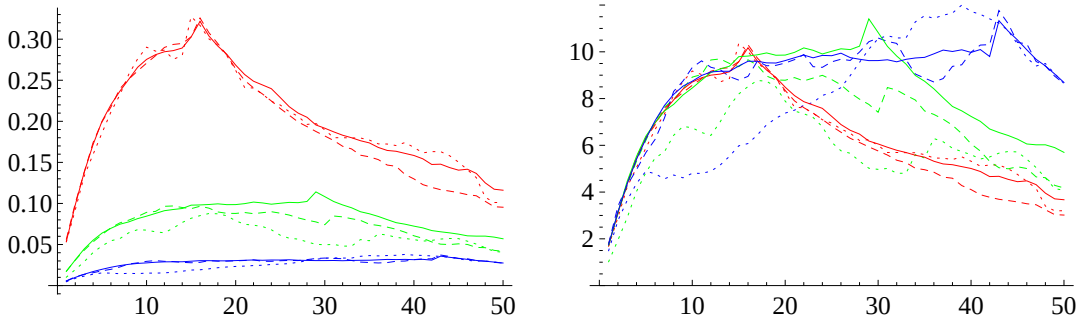


(a) $10 \times 1\,000$ **(b)** $10 \times 10\,000$ **(c)** $10 \times 100\,000$

(d) $100 \times 1\,000$ **(e)** $100 \times 10\,000$ **(f)** $100 \times 100\,000$

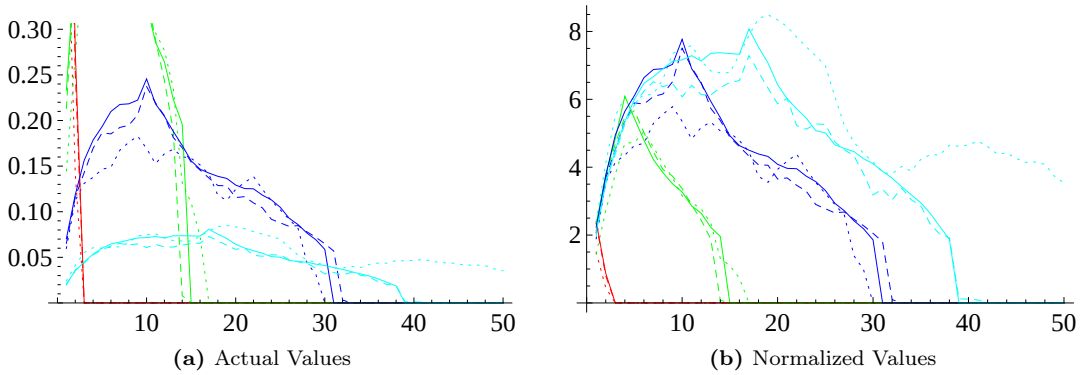
(g) $1\,000 \times 1\,000$ **(h)** $1\,000 \times 10\,000$ **(i)** $1\,000 \times 100\,000$

Figure 4.11. Example of Auto Correlation Analysis of Several Samples: The figures show the logarithm of the average and the autocorrelation quantiles 0, .05, .25, .5, .75, .95, and 1 for different numbers of repetitions times the number of steps in the random walk.



(a) Actual Values **(b)** Normalized Values

Figure 4.12. Quantile Spread Analysis of the Auto Correlation Function: The difference of the 95^{th} and the 5^{th} percentile of the auto correlation function for the `bur26a` instance of the QAPLIB. The colors are red for 1 000, green for 10 000, and blue for 100 000 steps, while the dotted lines are for 10, dashed for 100 and continuous lines for 1 000 different locations. The distinctive peaks occur when the lower quantile is forced to zero because of insufficient significance in the auto correlation analysis. It can be seen as the "resolution" or statistical significance boundary of the analysis for different walk lengths. Dividing all values by the square root of the respective walk lengths shows their relative correspondence in Figure (b).



(a) Actual Values **(b)** Normalized Values

Figure 4.13. Another Example of Auto Correlation Quantile Spread: For another QAPLIB instance, `els19`, the values of the quantile spread, after normalization can, again, be estimated regardless of walk length or number of repetitions. This time, the color coded walk lengths were 10, 100, 1 000, and 10 000 steps.

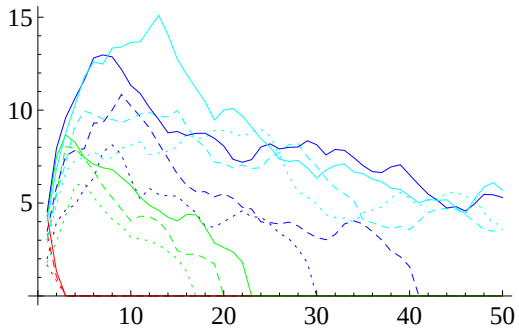


Figure 4.14. Example of Auto Correlation Range: In this case the whole range, i.e. QDiff_0^1, of the auto correlation function is shown, which gives a less clear picture of the anisotropy over different walk lengths and sample numbers. The colors are again 10, 100, 1 000, and 10 000 steps, while the line styles dotted, dashed and continuous are again 10, 100, and 1 000 different locations, respectively.

However, these walk lengths have two important shortcomings:

- The walks are quite short in comparison with typical fitness landscape analysis, where at least 100 000 steps up to several million steps are usually analyzed to get a good measurement of the whole solution space.
- The walks are quite long in comparison with the diameter of the whole solution space. This results in values that do not accurately represent local features of the solution space but are rather averages over large areas.

This puts us into the dilemma that typical walks are at the same time too short for a good analysis and too long for the analysis of local features. In Section 4.2.5.2, a simple method is presented to mitigate this problem.

4.2.5.1 Initialization Strategies

While the low likelihood of overlapping random samples in reasonably sized solution spaces has been discussed, there exist other possibilities for initialization. As a complete analysis is restarted from different points, of prime interest is a representative sample of the different "geological" structures of the fitness landscape at this location.

A very straightforward possibility is to evenly space out the starting points throughout the fitness landscape, creating a grid with several samples along each dimension. While this sounds like a very fair and easy possibility, we tend to forget the typical number of dimensions in hard optimization problems. With 100 dimensions and 100 samples for each axis we end up with $100^{100} = 10^{200}$ grid points. Even if just two samples are taken, there are still one million starting points. In the case of a binary vector, actually the whole solution space would be enumerated by taking two samples along each axis.

Of course, one could come up with new strategies to sample along the axes, or take only certain axes or take variable number of samples on different axes, and so on. However, with all these complications in mind it seems most reasonable, and generally applicable to just take random samples for the starting points. They are easily scalable, require minimal effort for even distribution and are unbiased and not influenced by size and number of dimensions.

In certain cases, however, it can be beneficial to obtain a biased sample, for example, to explore parts of the landscapes that are more likely to be perceived by an optimization algorithm. It is important to be *extremely* careful with any conclusions, however, as any bias in the choice of the samples could be misleading. Therefore, it is recommended to use a very exploratory optimization technique to obtain the optimized sample. One suggestion is referenced in Section 3.2.2, where a very slowly converging evolutionary algorithm is used to obtain a delicately optimized sample with high diversity. Even in this case, the sample should be essentially randomly distributed throughout the solution space, albeit within regions with higher fitness. This is sometimes called an *optimized sample*, not because of its suitability for analysis, but because it has been obtained through a precursory optimization step. In general, an analysis using optimized samples should always be complemented with an analysis using an unbiased sample to avoid a too narrow view.

4.2.5.2 Combination with Repeated Local Analysis

As described in Section 4.2.5, random restarts alone have the problem of either insufficient significance or insufficient locality. One possibility of increasing locality would be to force the random walk to return to the starting point after some time, this, however, would make this walk less random. Therefore, another simple possibility is to restart the walk from the same staring point and combine the results from several iterations from the same starting point. This gives analysis trajectories with good locality and, with sufficient repetitions, enough statistical power to provide

useful measurements. Now, the analysis of any underlying values has to undergo two aggregation steps:

1. The first step is to aggregate the local repeats into an analysis that resembles a single, longer analysis.
2. The second step is to perform the usual variability analysis using any anistropy measure.

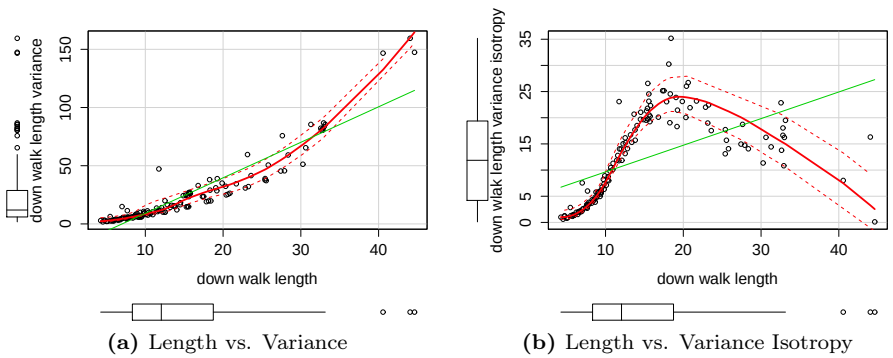
4.2.5.3 Aggregation

In the case of auto correlation, the repeated auto correlation functions can be simply averaged to give the average auto correlation from the central starting point. This average auto correlation function can be used as before for the distribution analysis of isotropy.

Similarly, other fitness landscape analysis measures, such as information content or up-down walk lengths can be repeated from a single starting point, averaged locally and then subjected to a landscape-wide distribution analysis.

The information analysis introduced in Section 3.3.5 can also be extended. As described in Section 4.3 it has been extended with an improved algorithm for the selection of ε-differences. While this speeds up the analysis and provides a more concise overview of the different information analysis results, it slightly complicates the aggregation of repeated information analysis walks. The basic idea, however, is very similar to the auto correlation aggregation. From several parallel walks from the same starting point the results have to be combined into a single trajectory analysis. This can, again be achieved by averaging the results with equal ε values.

Another simple possibility is to measure up-down walk characteristics using multi-trajectory repeated local analysis. In this case the up-down walk characteristics can be combined similarly. Care has to be taken, however, as noted in the analysis of auto correlation, that the inner walk lengths limit the maximum possible values of the up or down walks. This is especially true for the measurable up-down walk length variance. This can be seen in Figure 4.15, where the down walk length variance is plotted against the down walk length itself and once again using the derived down walk length variance isotropy which uses much shorter walks. In the second case, similar to the autocorrelation analysis, the anistropy measure declines for longer walks. The simple reason is that if the total walk length is shorter, longer sub walks cannot be repeated so often and, therefore, have less variance.



(a) Length vs. Variance **(b)** Length vs. Variance Isotropy

Figure 4.15. Isotropy Error Threshold: Because of the shortened walks for the measurements of local properties and estimation of isotropy, some measurements are only accurate up to a certain error threshold.

4.2.5.4 Other Uses of Multi Trajectory Analysis

The multi trajectory analysis is actually not confined to the application of isotropy analysis. Several more complex measures could actually benefit from the simple scheme of different restarts.

One example is the neutrality analysis. While one option is to to attempt to perform a neutral walk and if that fails continue with a random walk instead, the resulting statistics can easily get blurred. To effectively perform neutral walk attempts without polluting the results with partial random walks, different starting points should be tried. If no neutral step is found any more, the analysis is aborted at this location.

Alternatively, similar to a multiple-restart local search, an up-down walk can be dissected into non-consecutive up-down phases. Instead, from a single starting point, several up and several down walks can be tried. This facilitates the analysis with respect to a random starting point in addition to the analysis between local extremal points.

4.2.6 Combined Isotropy Measures

Ideally, *the* isotropy of a certain problem instance could be described by a single value. However, form the existing tests, this does not seem to be immediately possible. By examining the box charts of several problem libraries and correlation analysis of several isotropy values it can be concluded that further normalization is necessary before these values can be reasonably combined. In fact even the base measures currently lack normalization that could make them directly comparable. This is a non-trivial issue, since solution space space size, neighborhood size and the distribution characteristics of the measurements are all involved. As much as we would have liked to create a weighted average of different isotropy measures, only appropriate normalization can put them on equal ground and make them combinable.

The second question, even if it is assumed that different isotropy measures based on different base measures are able to obtain comparable estimates of isotropy, it is questionable whether an average value would be sufficient. Just because anisotropy cannot be observed from a certain perspective or even from several perspectives does not imply that a problem instance is isotropic. Therefore, once normalization issues have been settled, it seems much more reasonable to combine them using a higher order norm such as the uniform norm shown in Equation 4.27 or at least a higher p-norm shown in Equation 4.28 with $p \geq 2$.

$$||x||_\infty := \max\{|x_1|, \ldots, |x_n|\} \tag{4.27}$$

$$||x||_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}} \tag{4.28}$$

4.2.7 Non-Isotropic NK Landscapes: NKi Landscapes

With the popularity of NK landscapes and the history of introduction of variants as a showcase for new measures, it was attempted in this work to create a variation of NK landscapes. As in this work new measures of isotropy are proposed it is also attempted to construct a variation of NK landscapes with varying isotropy. These *NKi landscapes* are designed to introduce anisotropy into the extremely isotropic NK landscapes.

Before we begin to modify NK landscapes, or actually the more general block model introduced in (Altenberg, 1994b) as described in Section 3.14.3.4, let us recall Equation 3.62 which gives the

general structure.

$$f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(n_{i,1}(x), n_{i,2}(x), \ldots, n_{i,k_i}(x)) \qquad \text{(3.62 revisited)}$$

The basic idea of NKi landscapes is to introduce location-based differences, so that different locations on the bit vectors exhibit different properties. This has been tried with several different configurations. There are two aspects of NK landscapes that can be modified to achieve anisotropy:

- On the one hand, the component selection functions $n_{i,j}$, which determine the epistatic structure, can be modified to group certain solution candidate components x_i together.
- So that, on the other hand, the component functions f_i and the magnitude of their contribution to the overall fitness can differentiate between different "areas" of the solution candidates.

In such a tightly coupled landscape this is relatively difficult as every solution candidate component, or in other words, every dimension is only a binary value. Nevertheless, some ideas are presented to make this structure less isotropic.

The following list briefly describes the different modifications that were examined for the neighborhood interaction or component selection function to influence the epistatis setup.

- The basic *random interaction* model randomly assigns solution candidate components, or genes, to fitness component functions. This is the default and equals the generalized block model.
- With a *limited range interaction*, a model similar to the adjacent neighborhood structure has been introduced, where not only directly adjacent neighbors but neighbors up to a certain maximum distance are assigned to each component function.
- With a *deliberate interaction size* the number of epistatic interactions of different fitness component functions has been directly selected so that f_i has exactly i input values.
- Similarly, a *sorted random interaction* model first assigns random interactions to all fitness components but then sorts the genes according to their component function affinity. By itself this does not change the interaction structure but relies on a systematic change of fitness component function magnitude.

Moreover, the following changes to fitness component functions were examined. In fact, the fitness component functions themselves cannot easily be influenced, as these random tables are not actually tangible. As explained in Section 3.14.3.5, the random tables are in general too large to be available and have, therefore, been replaced by a pseudo-random number generator, or a hash function in our implementation. For this reason the only accessible aspect are artificial weights for each of the fitness component functions as shown in Equation 4.29, which were already part of the original NK landscapes but have been subsequently removed from the generalized block model.

$$f(x) := \sum_{i=1}^{n} c_i \cdot f_i(n_{i,1}(x), n_{i,2}(x), \ldots, n_{i,k_i}(x)) \qquad (4.29)$$

By systematically assigning these component function weights in concert with the modified neighborhood interactions, described above, it was tried to introduce some anisotropy. The strategies for the fitness component weights were the following:

- Assigning an *equal weight* to each of the fitness component functions is the default and should, conceptually, not cause any anistropy.
- However, by assigning *linearly increasing weights* with $c_i = i$ it was attempted to disturb the equal distribution of landscape features.

- An even more disruptive change of the fitness landscape was attempted by using *exponential weights* for the different fitness component functions, where $c_i = 2^i$.

Together these components can create a skewed NK landscapes, where e.g. the first dimensions have much more influence on the overall fitness than the later ones. This allows to introduce measurable anisotropy as documented in Section 5.2.

4.2.8 Summary

The analysis of isotropy is a delicate subject. In many practical cases, problem instances are highly isotropic. In these cases, isotropy measurement might seem superfluous and not worth the effort. However, to discover that these instances are isotropic is an important reassurance for the validity of many other analysis results. Currently, it is relatively easy to detect the presence of significant isotropy which can have observable impacts on both other measurements as well as optimization attempts. However, as detailed in Section 5.2 the exact measurement of isotropy requires careful attention to details such as inner walk length and used neighborhood definitions to become observable. Using one particular neighborhood a landscape can seem just rugged, while, on closer examination the underlying anisotropy can be discovered.

4.3 Extensions of Existing Measures

During the course of implementing existing measures, some new ideas where tried as minor extensions of existing measures, which are summarized for reference in the following paragraphs.

4.3.1 Information Analysis

In the original paper (Vassilev et al., 2000), where information analysis is introduced, the possibility of different thresholds ε for the relaxed sign function is described. However, meaningful choices are all fitness differences across the underlying trajectory. To limit this number, as a first step only a fixed number of quantiles over the fitness differences was analyzed. However, in some cases with high neutrality, many of these quantiles are equal to zero or duplicate. In these cases it might be more interesting to explore different quantiles. For these cases, the equal ε values are first merged to a single pseudo-quantile and the remaining ranges are sampled with a correspondingly increased number of points. Moreover, when the resulting entropy values are plotted over these, more interesting, ε values a very characteristic distribution can be observed. In particular, in the experiments here, a characteristic maximum of each entropy value for certain ε values was found, which could highlight those fitness difference levels, where the grand structure of the fitness landscape becomes visible.

In addition, the entropy values over larger combination of micro states have been examined. While in the original description only two consecutive slope changes where used for the definition of micro states, the analysis with up to four consecutive symbols has been performed. However, as the number of slope combinations increases, the number of shape repetitions decreases accordingly and, therefore, reduces the entropy measures and is, in general, proportional to the existing measures. However, an interesting extension of this increased shape size would be to directly compare the frequencies of these larger structures.

4.3.2 Other Extensions

For the up-down walks the typical fitness levels at which the extrema occur have also been analyzed. These values are a little problematic since they correspond directly to fitness levels of the

concrete problem instance. Therefore, they cannot by directly compared. Instead they have to be normalized with respect to the fitness value distribution of the problem instances. For example between the theoretical lower and upper bounds of a problem instance or between the average and the best solution quality. Another simple possibility would be use the average and variance of a random sample as the scaling points.

4.3.3 New Problem Specific Measures

During the comparison of fitness landscape analysis methods, easily obtainable problem-specific measures have been included for comparison. For both the vehicle routing problem and the quadratic assignment problem some simple statistics have been used successfully as complementary measures for hardness prediction and problem similarity characterization.

For the vehicle routing problem, a simple clustering index was used, as this property is informally often described in the literature. This index is calculated by repeatedly by creating a k-means clustering (see Lloyd, 1982) with k varying between one and the number of cities. The clustering with the best quality is then divided by the number of cities to give a clustering index. The quality of the clustering is defined as the minimum distance between clusters minus the average distance inside the clusters.

Moreover, some basic statistical properties of the distance and demand matrix such as average and standard deviation have been included after normalizing by projecting the city coordinates into the unit square and dividing the demands by the vehicle capacity.

In addition, it was tried to formalize the eccentricity of the depot as a factor that is visually apparent. Several different methods were formulated, to either obtain the geographic eccentricity, the distance weighted centroid eccentricity, or the distance and demand weighted centroid eccentricity, which are all formulated by first calculating a center location and then measuring the distance of the depot to this center location. However, all these measures correlate very strongly with each other and it suffices to use only one of them in practice.

- The geographic center is defined as the mid-point inside the range of all cities. If the cities are reprojected to the unit square this will always be at $(0.5, 0.5)$ except for degenerate cases with all cities along the same axis.
- The distance-based centroid is calculated by simply averaging the coordinates of all cities.
- The distance and demand-based centroid is calculated as the weighted average of all coordinates, where each coordinate is weighted by the relative demand.

Also for the quadratic assignment problem, several new measures where defined as detailed in (Pitzer et al., 2013a). On the one hand, simple statistics, such as mean, average, skewness, kurtosis and normality (see e.g. Jarque and Bera, 1980), for both the distance matrix and the flow matrix have been used. On the other hand, several compound characteristics have been calculated:

- For both of the two matrices a measure for sparsity was introduced which is simply the ratio of zeros within each an $n \times n$ matrix (a_{ij}) i.e. $|\{a_{ij}|a_{ij} = 0\}|/n^2$.
- Also, two measures describing the asymmetry of the matrices were used. On the one hand the average difference between diametrical entries, $|a_{ij} - a_{ji}|$ as well as the ratio of such pairs with non-zero difference.
- Another interesting property that could be measured was the intransitivity, where for each triple the extent to which the triangle inequality was violated was estimated i.e. $\max(0, a_{ik} - (a_{ij} + a_{jk}))$. Then again the intransitivity sum and ratio of non-zero triples was used as another problem-specific characterization.

- Finally, the throughput across graph nodes, was measured for both matrices. These measures turned out to be particularly useful for the prediction of problem hardness as documented in Section 5.4.2. Equations 4.30 through 4.33 show the definitions of *in flow*, *out flow*, *flow sum* and *flow surplus* respectively.

$$a_{i\_} := \sum_i a_{ij} \tag{4.30}$$

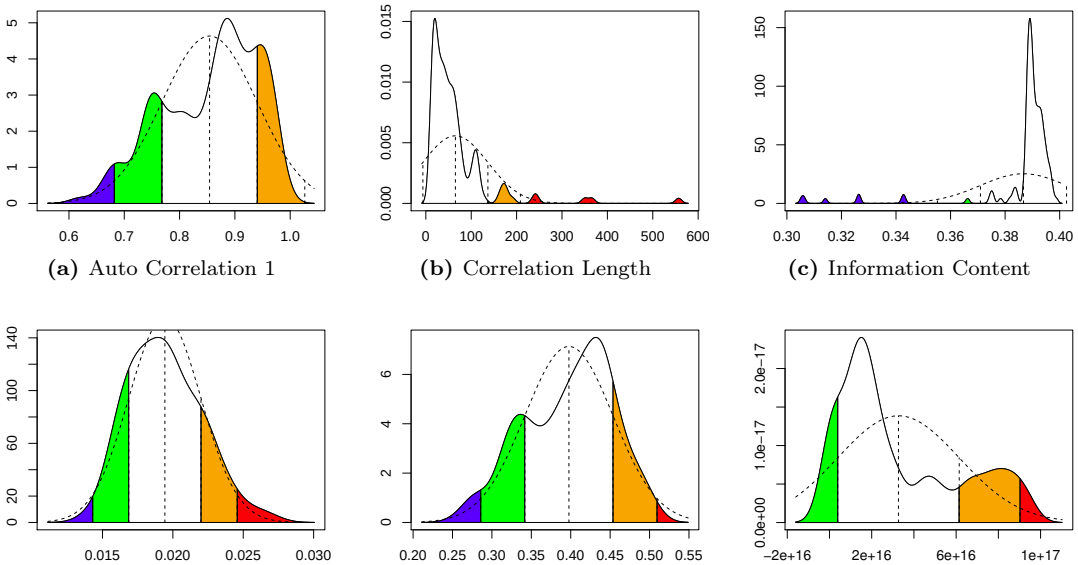
$$a_{\_j} := \sum_j a_{ij} \tag{4.31}$$

$$a_k := a_{k\_} + a_{\_k} \tag{4.32}$$

$$a_k^+ := a_{k\_} - a_{\_k} \tag{4.33}$$

4.4 Problem Instance Characterization

Studying the wealth of information generated through the analysis suite can be a daunting task. At first, there is no real frame of reference so a certain value does not really make sense on its own. Therefore, the first few steps were to gain some understanding on the distributions of all these analysis results. To do so, these distributions were examined in isolation. Figure 4.16 shows some examples from the analysis of the quadratic assignment problem library (QAPLIB). Average values are less than one standard deviation away from the mean, high and low values are more than one but less than two standard deviations away from the mean and very high and very low are more than two standard deviations away from the mean.



(a) Auto Correlation 1 **(b)** Correlation Length **(c)** Information Content

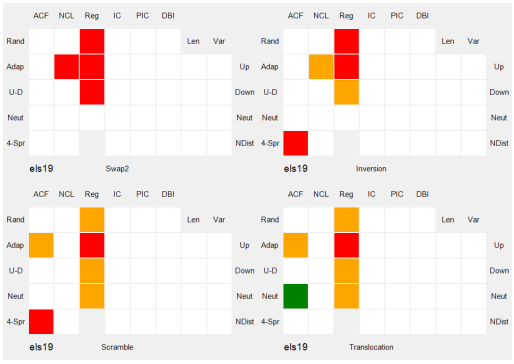
(d) Auto Correlation Quantile Spread **(e)** Correlation Length Quantile Spread **(f)** Up Walk Length

Figure 4.16. Distribution Analysis across the Quadratic Assignment Problem Library: While the superimposed normal distribution is only sometimes a good fit, the distributions establish a frame of references. With this reference frame, individual values can be classified as average (white), high (orange), very high (red), low (green) and very low (blue).

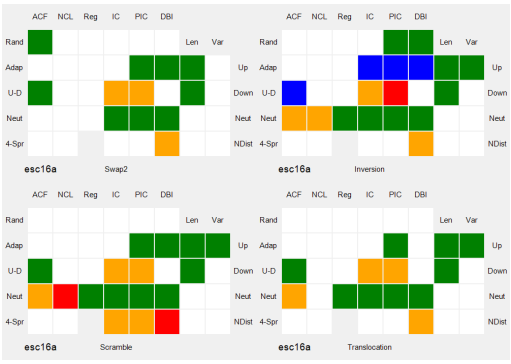
4.4.1 Box Charts

The classification used with respect to a large scale study enables us to talk about the properties of fitness landscapes in more familiar terms like high auto correlation or low information content. The next step was to order and visually group these pieces of information to provide a quick overview, similar to a patient chart at the foot of a patient's bed in a hospital, a researcher can quickly gauge the characteristics of the problem instance. Figure 4.17 shows a few examples of these box charts that color code the fitness landscape analysis values with respect to the whole problem class.

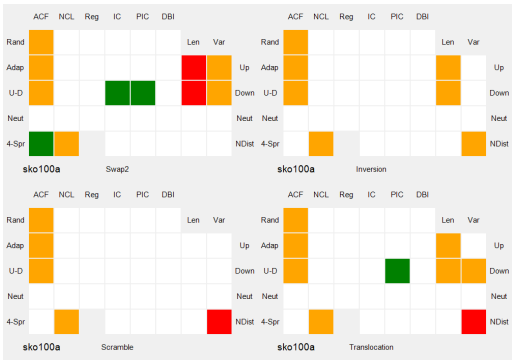
(a) bur26a



(b) els19



(c) esc16a



(d) sko100a

Figure 4.17. Box Chart Summary of Several Instance of the QAPLIB. Each problem instance was analyzed using different neighborhoods, or in this case, different mutation operators, each of which defines its own landscape variant. However, with respected to an appropriate frame of reference the categorization is similar for all variants. Table 4.4 lists all the abbreviations used in the box charts. The rows are measurements from different sampling trajectories while the columns are different measurements. the smaller right hand section is a little more complicated, where for the neutral and up-down walks average lengths and variances are shown. The rightmost bottom two values, finally, are the length and variance of neutral walk distances.

These charts can provide a first insight in the diagnosis of an algorithm application. If an algorithm does not perform as expected for a certain problem instance, markedly high or low values can be quickly identified, which can help to speculate over reasons for the performance problems. However, speculating over possible landscape characteristics and guessing correct algorithm parameterization by manual interpretation of these charts is rather difficult and can sometimes lead the right way but at other times be completely wrong. Therefore, it could be seen as a fitness landscape horoscope only comprehensible by the adepts.

Table 4.4. Abbreviations Used in the Box-Charts

Code	Explanation
ACF	auto correlation
NCL	normalized correlation length
Reg	regularity
IC	information content
PIC	partial information content
DBI	density basin information
Rand	random walk
Adap	adaptive walk
U-D	up-down walk
Neut	neutral walk
4-Spr	repeated multi trajectory walk quartile spread (25-75)
Len	length
Var	variance
Up	upward sub-walk
Down	downward sub-walk
NDist	neutral walk distance

4.4.2 Hardness

Problem hardness estimation is not an easy task. Most importantly, there are two aspects to problem hardness:

- On the one hand, *inherent complexity* or simply the size of the search space is typically a dominating factor and mostly apparent from the problem specification.
- On the other hand, structural properties of a problem instance can cause problems for certain optimization schemes. These properties should be discoverable through fitness landscape analysis, but pose different levels of difficulty for different algorithms.

One could imaging this inherent complexity as increase in the information necessary to progress through the landscape and also the ability to see this progress. A very simple problem instance will have direct correlation with fitness and progress towards the optimum with no deceptive local optima in between. As the inherent structure decreases, epistasis between the components starts to manifest, where changes in one dimension cause unforeseen changes in the structure of other dimensions. Finally, for very hard problems, internal structure breaks down completely, and the landscape becomes a random table with no structure that could be used to guide the optimization process.

4.4.2.1 Hardness Measurement

On the target side of problem characterization, the performance of an algorithm has to be examined when applied to a certain problem instance. Very often, algorithms are compared competitively against each other. Numerous research papers justify the introduction of a new methods by pitting the new algorithm against an established one. Therefore, the most prominent algorithm performance measure is relative.

The reason for this frequent choice is the difficulty of other, more absolute terms. In general any algorithm application to a problem instance, leaving all parameter choice and implementation details aside, still hast at least two performance indicators. These are

- *convergence speed* and
- *solution quality.*

Ideally, these two factors should be combined into a single indicator that simply states how good an algorithm performs. However, due to different convergence speeds in earlier and later stages of different algorithms a single performance value is not possible. Therefore, in the following sections several alternative methods for estimating a problem's hardness for different algorithms have been used.

One possibility is to allow very long run-times for each algorithm so that most configurations will be able to find the best known solution. In this case, the final quality becomes irrelevant and the algorithm performance can be identified with the convergence speed. For this scheme to succeed, however, every single configuration has to converge to the best known solution, or these cases have to be otherwise eliminated.

Alternatively, both factors, convergence speed and final quality, can be taken together as a weighted sum. This is a very simple solution that can be applied to any multi-objective optimization problem, where all objectives are weighted against each other as shown in Equation 4.34.

$$f(x) = \sum_i \alpha_i f_i(x) \tag{4.34}$$

In this case, the difficult question becomes the normalization of all factors and the choice of the weights which usually is deliberate but remains, in fact, arbitrary. In the analysis, most weight has been attributed to the solution quality and less weight to the convergence speed, however, in different circumstances this requirements might be reversed.

Finally, a completely fair comparison is only possible if the algorithm performances are actually bilaterally compared as is done in most studies. This is, however, only possible for a moderate number of algorithms or configurations that have to be compared with each other, as the number of comparisons increases more than exponentially. In Section 4.8, a bilateral domination analysis has been used to directly decides between two algorithms.

4.5 Problem Instance Comparison

While the analysis results from Section 4.4.1 can give a first hint to the interested researcher, they are far from hard and fast results towards algorithm selection. However, the underlying numbers can be used as a first handle along this way.

Most of the fitness landscape analysis results are simple scalars that can be combined into a vector of characteristics for each problem instance. These vectors can then be seen as projections of the vast problem instance space to the simpler and measurable fitness landscape space. The more analysis results are available, the higher the chance to discern between different aspects of problem instances. However, at the same time, many different analysis methods yield only slightly different perspectives of the same properties. Therefore, care has to be taken to not overemphasize aspects which are easier to measure in comparison to other aspects, where maybe only a single measure exists.

4.5.1 Distance Calculations

With the help of a fitness landscape description vector, the differences between problem instances can be quantified by analyzing the distances between these vectors. In a first attempt, problem similarity was modeled as the distance between fitness landscape features.

An important aspect for the distance calculation is the emphasis that different components of the fitness landscape vectors receive in relation to each other. For this reason, the components were first subjected to a correlation analysis with each other. Very highly correlated values were

simply removed. Most obviously, values which corresponded exactly have always been removed. Moreover, variables with both Pearson correlation above 0.9 and a Spearman's Rank Correlation above 0.9 have been removed (see e.g. Larsen and Marx, 2011). Alternatively, or in addition to simple mutual correlation analysis a principal component analysis (PCA) can be performed to to obtain the right "view" onto the space of feature vectors (see Jolliffe, 2002).

4.5.2 Clustering

A first showcase for the validity of the claims that fitness landscape analysis can elucidate the inner structure and relationship between optimization problems was obtained using cluster analysis that uses the distances between the fitness landscape feature vectors to produce a genealogical analysis and, therefore, establish a larger neighborhood or grouping of different problem instances.

An example of a cluster analysis using smaller instances form the Quadratic Assignment Problem Library is shown in Figure 4.18. The heat map encodes the Euclidian distances between the feature vectors. These distance have been used to perform a clustering using the Ward's cluster analysis (see Ward, 1963).

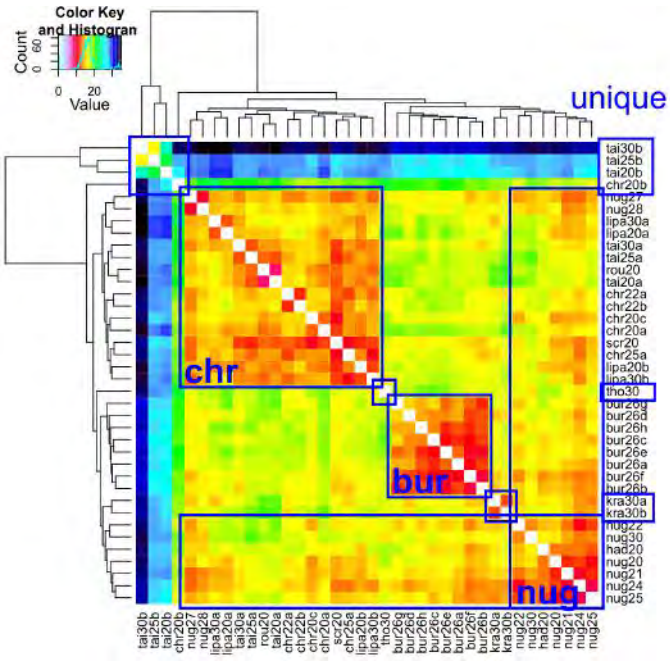


Figure 4.18. Clustering of Small QAPLIB Instances: The problem instances come from different domains or generators indicated by the names. Interestingly, the results of clustering seem to be able to discern between these different problem groups and cluster them back together.

4.5.3 Visualization

Other methods for the visualization of the obtained distances between problem instances can be obtained using, for example, principal components analysis (PCA) (see Jolliffe, 2002) or neighborhood component analysis (NCA) (see Goldberger et al., 2005). While PCA uses only the extent of variation to obtain the right perspective, NCA requires a labeling to come up with the right cross sectional view. An unsupervised method for reducing highly dimensional data could be unsuper-

vised clustering before the neighborhood component analysis as has been done by my colleague Andreas Beham in (Pitzer et al., 2013b) with insightful results.

A more direct approach is presented in Figure 4.19, where all collected features from the analysis suite in Section 5.3 are use to create a two dimensional self organizing map (see Kohonen, 1982). This approach directly organizes the problem instances according to their multidimensional distances. Therefore, problem instances with low distances should be still near to each other in the map. As can be seen in this Figure, the characteristics that stem from the problem domains or generators are nicely reflected in the distances as obtained through fitness landscape analysis. The color-coded names on the left hand side correspond to the problem generators while the symbols on the right hand side denote problem variants (if any).

This map is solely based on the features derived from fitness landscape analysis. As shown in Figure 4.20 these features help to separate different problem instances based on their properties.

These promising results show that some inherent characteristics of each problem instances can be extracted successfully using fitness landscape analysis. However, using these insights, it can be hoped that these features do not only reveal properties pertaining to problem generators and variants but can also be used for the prediction of more practical properties such as problem hardness for a certain algorithm or the differentiation between suitable algorithms or parameter settings.

4.6 Algorithm and Parameter Selection

Using the same kind of data as for the visual problem instance characterization, it should be possible to gain even more insights. For this reasons, an algorithm testbed has been prepared, where a whole array of algorithm configurations has been applied to a collection of problem instances (see Pitzer et al., 2012a). In this work, the quadratic assignment problem library (QAPLIB) was used again as a collection of test instances and a parameter grid of different algorithm configurations was tested.

4.6.1 Algorithm Performance Measurement

On the one side, there are the same fitness landscape analysis results as in the previous sections, this time, confronted on the other side with a grid of algorithm configurations that were applied to each problem instance. So, for each problem instance, in the set of given configurations, some will work better than others. Figure 4.21 shows several examples of such a parameter landscape. These optimization experiments have been set up and conducted together with my colleague Andreas Beham.

In this case, the robust taboo search (Taillard, 1991), an extension by Taillard to the Tabu Search (Glover and Laguna, 1997), that is especially suited for the quadratic assignment problem has been examined in detail. For the two parameters, maximum taboo tenure and alternative aspiration tenure, 90 different parameter configurations have been tried. Each parameter setting has been tried 30 times. This corpus of experiments has then been used to examine the power of fitness landscape analysis to predict good parameter configurations.

As an important side note, the robust taboo search has been specifically designed for the quadratic assignment problem and performs exceptionally well. Therefore, the performance variations between different configurations for a certain problem instance are not very large and an average configuration performs already very good. Moreover, for some instances, any configuration works quite well, while for others no configuration works really good, which leaves very little room for improvement. However, keeping these restrictions in mind, the performance of fitness landscape analysis predicting suitable configurations is acceptable.

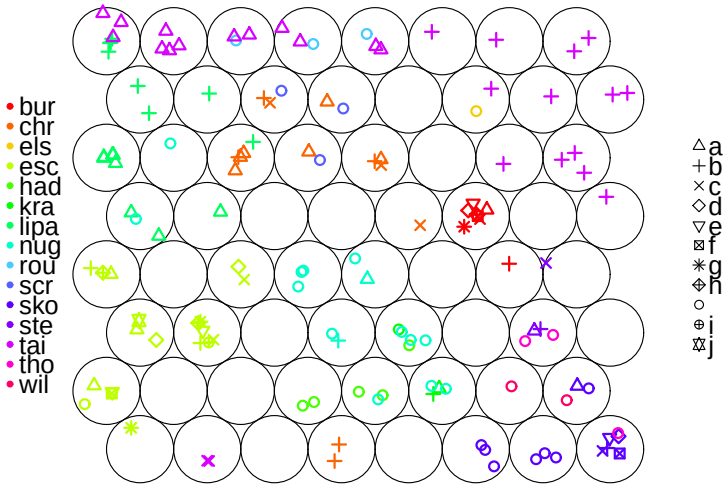
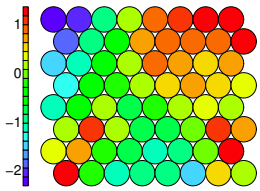
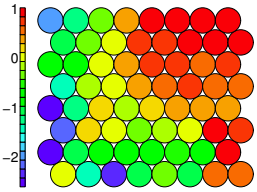


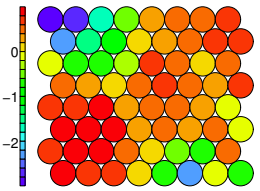
Figure 4.19. Kohonen Mapping of Fitness Landscape Analysis Features: The distances between the feature vectors have been projected to two dimensions using a self organizing Kohnen map. The different problem generators and their subtypes seem to have great influence on the measurable properties which enables the map to group them accordingly.



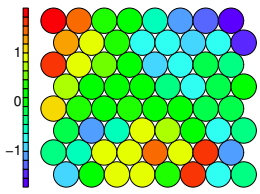
(a) Random Auto Correlation



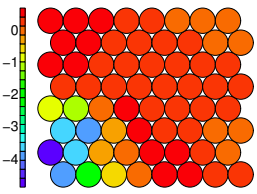
(b) Up/Down Auto Correlation



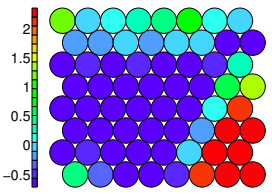
(c) Neutral Auto Correlation



(d) Auto Correlation Isotropy

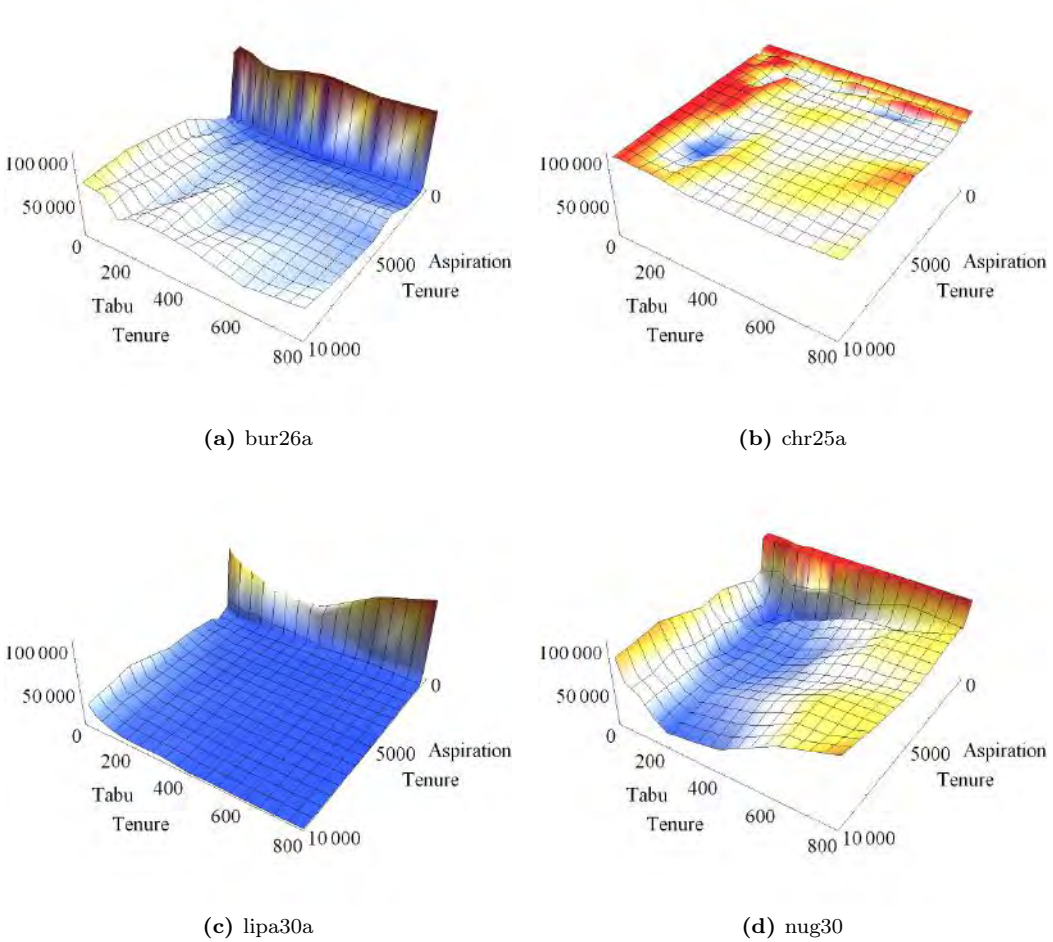


(e) Information Content



(f) Regularity

Figure 4.20. Kohnen Map Components: Several fitness landscape analysis features as remapped on the Kohnen map.



(a) bur26a **(b)** chr25a

(c) lipa30a **(d)** nug30

Figure 4.21. Parameter Landscapes: These surfaces show the average number of iterations for different parameter settings. Each diagram visualizes 90 different parameter settings for each problem instance. While the landscapes themselves look quite different to each other, it is relatively apparent that "average" settings in the center of the parameter landscapes are relatively good in any case.

4.6.2 Robust Taboo Search Configuration Grid

The robust taboo search uses a random variable for the description of the tabu list length that is, therefore, not fixed any more. In addition a forced diversification step is introduced using another random variable, where a solution or a *move* is made that has not been seen for a certain number of steps. These are the only parameters of the robust taboo search. The configurations that have been tried are shown in Table 4.5. The possibilities were later extended as it was discovered that good parameter settings seemed to lie between the values that were tried previously, as shown in bold in Table 4.5.

Table 4.5. Robust Taboo Search Configuration Settings: The entries marked in bold were added in later experiments.

Parameter	Values
tabu tenure	25, 50, 100, **150**, 200, **300**, 400, **600**, 800
alternative aspiration	100, 500, **750**, 1000, **1500**, 2000, **3500**, 5000, **7500**, 10000

4.6.3 Neighbor Selection

One the of the simplest regimes to select a suitable parameter configuration is what an algorithm expert would probably do: The expert would consult his expertise on similar problem instances and choose a setting that is an average of settings that worked in the past. This can be implemented, for example, with a k-nearest neighbor algorithm (see Shakhnarovich et al., 2005). While this approach is very simple, for the proposed analysis suite it contains some difficulties as nearest neighbor methods suffer from the "curse of dimensionality" (see Beyer and Schwefel, 2002): As the number of dimensions increases the method becomes increasingly sensitive to small variations and, hence, becomes *unstable*.

This problem can be somewhat mitigated by selected a larger number of considered neighbors. However, in this case the choice of algorithm parameters will be less specific. As a first attempt towards automatic parameter selection, however, it shows promising results. In Table 4.6, a short summary is presented that compares the total number iterations necessary for predicting some instances of the QAPLIB using different methods:

- The baseline is to use one configuration that works well on average for all instances.
- The other extreme to use the best possible configuration for each instance individually. Of course, this choice can only be made a-posteriori.
- Another test was to use a simple "prediction" scheme with the problem size as discriminant. Or in other words, the average-best configuration for each problem size.
- Finally, this is compared to using k-nearest neighbors to select similar problem instances and build average parameters. The best results were achieved using six neighbors and an average over their eight best configurations for the small configuration grid, where only 36 different configurations were tried as shown in Table 4.6a.

In this case the bandwidth of performance ranged between two fictitious points, both of which are only achievable a-posteriori, when all experiments have been performed. However, the average best configuration does not seem to be very far fetched. Using a single representative configuration results in a 39% reduction in performance compared to the best possible a-posteriori choice for each scenario. On the other hand using fitness landscape analysis and a simple nearest neighbor selection of configuration can go half the way towards the best possible solution and is only about 23% percent away from from the best possible choice.

Table 4.6b shows another experiment, where a finer parameter grid was used. Again, using a single, average-best configuration is about 50% worse than the a-posteriori best solution, and again, the k-nearest neighbor selection scheme comes about half the way towards the best possible parameter choice at only 27% worse than the best a-posteriori algorithm choice. This time, a larger number of neighbors but a smaller number of best configurations was used, namely eleven neighbors and only their respective single best configurations were considered. Moreover, another improvement was achieved by a cut-off distance, where, in extreme cases, very far neighbors were disregarded and so, effectively fewer than eleven neighbors were used for the more obscure problem instances.

Table 4.6. Performance of Different Configuration Selection Schemes

(a) Coarse Parameter Grid

Scheme	Average Number of Iterations	Relative Difference
best	16477.37	-
knn	20295.77	23.2%
average	22872.5	38.8%
size-only	24377	47.9%

(b) Finer Parameter Grid

Scheme	Average Number of Iterations	Relative Difference
best	14735.22	-
knn	18968.71	28.7%
average	22156.19	50.4%

While these results are not perfect, they are encouraging, especially with respect to the fact that robust taboo search is very well tuned for the quadratic assignment problem and for many problem instances it either works well or not regardless of parameter settings. Therefore, one could argue that this test is not very suited to evaluated the power of fitness landscape analysis methods. However, in the subsequent sections, other approaches are described as well.

4.7 Problem Hardness Estimation

As stated in Section 4.4.2.1, the objective measurement of problem hardness is a non-trivial problem. In general, problem hardness can only be measured with respect to a certain algorithm configuration. In this case, it was attempted to model problem hardness as the average hardness over a whole array of different configurations. With an extensive experiment array at hand as described in Section 4.6.1 it is possible to use an average over the collected performances of all configurations as a rough estimate of the problem hardness with respect to robust taboo search.

At first, the weighting between convergence speed and solution quality has to be decided. As stated earlier, most algorithm configurations converge to the best known solution so it can be assumed that mostly only the number of iterations are relevant. Nevertheless, for those cases, where convergence is not achieved, these problems have to be seen as extraordinarily hard. Therefore, the weights between quality and number of iterations have been, arbitrarily, assigned as ten to one, shown in Equation 4.35, where problem hardness $h(p)$ of problem instance p is described in terms of solution quality $q(p)$ and number of iterations $it(p)$ which has to be normalized with respect to the maximum number of allowed iterations.

$$h(p) := \frac{9}{10} \cdot q(p) + \frac{1}{10} \cdot \frac{it(p)}{it_{\max}} \tag{4.35}$$

Moreover, quality itself has to be put in relative terms. A measure that was deemed appropriate to compare algorithm performance between different problem instances is the *scaled difference* to the best known solution. In many cases in the literature, algorithm performance in only compared relative to the same problem instances, so the need for quality scaling does not arise. Here, however, the achieved quality is compared in relation, not only to the best known solution (q/q_{best}) but also with respect to the average quality that would be achieved by a random solution. Therefore, scaled difference results in a measure between zero and one, where zero is the best known solution and one is the quality achieved by a random solution on average as shown in Equation 4.36. This also means that problems, where a random solution is already quite close to the optimum are perceived more difficult as the scaled difference is a measure of progress from an average solution candidate towards the best known solution.

$$sd(p) := \frac{|q(p) - q_{\text{best}}(p)|}{|q_{\text{avg}}(p) - q_{\text{best}(p)}|} \tag{4.36}$$

As a final note on solution quality and the derived problem hardness, please note that $q(p)$ or $sd(p)$ are actually stochastic, and, therefore, different between different trails.

Now, several different measures pertaining to a problem instance can be constructed. While the previous quality measures have all been obtained for a single application of one algorithm configuration to one problem instance, a single measure per problem instance over all configuration is desired. Therefore, the first step is to calculate averages over repeated runs of the same configuration. From there on, there are several measures of problem instance hardness as follows:

- The *average hardness* $\overline{h}(p)$ is probably the most straightforward measure, which also best captures what would be called the inherent general problem hardness and is the average quality achieved over all algorithm configurations for a single problem instance.
- Moreover, the *best hardness* $h_{\text{best}}(p)$, while sort of an oxymoron, describes the best quality that was achieved for any algorithm configuration. This value can be seen as the global optimum of a parameter landscape as shown in Figure 4.21.
- Finally, another interesting problem hardness characteristics can be derived, the *hardness specificity*. This values describes the potential of selecting the correct algorithm configuration and can be defined as $s(p) = \overline{h}(p) - h_{\text{best}}(p)$. A problem instance with a high hardness specificity needs more careful parameter tuning than a low specificity problem which has more or less the same hardness regardless of parameter settings.

In summary, the values defined Equations 4.37, 4.38, 4.39, and 4.40 have been used, which are based on the quality $q(p, c, i)$ measured for each problem instance p, configuration c, and repetition i. Similarly, the number of iterations it has been measured. Then, the average hardness is the total average over all configurations and repetitions, the best hardness is the minimum scaled difference over all configurations, averaged only over the repetitions and the hardness specificity is their difference for every problem instance. Here, n is the number of configurations and m is the number of repeated runs of each configuration.

$$h(p, c, i) := \frac{9}{10}\frac{|q(p, c, i) - q_{\text{best}}(p)|}{|q_{\text{avg}}(p) - q_{\text{best}}|} + \frac{1}{10}\frac{it(p, c, i)}{it_{\max}} \tag{4.37}$$

$$\overline{h}(p) := \frac{1}{n \cdot m}\sum_{i,c} h(p, c, i) \tag{4.38}$$

$$h_{\text{best}}(p) := \min_{c} \frac{1}{n}\sum_{i} h(p, c, i) \tag{4.39}$$

$$s(p) := \overline{h}(p) - h_{\text{best}}(p) \tag{4.40}$$

4.7.1 Hardness Prediction using Fitness Landscape Analysis

One successful use of fitness landscape analysis results was reported in our previous work in (Pitzer et al., 2012a), where problem hardness was estimated using the same parameter grid as detailed in the previous sections. However, instead of selecting a suitable configuration, based on the overall results of different configurations, a prediction of general problem hardness was attempted. In addition to robust taboo search, also Genetic Algorithms (see Holland, 1975) and Simulated Annealing (see Kirkpatrick et al., 1983) were used and a similar parameter grid based on initial experiments was tested.

In an initial search for direct correlations between problem hardness indicators and fitness landscape values, only moderate success was achieved. Even though, the original intent of many fitness landscape analysis measures is to directly measure an aspect related to problem hardness, this seems to be not true in general. Table 4.7 shows a summary of the most successful correlations as found in (Pitzer et al., 2012a).

Table 4.7. Best Direct Correlations of FLA with Average Hardness

Variable	r^2
Up-Down Auto Correlation Coefficient	0.59
Random Auto Correlation Coefficient	0.59
Problem Size	0.56
Up-Dow Down Walk Length	0.44
Up-Down Correlation Length	0.41
Up-Down Up Walk Length	0.41
Up-Down Information Content	0.36

Different versions of the auto correlation coefficients are only marginally better than the simplest predictors for problem hardness: *problem size*. Figure 4.22 shows a scatter plot of problem size against problem hardness. This correlation is almost to the best possible direct correlation between fitness landscape analysis measures and problem hardness. Interestingly, there seem to be two groups of problems with different correlation coefficients in Figure 4.22: One group of problem instances lies more or less along the main regression line, while the other group has significantly higher hardness but is still proportional to problem size. Hopefully, the difference between these two groups will become visible and better separable using several fitness landscape analysis measures in concert.

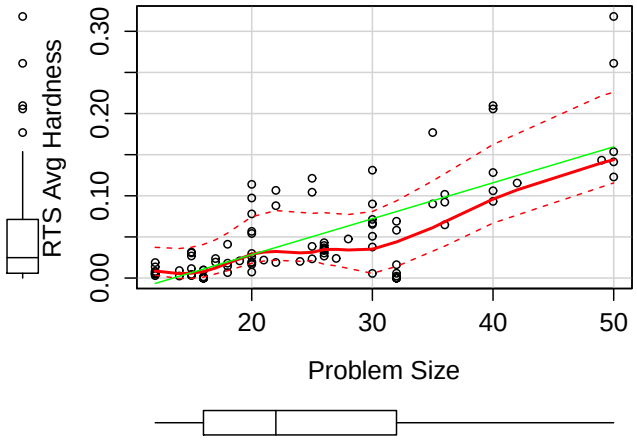


Figure 4.22. Problem Hardness vs. Problem Size: One of the best simple predictors for problem hardness is problem size.

As a side note, several FLA results were found to correlate with each other, which is not surprising as most of them analyze exactly the same data, namely the sequence of fitness values of the various walk types. In particular, it was found that (partial) information content correlates with auto correlation, and auto correlation correlates with problem size and correlation lengths. The most significant correlations or interferences between FLA values measured along random walks are summarized in Table 4.8.

Table 4.8. Overlaps of Fitness Landscape Analysis Values: While these overlaps or not really surprising, redundant values could later hinder prediction performance.

Variable 1	Variable 2	r^2
Random Information Content	Random Partial Information Content	0.97
Random Auto Correlation Coefficient	Problem Size	0.91
Random Density Basin Information	Random Information Content	0.90
Random Auto Correlation	Random Auto Correlation Coefficient	0.87
Random Density basin Information	Random Partial Information Content	0.87
Random Auto Correlation	Problem Size	0.77
Random Auto Correlation	Random Correlation Length	0.68
Random Correlation Length	Random Auto Correlation Coefficient	0.66
Random Correlation Length	Problem Size	0.54

While the direct correlation between fitness landscape measures and problem hardness was a little disappointing, this is not the end of the story. Instead, our assumption is that the combination of complementary measures might give a better insight into the problem structure and, therefore, allow a more successful modeling of problem hardness.

For this purpose, the least absolute shrinkage and selection operator (LASSO) described in (Tibshirani, 1996) and the implementation of elastic nets in R (see Friedman et al., 2010; Simon et al., 2011) were used. In this method, not only the error is minimized but also the extents of the coefficients. Therefore, this method can be used to implicitly perform variable selection and it helps to create a model with better generalization properties.

Different variable pre-selections were tried to determine which landscape variant and which walk type are the most informative. Moreover, the correlation threshold between input variables was varied. As detailed in the previous paragraph many fitness landscape analysis variables are strongly correlated to each other. Even though an automatic variable selection was employed, it is advisable to reduce the number of exactly or almost exactly equal input variables as can been seen from the results.

Detailed prediction models can be found in the results chapter in Section 5.4. A summary of the best regression models' variables and performances is shown in Table 4.9, which was already reported in (Pitzer et al., 2012a). The corresponding scatter plot of predicted vs. actual problem hardness values is shown in Figure 4.23.

In summary, the predictive power of fitness landscape analysis values for the forecast of problem hardness seems adequate. Many classical fitness landscape analysis values are useful in the prediction model. Moreover, the detailed analysis results of up-down walks seem to have payed of as well, as these measurements are also included in many prediction models. In the following section the inclusion of additional complementary measures is examined, in order to increase the prediction accuracy.

4.7.2 Combination with Problem-Specific Measures

In a previous paper (see Pitzer et al., 2012b), the cross correlation between problem specific measures and fitness landscape analysis measures was investigated. While there might be some overlap, these two classes of problem instance descriptors have very different nature and it can be expected that they complement each other well.

Table 4.9. Variables used in the Best Regression Models for Average Hardness: The achieved correlation between predicted and actual hardness was measured using cross validation and shows promising results.

	Measure	Random	Up-Down	Neutral	All
	Problem Size	☑	☑	☑	☑
Random	Normalized Auto Correlation	☑	☐	☐	☑
Random	Normalized Correlation Len	☑	☐	☐	☐
Random	Density Basin Information	☐	☐	☐	☑
Random	Partial Information Content	☑	☐	☐	☐
Up-Down	Auto Correlation	☐	☑	☐	☐
Up-Down	Normalized Auto Correlation	☐	☐	☐	☑
Up-Down	Correlation Length	☐	☐	☐	☑
Up-Down	Normalized Correlation Length	☐	☑	☐	☐
Up-Down	Density Basin Information	☐	☑	☐	☑
Up-Down	Regularity	☐	☑	☐	☑
Up-Down	Up Walk Length	☐	☑	☐	☐
Up-Down	Up Walk Length Variance	☐	☐	☐	☑
Up-Down	Down Walk Length Variance	☐	☑	☐	☑
Up-Down	Lower Level Variance	☐	☐	☐	☑
Neutral	Auto Correlation	☐	☐	☑	☐
Neutral	Correlation Length	☐	☐	☑	☑
Neutral	Density Basin Information	☐	☐	☑	☐
Neutral	Information Stability	☐	☐	☑	☐
Neutral	Regularity	☐	☐	☑	☑
Neutral	Average Walk Distance	☐	☐	☑	☑
Neutral	Average Walk Length Variance	☐	☐	☑	☑
Training r^2		0.68	0.76	0.77	0.84
Cross Validated Test r^2		**0.60**	**0.67**	**0.70**	**0.74**

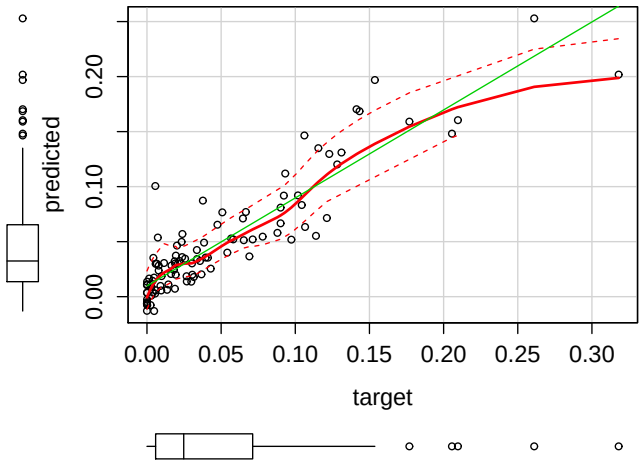


Figure 4.23. Robust Taboo Search Average Hardness Prediction on QAPLIB Instances: This regression model has an r^2 of 0.74 using 10-fold cross-validation and shows the predictive power of fitness landscape analysis values for hardness prediction.

An interesting example is shown in Figure 4.24, first published in (Pitzer et al., 2012b), where two different feature vectors were constructed to describe each problem instance. On the one hand, the previously used feature vector constructed from fitness landscape analysis measures were used, and on the other hand a similar vector using problem-specific measures for the vehicle routing problem was constructed.

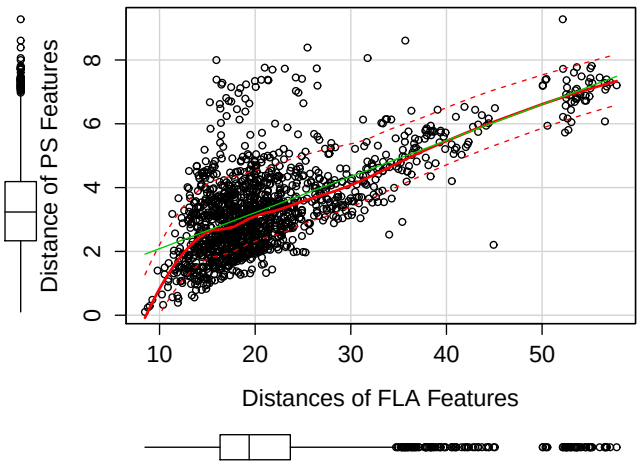


Figure 4.24. Distance Correlation of FLA and Problem-Specific Feature Vectors: Distances calculated using fitness landscape analysis measures compared to distances directly extracted using problem-specific measures have an apparent correlation for many extreme cases while the bulk of instances is rather undecided.

While the correlation is not perfect, the distances described by the problem-specific measures can be "understood" or related to by fitness landscape analysis measures. On the other hand, there seem to be some instances, in the top left corner, where problem-specific measures predict large differences which cannot be easily perceived using only fitness landscape analysis.

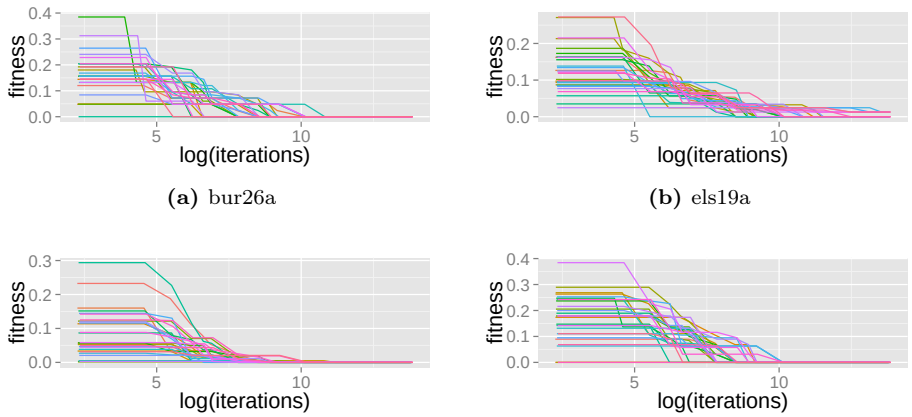
Therefore, while fitness landscape analysis seems to be on a good track to capture a lot of information that could be extracted using problem specific information, some parts are still not covered. In remains to be seen whether FLA will be able to also measure or relate to all problem specific values. However, for now, it seems that the combination of both should be beneficial, as there are some aspects, at least of vehicle routing problems, that are not yet captured using fitness landscape analysis methods. Therefore, in (Pitzer et al., 2013a), both aspects have been combined, using both fitness landscape analysis methods together with problem specific values for the prediction of problem hardness. The combined results are very promising, as shown in Section 5.4.2.

4.8 Algorithm Dominance Estimation

Another approach to algorithm selection is algorithm domination prediction. Instead of selecting one of several algorithms up front, the choice is simplified to predict only which of two algorithms will be better. While this involves training a larger number of classifiers, it can help to get a better picture of the concrete situations where an algorithm dominates. A summary of this work can also be found in (Pitzer et al., 2013b), where, in addition to the Robust Tabu Search, another very successful algorithm for the quadratic assignment problem, the variable neighborhood search (see Hansen et al., 2010), was used.

The important question here is: "When does an algorithm truly dominate another?" In many previous works we are made to believe that one algorithm is better than another if it is closer to the best known solution after a certain number of iterations. Unfortunately, this is only half of the story. Typically, an optimization algorithm improves over time, however, different strategies have different convergence speeds at different times. One algorithm, might be very exploitative in the beginning, and converge fast at first, but due to lost diversity might have difficulties later on, while another one, deliberately converges slowly at first but has the necessary diversity later on to converge faster. The key point is that either approach might be the right one in certain circumstances. Therefore, it cannot be decided in general which of these strategies is preferable until the available computing resources and allowed computation times are known.

An additional complication is the stochasticity of many metaheuristics. This means, that convergence speed is actually a probability distribution that should be considered. Not only the average algorithm performance is of interest but also its variance. Therefore, it depends very much on the situation, which algorithm is preferable. Figures 4.25 and 4.26 show some examples of the robust taboo search and the variable neighborhood search applied to some instances of the quadratic assignment problem.



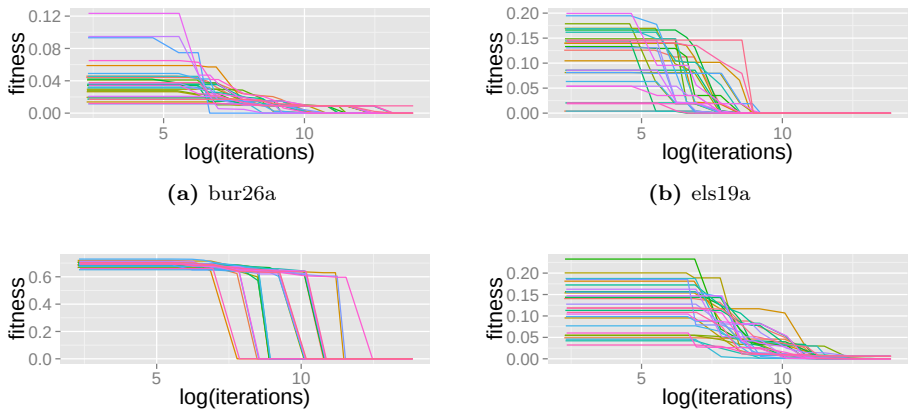
(a) bur26a **(b)** els19a **(c)** lipa40b **(d)** tai60b

Figure 4.25. Examples of Robust Taboo Search Applied to Quadratic Assignment Problems

The first step, therefore, is to determine this convergence probability distributions for both algorithms, as shown in Figures 4.27, where the mean and standard deviations of the algorithms at certain snapshots are used to succinctly describe their general convergence behaviors in comparison to each other. From these examples it seems that many times, an algorithm is always better than another one for certain problem instances. Sometimes, it is not clear whether the other algorithm has become better after further iterations. This gives rise to the idea of *significant domination*.

4.8.1 Snapshot Dominance

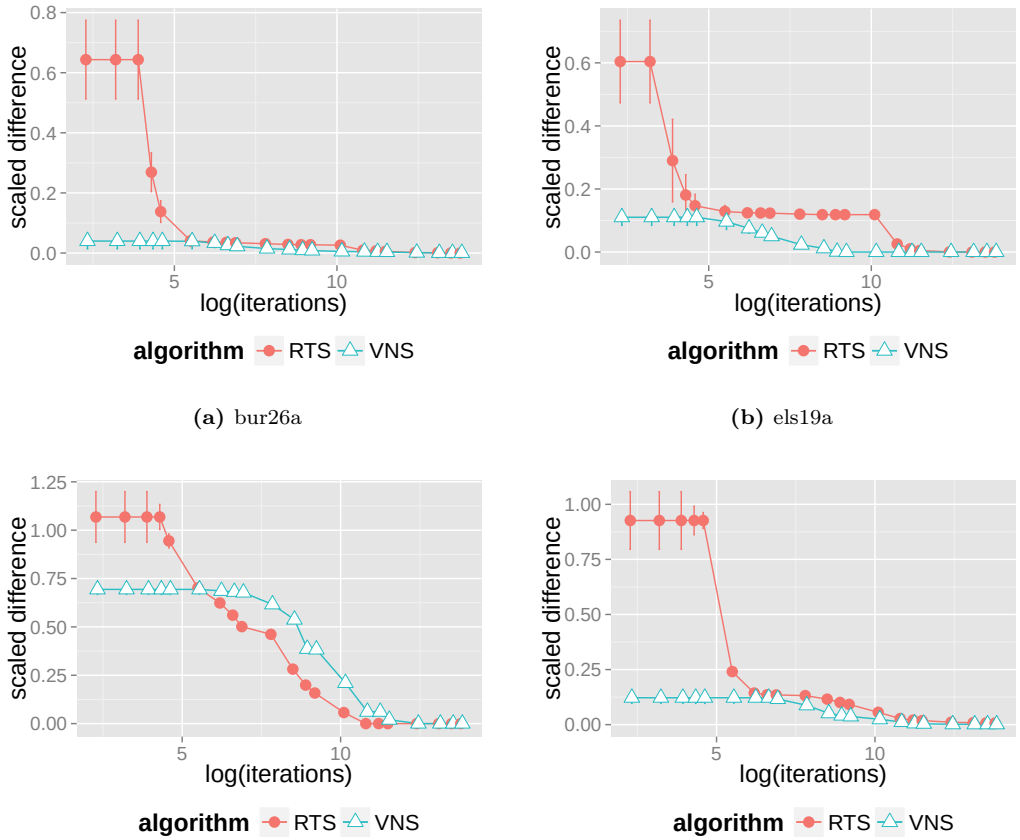
The first notion to introduce is *snapshot dominance*. For any snapshot of time or iterations, an algorithm can either be dominant, non-dominant or dominated, depending on its quality distribution at a current time snapshot regardless of past or future values. An algorithm is defined as *dominant* if its current solution quality distribution is significantly higher than that of the other algorithm, which at this point is said to be *dominated*. This significance can be determined with a simple t-statistic, using Welch's t-test (see Welch, 1947), as shown in Equation 4.41, where p_1 and p_2 are the current algorithm performances or solution qualities, an $\mathrm{E}[x]$ and $\mathrm{Var}[x]$ are the



(a) bur26a **(b)** els19a

(c) lipa40b **(d)** tai60b

Figure 4.26. Examples of Variable Neighborhood Search Applied to Quadratic Assignment Problems



(a) bur26a **(b)** els19a

(c) lipa40b **(d)** tai60b

Figure 4.27. Algorithm Convergence Distributions

expected value and variance of x, respectively.

$$t(p_1, p_2) = \frac{\mathrm{E}[p_1] - \mathrm{E}[p_2]}{\sqrt{\mathrm{Var}[p_1]/n + \mathrm{Var}[p_2]/n}} \tag{4.41}$$

If the algorithms' performances are significantly different, i.e. the t-statistics magnitude is larger than a predefined significance level, i.e. $s < |t(p_1, p_2)|$, one algorithm is dominant while the other algorithm is dominated. Otherwise, both algorithms are non-dominant as their average performances at this snapshot are not significantly different from each other. Therefore, it can not be decided unambiguously which algorithm is better.

4.8.2 Timespan Dominance

Based on the snapshot dominance, the domination history can be incorporated to come up with a recommended choice of algorithm given a predefined amount of resources, which can either be wall clock time, number of iterations or number of evaluations. For this amount, a particular algorithm that is either currently dominant or was previously dominant but has since not been dominated can now be recommended. This accounts for the fact that good algorithms are often quite close to each other in the end but can have significant differences in earlier stages. Therefore, our preferred choice is the algorithm which arrives at the higher solution quality faster, without significantly falling behind later on. Again, this dominance can be defined for any snapshot using the history of preceding snapshot dominances and defines a *domination timespan* from the time on, where one algorithm dominates until it is dominated again by the other algorithm, disregarding intermediate non-dominant phases.

4.8.3 Domination Series

Ideally, for each pair of algorithms, one algorithm would dominate all the time, so a clear choice can be made for every problem instance, disregarding the allowed computing resources altogether. Unfortunately, in practice this is not always the case. However, for a reasonably chosen significance level, it can happen quite frequently. Table 4.10 shows the processed series of domination snapshots. In this case, an algorithm is also counted as dominant if it has reached the best known solution, even if the quality difference to the other algorithm is not significant. Only if the other algorithm also reaches the best known solution, it is counted a non-dominant anymore. Nevertheless, because it converged to this level first it cannot be dominated by the other algorithm anymore (unless a new best known solution would have been found).

Table 4.10. Examples of Domination Series: A '`V`' indicates dominance of VNS while an '`R`' indicates dominance of RTS. Lower case letters are non-dominant snapshots, where the previous algorithm is kept.

Problem Instance	Domination Series	Compressed Domination Series
chr12b	`VVvvRrrrVvvvvVRrrrrrr`	`VRVR`
chr22b	`VVVVVRRRRRrrrVvvvvRRR`	`VRVR`
lipa70a	`VVVVVvRrVVVVvRrrVVVV`	`VRVRV`

As the level of significance is increased, i.e. a clearer domination is required, in many cases, the convergence series can be simplified as shown in Table 5.8 in Section 5.4. In the end only a few cases, remain in this example.

4.8.4 Dominance Prediction

After this rather complex analysis of algorithm domination, however, a good picture over which algorithm is superior at which problem instance has been obtained. In this case, the results are almost clear cut and a definitive recommendation of one algorithm over another for almost all problem instances can be extracted. If the dominance changes over time, this can be called a tie and either algorithm can be recommended.

Using this data set it can now be tried to use the fitness landscape analysis results from Section 5.3 once again to predict the dominant algorithm for each problem instance of the QAPLIB. In Section 5.4, the detailed results of different classifiers in Table 5.9 are shown.

In the end, it is not about the concrete method that comes up with the best results but rather the success of this investigation as a whole. With many different classifiers and over 80% accuracy a correct recommendation for one of two algorithms for the optimization of a particular problem instance can be made. While there is still a long way from here to completely automatic algorithm selection and parameter tuning methods, these results are very promising.

In this study, the number of evaluations was used as the prime criterion of algorithm performance. While for many problems, the evaluation of $f(x)$ is indeed the driving cost factor, different implementations also have different inherent costs. For example, improvement heuristics that do not actually evaluate $f(x)$ on a new solution candidate but rather the incremental change $\Delta f(\Delta x_i)$ that can often be evaluated much more efficiently, have a tremendous advantage compared to other algorithms. Therefore, to be fair, what counts in the end, is the concrete implementation and most often the wall clock time of the optimization procedure on a particular machine. While this makes an objective comparison even more difficult it should be kept in mind that in practical circumstances these other factors should be considered as well. Moreover, most comparisons in the literature are evaluated like in this work with the number of evaluated solutions as the primary measure of performance.

Chapter 5

Experiments & Results

5.1 Basin Analysis

The basin analysis introduced in Section 4.1 provides an efficient method to explore basins of attraction and nice visualizations for two dimensions. Figures 5.1 and 5.2 show two examples of popular test functions. While they have both been designed to be difficult by creating large numbers of optima, using the new basin analysis technique it could be seen that they are actually quite similar and that their basins are similar to each other. This means, that they have very high isotropy and an algorithm can be perfectly tuned for typical shapes of their basins.

Theses figures show the popular Ackley and Rastrigin test functions which are introduced in (Ackley, 1987) and (Törn and Zilinskas, 1989) and are shown in Equations 5.1 and 3.41 (in Section 3.7) respectively.

$$f(x) := -a \cdot \exp\left(-b \cdot \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n} \cdot \sum_{i=1}^{n} \cos(c \cdot x_i)\right) + a + \exp(1) \tag{5.1}$$

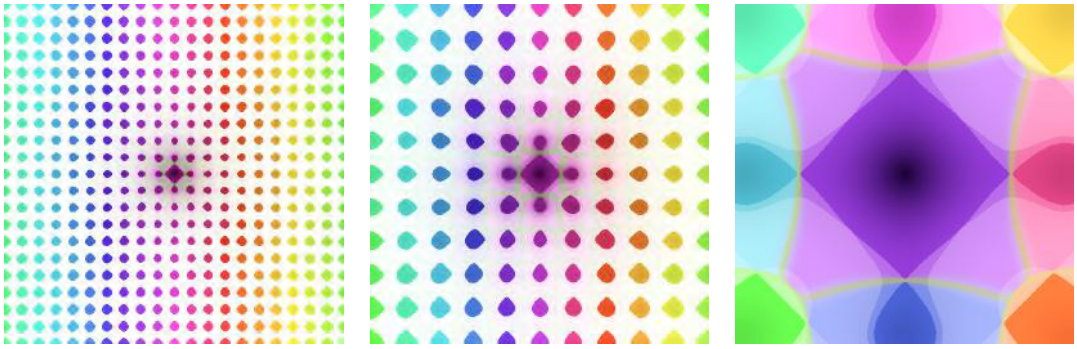
$$a = 20; b = 0.2; c = 2\pi; i = \{1, \ldots, n\}; -32.768 \leq x_i \leq 32.768$$

As described in detail in Section 4.1.4, this led to the formulation of an alternative test function that has fewer optima but higher anisotropy and, hence, is more difficult to optimize as the parameter settings cannot be tuned for a general basin of attraction but the optimization algorithm has to escape the deceptive nature of large shallow basins.

This method for basin analysis can provide an interesting first insight into optimization problems and has helped to create a better understanding of the fundamental properties of fitness landscapes such as basins of attractions, barriers and attraction volumes.

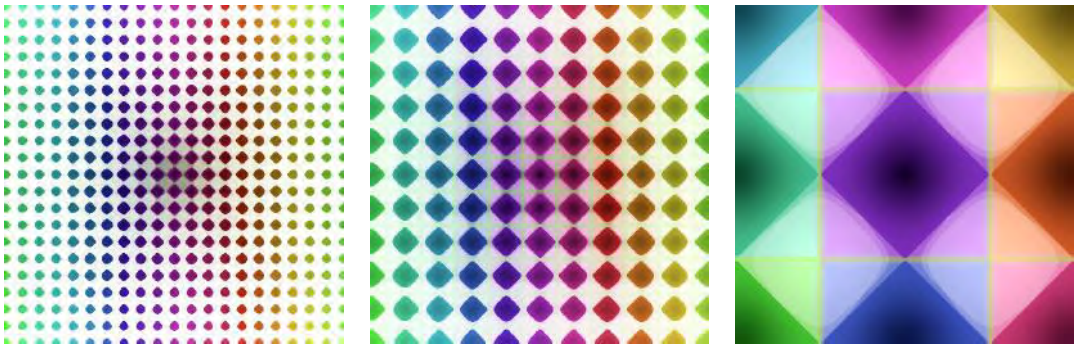
5.2 Isotropy Measures

In this section, first results obtained with isotropy measurements are examined. While they have proved to be effective for other purposes their strength for actually measuring isotropy is illustrated in this section.



(a) Extent ±10 **(b)** Extent ±5 **(c)** Extent ±1

Figure 5.1. Basin Analysis of Ackley's "Porcupine" Function with different extent sizes or zoom levels. The regularity of the shapes throughout the landscape is striking. This might be the reason why modern metaheuristics have relatively little trouble with optimization despite its high multi modality



(a) Extent ±10 **(b)** Extent ±5 **(c)** Extent ±1

Figure 5.2. Basin Analysis of the Rastrigin Function: This plot can be easily mistaken for the Ackley function from afar. However, the "tiles" of the individual basins are even more regular.

5.2.1 Correlation to Existing Measures

To validate the usefulness of the new measures, they are compared and put in relation to their existing base measures. Even though the process for obtaining multi trajectory analysis results is quite different, it is not impossible to measure something very similar to an existing result. As shown in Table 5.1 this actually happened for some of the new isotropy measures. In particular, isotropy measures based on information analysis seem to be highly correlated to existing measures and also up-down walk isotropy measures.

Table 5.1. Correlations of Isotropy Measures with existing Measures: In this table non-obvious correlations between fitness landscape analysis values are shown. Especially isotropy values based on information analysis and up-down isotropy measures have a high correlation to existing measures.

x	y	r	r^2
U/D Down Walk Len Var	U/D Up Walk Length 90% Spread	0,98	0,96
U/D Up Walk Len Var	U/D Up Walk Length 90% Spread	0,98	0,96
U/D Up Walk Length	U/D Down Walk Length 90% Spread	0,98	0,96
Rnd Information Content 90% Spread	Rnd Information Content	-0,98	0,96
U/D Down Walk Length	U/D Down Walk Length 90% Spread	0,98	0,95
Inf Peak Q Delt/Part Inf Cont 90% Spread	U/D Lower Level	0,97	0,94
U/D Correlation Length	U/D Down Walk Length	0,96	0,93
Rnd Inf Cont 90% Spread	Rnd Partial Information Content	-0,96	0,92
U/D Correlation Length	U/D Up Walk Length	0,96	0,92
Adp Information Content	Rnd Inf Val Dens Bas Inf 90% Spread	-0,96	0,92
U/D Correlation Length	U/D Down Walk Length 90% Spread	0,96	0,91
Rnd Information Content 90% Spread	Rnd Density Basin Information	-0,95	0,91
Adp Information Stability	U/D Lower Variance	0,95	0,91
U/D Information Stability	U/D Lower Variance	0,95	0,90
U/D Auto Correlation 1	U/D Partial Information Content	-0,95	0,90
Rnd Density Basin Information 90% Spread	Rnd Density Basin Information	-0,95	0,90
U/D Normalized Correlation Length	U/D Normalized Down Walk Length	0,95	0,89
U/D Down Walk Length Var	U/D Down Walk Length 90% Spread	0,95	0,89
Rnd Information Stability	U/D Lower Variance	0,95	0,89
U/D Up Walk Length Variance	U/D Down Walk Length 90% Spread	0,94	0,89
Rnd Peak Q Delt/Part Inf Cont 90% Spread	U/D Upper Level	0,94	0,88

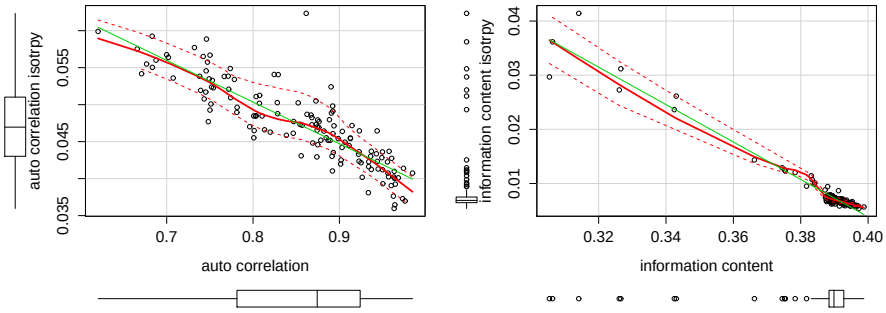
While a high correlation for some examples does not necessarily make these values useless, it has to be evaluated if these values can actually be omitted in the future. As can be seen from Figure 5.3a, the values of auto correlation and its isotropy measurement are strongly correlated but with an $r^2 = 0.757$ not nearly identical. However, it indicates, that additional normalization steps will be needed. Nevertheless, in the case here, where relatively isotropic quadratic assignment problem is analyzed, only those instances that exhibit isotropy stand out.

5.2.2 Examples

To validate isotropy, it was tested on different real vector functions that should conceptually exhibit different levels of isotropy. On the one hand, the relatively smooth but multimodal Rastrigin test function, as shown in Equation 3.41 in Section 3.7, is compared to a simple function with very high anistropy, shown in Equation 5.2. This function, while relatively smooth near the axes, is increasingly oscillating. It was actually used to produce the sample of very high ruggedness of Figure 3.4c in Section 3.3.2. This example has not only high ruggedness but also high anisotropy, as different parts of the landscape have different oscillation frequencies.

$$f(x, y) := \sin(x^y) \tag{5.2}$$

Indeed, as shown in Figure 5.4, the function of Equation 5.2 exhibits a very large amount of measurable anisotropy, in contrast to the quite smooth Rastrigin test function. This can serve



(a) Auto Correlation Isotropy **(b)** Information Content Isotropy

Figure 5.3. Existing Measures and Their Dependent Isotropy Values: While most instance of the QAPLIB show very high correlation of isotropy measures to their corresponding base measures, some special cases such as the nearly flat `esc` or `tai` instances stand out.

as a first validation of the introduced anisotropy measure. Another interesting aspect shown in this figure is the influence of the walk length on the isotropy measurement in this case. As the individual walk lengths increase the anisotropy can be measured with more confidence. This can be attributed to the more robust samples at each of the points. Moreover, as in real vector spaces an infinite number of different sample points are available, a greater insight into the local structure can be gained. Interestingly, already very short walks with only ten steps can clearly identify this case with an extreme level of anisotropy.

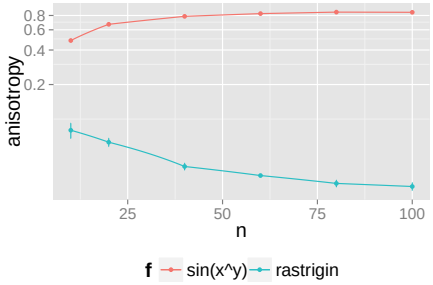


Figure 5.4. Comparison of Real Vector Functions with Different Anisotropy: In particular, this plot shows the auto correlation quartile spread. The relatively homogeneous Rastrigin function is compared to the sine of the exponentiation which has varying degrees of oscillation throughout the landscape.

Figure 5.5 shows the auto correlation isotropy measure for several different variants of the NK and NKi landscapes. While the default NK landscapes exhibit very low anisotropy, especially for low values of k and only a slight increase as k increases, all variants including an exponential weight distribution for the fitness components show a very high value of anisotropy. Interestingly, the isotropy decreases as k increases, reversing the trend as seen in the NK landscapes. The effect of increased anisotropy is only clearly visible for values of $k \leq 5$. From this point on, it seems that the effects of epistasis overlay the individual components enough to make the landscape more and more isotropic again.

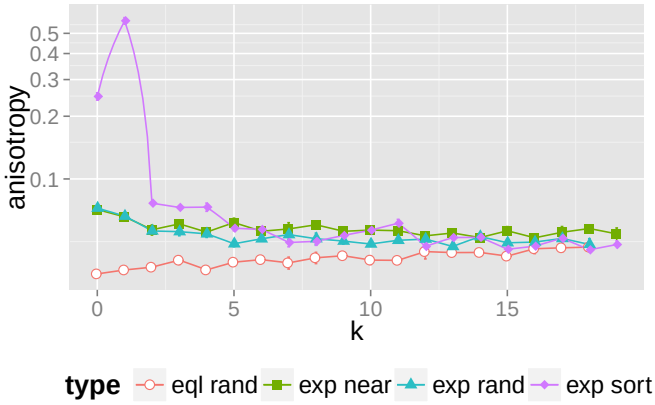


Figure 5.5. While the default NK landscapes, with equally distributed weights and randomly assigned gene interactions, have very low anisotropy slowly increasing with k, all NKi variants with exponential weight distribution of their component fitness functions show significantly more isotropy that decreases as k increases. In all variants with exponential weight distributions show much higher anisotropy. The different gene interaction assignments are either limited range neighbor interaction (`near`), random neighbor interactions (`rand`) as in the original NK landscapes, or lexicographically sorted random interaction (`sort`).

5.3 Comprehensive Analysis Suite

The foundation for deep problem understanding, instance comparison, hardness prediction, algorithm and parameter selection as well as algorithm dominance prediction is the integration of previous approaches, their extensions and the new developments into a comprehensive analysis suite. In this section, all employed analysis methods and the calculated values are detailed. The underlying idea is obvious, the more different perspectives are available, which can potentially describe different characteristics of problem instances, the easier it should become to actually discover relationships between different problem instances as well as between problem instances and algorithm performances.

The whole analysis suite is implemented in HeuristicLab (see Wagner et al., 2008; Wagner, 2009), an extensible, open-source optimization framework as an additional plug-in which is available online[1].

5.3.1 Sample Generation

While most frequently in the literature only random walks are used to elucidate properties of fitness landscapes, through the study of landscape properties it becomes apparent that other sample generation techniques can yield complementary results. Therefore, a total of four different sample generation techniques was used as the basis for further measurements.

5.3.1.1 Random Walks

The first and most obvious choice was a random walk as the fundamental surrogate for analysis. Several different walk lengths were tested, typically with very stable results after about 1,000 iterations. In the literature, walk length of around 1,000,000 steps are used, therefore a compromise

[1] at `http://dev.heuristiclab.com` under additional packages

between speed of analysis and accuracy of 100,000 iterations was used. Moreover, different neighborhood definitions were used to generate the trajectories. Interestingly, while the measurements differ markedly, the resulting predictions and comparisons with other problem instances, showed little difference. This leads to the assumption that the neighborhood structure, at least for typical neighborhoods, which correspond proportionally to a "natural" distance definition on the employed solution space, yield very similar insights.

5.3.1.2 Adaptive Walks and Up-Down Walks

Many of the measurements defined for random walks can directly be applied to samples from different trajectories. One of the more obvious alternative choices are the adaptive walk and the up-down walk. Initially, both adaptive and up-down walks were used, however, an adaptive walk without the ability to escape a local optimum becomes quite uninteresting after it gets stuck. Therefore, only up-down walks were used in later experiments, again with a trajectory length of 100,000 steps in our analysis suite. Moreover, the number of neighborhood samples was limited to ten, which speeds up evaluation and makes the adaptive and up-down walks only slightly adaptive. Therefore, also a slightly larger area of the landscape might be explored, in contrast to selecting the best neighbor from a larger or even from an exhaustive neighborhood. In fact, in (Jones, 1995) it is shown that a first-improvement hill climber and a best-improvement hill climber have very comparable performances. While the first one uses less evaluations in each step the latter has typically fewer iterations, leading to a comparable number of overall evaluations and competitive run-times. Moreover, for these settings, the goal is not to quickly reach an optimum, but to quickly gain insight into the landscape.

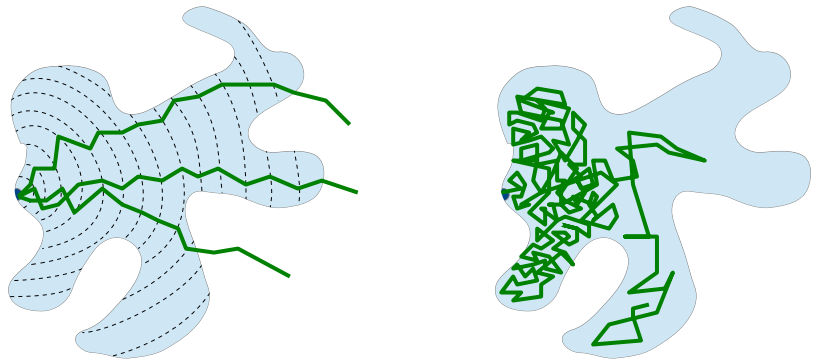
5.3.1.3 Neutral Walks

A very interesting addition to landscape exploration are neutral walks. In this case, it is tried to explore the flat or almost flat areas of a landscape. A neutral walk that examines 100 neighbors in each step was implemented to see if any of the neighbors have equal fitness. If this is the case, the walk is continued on this plateau or otherwise flat area. Moreover, the neutral area is sought to be traversed diametrically. Therefore, whenever a neutral area is discovered, the entry point is remembered and only steps are accepted that increase the distance to the starting point. This is shown in Figures 5.6(a) and (b). If the neutral walk is only restricted in the way neighbors are selected it would quickly become stuck in a neutral area and the measurements would be almost useless or require much more effort for analyzing the extent of the repeatedly traversed neutral area.

Therefore, the approach of allowing only an increase in the distance to the entry point of the neutral area ensures measurements that are more amenable to a systematic analysis. Once the neutral area is exited, the exit point is again used to increase the distance and, therefore, encourage the discovery of new neutral areas that will be traversed in subsequent steps.

To conduct this kind of analysis, a meaningful distance measure has to be provided that allows the measurement between the starting point and possible continuation points in the neutral and non-neutral parts of the landscape. In many cases, even a very simple distance measure can suffice as only the immediate recurrence should be avoided. This simplifies distance calculations in tightly coupled solution spaces such as permutations or binary vectors, where all solution candidates are only a few steps apart from each other.

An alternative to using a distance measure to support the straight traversal of a neutral area would be the collection of all neutral points. While this could lead to a more comprehensive study of the neutrality, the time and space requirements, however, would be much larger. Therefore, for now, the focus has been to find the simplest possible meaningful analysis using a repellent traversal of neutral areas from the entry point.



(a) Diametrical Neutral Walk **(b)** Random Neutral Walk

Figure 5.6. Diametric Neutral Walk: Upon entering a neutral area, the distance to the starting point can only be increased in (a) to avoid creating cycles inside the neutral area that would skew measurements. In (b), any neutral neighbor is accepted. In this case, the walk continues indefinitely inside the neutral area.

In the following section, the measurements that have been collected along all these trajectories are detailed. While for the neutral walk many of the measurements were taken that have been used for random and up-down walks, their analysis in this case is a little more difficult: This time the walk consists of parts inside and parts outside of neutral areas, therefore, the resulting measures also represent this mixture. As expected, the classical measures, therefore, bear relatively little novelty which is the reason new measures were included to specifically analyze the properties of the contained neutral parts of these hybrid walks.

5.3.1.4 Repeated Multi Trajectory Analysis

The combination of repeated restarts at different locations and short random walks allow an insight into the isotropy of fitness landscapes. For this reason, 50 random walks, each with 50 steps starting from 300 random starting points, were conducted. This corresponds more or less to the number of samples taken in the other walks but includes a structure that allows conclusions about the isotropy of the fitness landscape.

5.3.2 Measurements

From the collection of measurements described in Chapter 3, the most promising ones were taken, also with regard to their generality, practicality and applicability. For example, measures that require exhaustive analysis of the solution space or require knowledge of the global optimum have not been used. The aim of this analysis suite is to provide an efficient first characterization of the fitness landscape that enables triaging and quick determination of suitable solution methods. Therefore, the effort for both, the adaptation of the methods to new problem classes as well as the actual measurements, should be minimal.

5.3.2.1 Ruggedness

The measurement of ruggedness is very simple and several concrete values have been deduced from the trajectories. On the one hand, the autocorrelation functions were calculated for all statistically

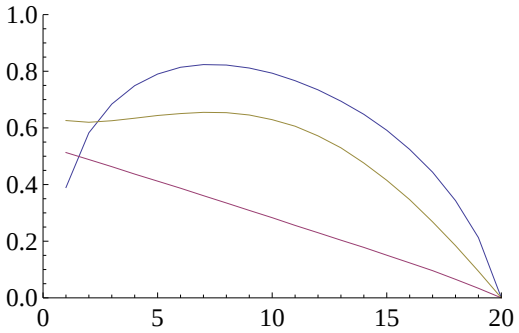
significant steps as described in Section 3.3.3.1, which were used for visual inspection, however, for further analysis and numerical comparison, only a few key figures such as the auto correlation after one step and the last statistically significant correlation length were used. Moreover, the auto correlation coefficient as described in Section 3.3.2.2 and a normalized correlation length and auto correlation coefficient were included. The unnormalized values correlate very well with the problem size, therefore, as the problem size has been included as a separate value, both of these measurements where divided by the problem size to give a "normalized" value that reflects only the remainder, as an attempt to create a more orthogonal description, shown in Equation 5.3, where λ is the autocorrelation coefficient, n is the problem size and l is the correlation length.

$$\lambda' := \frac{\lambda}{n} \qquad l' := \frac{l}{n} \tag{5.3}$$

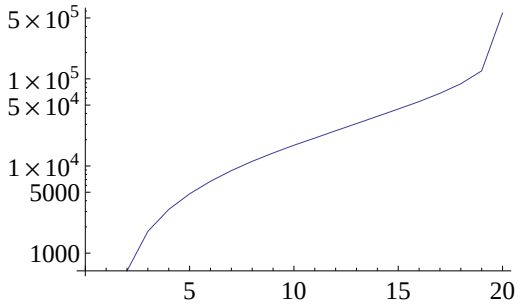
For the multi trajectory analysis, the different ruggedness values have first been aggregated per starting point and then subjected to a distribution analysis. Most importantly the 90% quantile spread was used as a measure of the ruggedness isotropy.

5.3.2.2 Information Analysis

The information analysis as described in Section 3.3.5 was also employed. Moreover, the method described in Section 4.3 was used to select further values for ε for which the analysis has been performed. Again, two stages of analysis have been performed, the first was a comprehensive study of the information curves through all values of ε as shown in Figure 5.7. Most of the charts are quite similar at first glance, however, the difference of the initial values at $\varepsilon = 0$ contain a great deal of information as was discovered in subsequent analyses. Moreover, the apparent maximum values also bear an interesting piece of information. The corresponding values of ε are those fitness differences which reveal the most interesting structure. Smaller values make the landscape seem too rugged to contain much information while larger values make the landscape appear to flat.



(a) Information Analysis: Top to bottom: information content, density basin information, and partial information content.



(b) Corresponding Values of ε

Figure 5.7. Detailed Information Analysis: The x-Axes are the different quantiles of ε that have been used.

Finally, only the initial values were used as in the original paper, in (Vassilev et al., 2000), with $\varepsilon = 0$ and for the isotropy analysis also the maximum values that could be achieved by selecting an appropriate ε value for each of either information content, density basin information, partial information content or information stability.

Moreover, the seemingly trivial value of regularity, which is the number of fitness differences encountered during a walk was used. This simple number can also give important insights into the neighborhood structure. Its diversity can be a hint for optimization algorithms to discern between

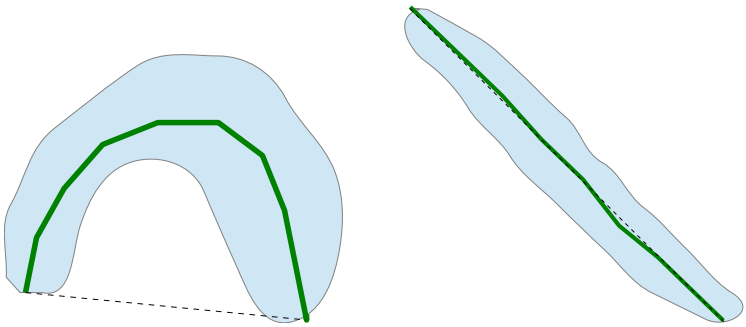
solution candidates that are better. The more levels are comparable, the finer grained the answer to how *much* better a certain candidate is in comparison to an other.

Similar to the ruggedness isotropy, the quantile spread of the various information measures has been used to estimate the information anisotropy of the problem instances.

5.3.2.3 Sub-Walk Characteristics

Finally, the results of ruggedness and information analysis for neutral and up-down walks might be hard to interpret. However, they provide additional possibilities for analysis. Several more characteristics of the up-down and neutral walks were measured as follows:

For both upward, downward and neutral portions of the respective walk types, lengths of these sub-walks have been collected and simple statistics such as averages and variances were calculated. Moreover, for the neutral walks the distances between entry and exit point in the neutral area were collected and their averages and variances have been measured. While the distance may seem very similar to the length which in turn is simply the number of steps, it might be quite different in different circumstances, especially for the neutral walk. Figures 5.6(a) and (b) show two different scenarios with the same neutral walk lengths but different distances.



(a) Curved Neutral Area **(b)** Straight Neutral Area

Figure 5.8. Relevance of Both Distance and Length Analysis: These two examples show two neutral walks that both have the same lengths, however, due to (b) being straight and (a) being curved, analyzing the distances of these neutral walks can reveal additional information.

5.3.3 Isotropy Analysis

Based on the samples obtained with multi trajectory analysis and repeated random walks several isotropy measurements have been carried out. In particular, the following depended isotropy measures of the following base measures have been calculated:

- A combined analysis of both auto correlation and correlation length has been obtained by first averaging over repeated walks from the same staring point and then, performing a distribution analysis over different starting points of the auto correlation at one step, $\rho(1)$, and correlation length.
- The combined analysis of all information values have been conducted similarly. While the basic information analysis values such as the information content or the density basin information at $\varepsilon = 0$ can be calculated with the same scheme as the auto correlation isotropy, a more comprehensive aggregation using different values of ε, as was done for the base measures, was also produced. As the values of ε can be different for each intermediate result the values were sorted first according to ε, based on the new distributions, new averages

using the same number of measurements for each interval have been produced. For example, with 20 quantiles for each of 50 sub analysis results the first 50 ε values from any of the sub-analysis results were used to create the aggregated averages for both ε and the corresponding information values.

- Moreover, up-down isotropy analysis was conducted, where the distributions of up and down walk lengths and their distributions were analyzed.

These trajectories are naturally the most complex to obtain as the intermediate walks are relatively short and have to be repeatedly aggregated before final results can be obtained.

5.3.4 Run-Times

In comparison to the conducted optimization experiments, the fitness landscape analysis has required comparatively little computation time. Especially the single trajectory analysis runs are fast. In Table 5.2, the average run times for problem instances of the QAPLIB are shown. These constitute problems of typical complexity. Of course, problem instances with more complex evaluation functions will require proportionally more run time. However, considering the low computational overhead and the relatively low number of evaluated solution candidates, it seems reasonable to assume that the expected average run times for many other problem classes will be similar and will remain within the scope of a few minutes. The analysis was carried out on several blade systems, each with two Intel Xeon E5420 quad core CPUs with 2.5 GHz. Two different isotropy analysis algorithms runs where conducted. One, where the inner trajectory was random for the measurement of ruggedness and information while, in a second run, repeated up-down walks where performed to explore the isotropy of adaptive features of the landscapes.

Table 5.2. Average Run Times of the Fitness Landscape Analysis Algorithms: This table summarizes the number of steps, the number of considered neighbors in each step and the total number of evaluations. Moreover, the average run time per instances as well as the amortized run time per evaluation are shown.

Sampling	steps	neigh	evals	⌀ RT	rel. RT
Random	100,000	1	100,000	59 s	0.59 μs
Up-Down	100,000	10	1,000,000	190 s	0.19 μs
Neutral	10,000	100	100,000	140 s	1.40 μs
MTA-1	$300 \times 50 \times 50$	1	750,000	480 s	0.64 μs
MTA-2	$100 \times 50 \times 50$	10	2,500,000	624 s	0.25 μs

5.3.5 Summary

Many different values have been collected as shown in Tables 5.3 and 5.4. However, many of these values are conceptually quite similar to each other. In fact, all of them have been extracted from a handful of different trajectories. In the following sections, the analysis results using these values are used to describe problem instances and predict different properties pertaining to the interplay between problem instances and algorithm applications. All of these values have been collected for different fitness landscape variants. As a different neighborhood structure, or mutation operator, conceptually defines its own landscape variant these values have been derived repeatedly to be able to compare different landscape variants of each problem instance.

Table 5.3. Summary of Sampling Generators in the Analysis Suite

Algorithm	Parameter	Value
Random Walk	Nr of Iterations	100,000
	Nr of Neighbors	1
Up-Down Walk	Nr of Iterations	100,000
	Nr of Neighbors	10
Neutral walk	Nr of Iterations	10,000
	Nr of Neighbors	100
Multi Trajectory 1	Sample Point Selection	Random
	Nr of Sample Points	300
	Nr of Iterations	50
	Nr of Repetitions	50
	Inner Trajectory	Random Walk
Multi Trajectory 2	Sample Point Selection	Random
	Nr of Sample Points	100
	Nr of Iterations	50
	Nr of Repetitions	50
	Inner Trajectory	Up-Down Walk
	Inner Nr of Neighbors	10

Table 5.4. Summary of Measures in the Analysis Suite

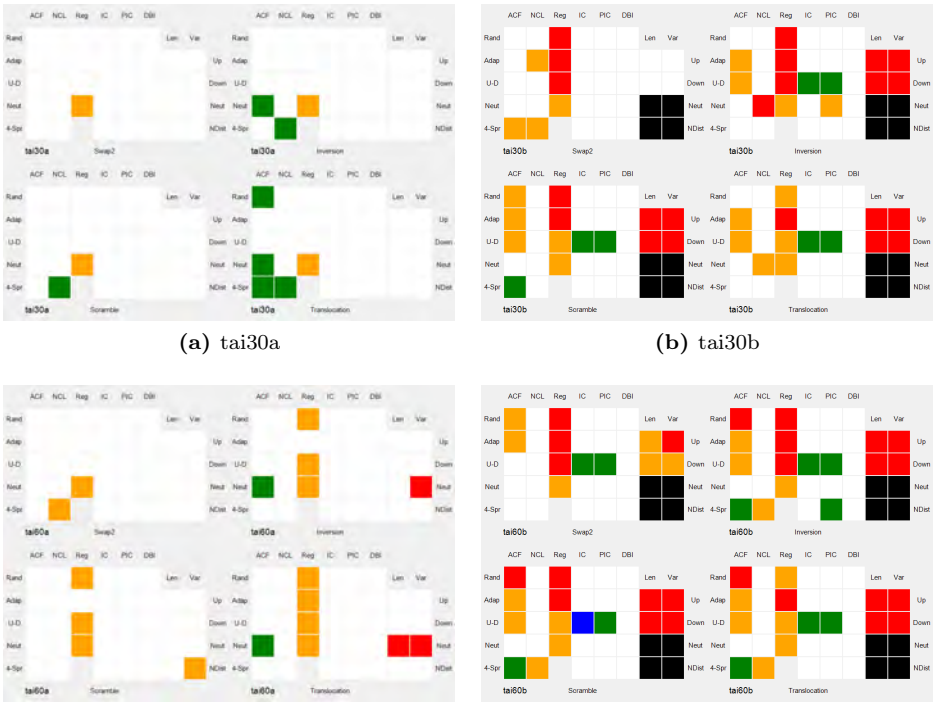
Used in				
Rand	U/D	Neut	MTA	Value
☑	☑	☑	☑	Auto Correlation 1
☑	☑	☑	☐	Auto Correlation Coefficient
☑	☑	☑	☐	Normalized Auto Correlation Coefficient
☑	☑	☑	☑	Correlation Length
☑	☑	☑	☐	Normalized Correlation Length
☑	☑	☑	☑	Information Content
☑	☑	☑	☑	Density Basin Information
☑	☑	☑	☑	Partial Information Content
☑	☑	☑	☑	Information Stability Content
☑	☑	☑	☐	Regularity
☑	☑	☑	☐	Relative Regularity
☐	☑	☑	☐	Average Sub Walk Length
☐	☑	☑	☐	Sub Walk Length Variance
☐	☐	☑	☐	Average Walk Distance
☐	☐	☑	☐	Walk Distance Variance
☐	☐	☐	☑	Auto Correlation Distribution
☐	☐	☐	☑	Correlation Length Distribution
☐	☐	☐	☑	Information Content Distribution
☐	☐	☐	☑	Density Basin Information Distribution
☐	☐	☐	☑	Partial Information Content Distribution
☐	☐	☐	☑	Information Stability Distribution
☐	☐	☐	☑	Peak Information Content Distribution
☐	☐	☐	☑	Peak Density Basin Information Distribution
☐	☐	☐	☑	Peak Partial Information Content Distribution
☐	☐	☐	☑	Peak Information Stability Distribution
☐	☐	☐	☑	Peak Information Quality Delta Distribution
☐	☐	☐	☑	Average Up/Down Walk Length Distribution
☐	☐	☐	☑	Up/Down Walk Length Variance Distribution

5.4 Quadratic Assignment Problem

The quadratic assignment problem has served as a grateful example for many of the analysis methods. In the following sections some examples of the generated results will be briefly commented on.

5.4.1 Instance Characterization

In Figure 5.9 several instances of box charts are shown from the *tai* family of QAP problems (see Taillard, 1991, 1995). These problem instances are randomly generated and uniformly distributed. The difference between the subclasses is that `b` instances are asymmetric. As can be seen in Figure 5.9, the individual classes correspond quite well to each other. This effect was already demonstrated in Section 4.5.3, in particular in Figure 4.19, where the resulting distances are used in a self organizing map.



(a) tai30a **(b)** tai30b

(c) tai60a **(d)** tai60b

Figure 5.9. Characteristic Properties of QAP Generators: The distinct properties of each generator class are clearly visible and can explain why the obtained distances work so well.

5.4.2 Hardness Prediction

As explained in Sections 4.4.2.1 and 4.7 the fitness landscape analysis values and the instance comparison show great potential for problem hardness prediction. Tables 5.5 through 5.7 show the details of the best models for predicting various performance indicators of robust taboo search, genetic algorithms and simulated annealing using the same data as described in Pitzer et al. (2013a).

Table 5.5. Best Regression Model for Average Hardness Prediction using FLA and Problem-Specific Inputs. This model has an r^2 of 0.93. The input values have been normalized so their extent directly reflects their influence on the problem hardness.

(a) Regression Model

Swap2 Random Information Content Value	.11
Swap2 Random Regularity Value	.09
Swap2 Up Down Down Walk Len Var	-.36
Problem Size	.75
Swap2 Random Auto Correlation1 Coeff Norm	-.06
Swap2 Up Down Auto Correlation1 Coeff	.56
PS Sparcity Distance	-.15
PS Distribution Flow Mean	-.02
PS Distribution Flow Variance	-.00
PS Distribution Flow Skewness	-.09
PS Distribution Flow Kurtosis	-.14
PS Distribution Flow Normality	.02
PS Distribution Distance Kurtosis	.09
PS Distribution Distance Normality	-.04
PS Asymmetry Flow Count	-.37
PS Intransitivity Flow	-.18
PS Intransitivity Distance	-.00
PS Throughput Flow Out Flow Variance	-.27
PS Throughput Flow Out Flow Skewness	.08
PS Throughput Flow Out Flow Kurtosis	.01
PS Throughput Flow Flow Sum Normality	-.02
PS Throughput Flow Flow Surplus Mean	.06
PS Throughput Flow Flow Surplus Variance	.06
PS Throughput Distance In Flow Mean	-.07
PS Throughput Distance In Flow Skewness	-.00
PS Throughput Distance In Flow Kurtosis	.08
PS Throughput Distance Out Flow Mean	-.00
PS Throughput Distance Out Flow Skewness	-.04
PS Cost Avg	-.12
PS Cost Variance	.15

(b) Scatter Plot

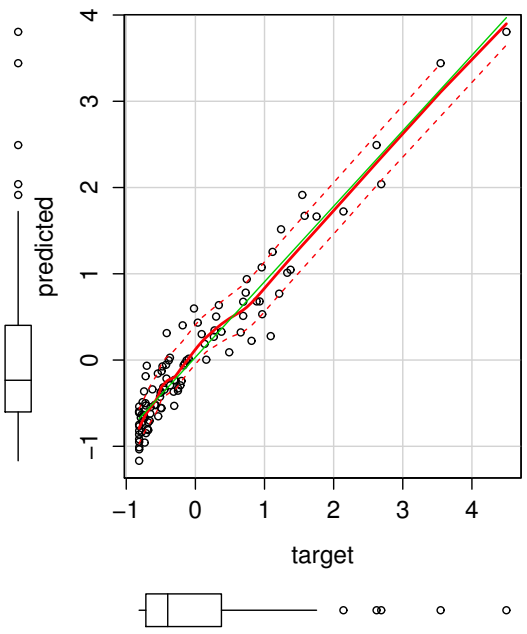


Table 5.6. Best Regression Model for Average Scaled Difference of the Genetic Algorithm. This model has an r^2 of 0.98.

(a) Regression Model

Inversion Neutral Auto Correlation1	-.02
Inversion Neutral Correlation Length	-.01
Inversion Neutral Density Basin Information Value	.15
Inversion Neutral Neutral Walk Length Variance	.00
Inversion Neutral Regularity Value	.05
Inversion Up Down Down Walk Len Var	-.16
Translocation Up Down Solution Creator Length	.49
Inversion Up Down Auto Correlation1 Coeff Norm	-.06
PS Sparcity Distance	-.05
PS Distribution Flow Mean	-.00
PS Distribution Flow Skewness	-.19
PS Distribution Distance Skewness	-.02
PS Distribution Distance Kurtosis	-.00
PS Distribution Distance Normality	-.01
PS Asymmetry Flow	.14
PS Asymmetry Flow Count	.38
PS Intransitivity Flow Count	-.06
PS Intransitivity Distance	.01
PS Intransitivity Distance Count	-.05
PS Throughput Flow In Flow Variance	-.22
PS Throughput Flow In Flow Skewness	-.02
PS Throughput Flow In Flow Normality	.03
PS Throughput Flow Flow Surplus Skewness	.19
PS Throughput Flow Flow Surplus Kurtosis	.00
PS Throughput Distance In Flow Mean	.00
PS Throughput Distance In Flow Variance	-.05
PS Throughput Distance Flow Surplus Skewness	-.13
PS Throughput Distance Flow Surplus Kurtosis	-.04
PS Throughput Distance Flow Surplus Normality	.05
PS Cost Avg	.05
PS Cost Variance	.02

(b) Scatter Plot

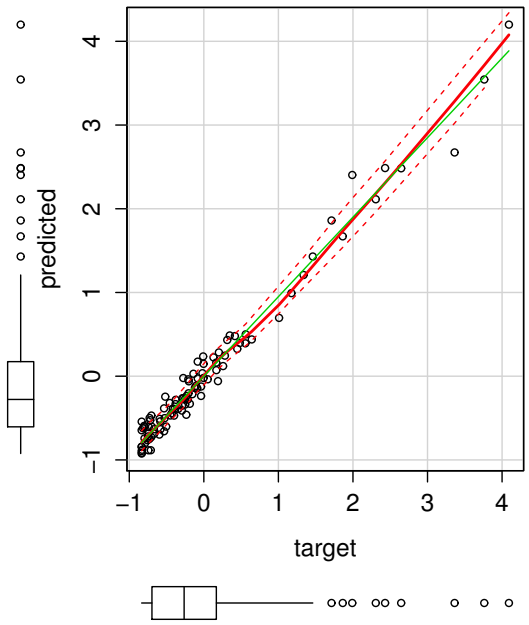
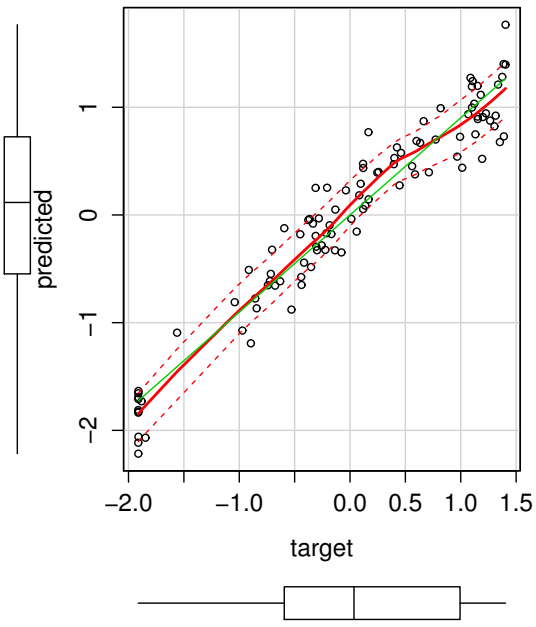


Table 5.7. Best Regression Model for Average Iterations of Simulated Annealing. This model has an r^2 of 0.93.

(a) Regression Model

Scramble Up Down Correlation Length	.11
Scramble Up Down Density Basin Information Value	.21
Scramble Up Down Regularity Value	.39
Scramble Up Down Upper Variance	.05
Translocation Up Down Solution Creator Length	.09
Scramble Up Down Auto Correlation1 Coeff	.21
Scramble Up Down Auto Correlation1 Coeff Norm	-.44
PS Dominance Flow	-.02
PS Distribution Flow Mean	-.11
PS Distribution Flow Variance	.09
PS Distribution Flow Kurtosis	-.19
PS Distribution Flow Normality	.01
PS Distribution Distance Variance	.00
PS Distribution Distance Skewness	.13
PS Distribution Distance Normality	.02
PS Asymmetry Flow Count	-.21
PS Asymmetry Distance	.06
PS Intransitivity Flow	-.02
PS Intransitivity Flow Count	.27
PS Intransitivity Distance	-.20
PS Intransitivity Distance Count	-.68
PS Throughput Flow In Flow Kurtosis	.02
PS Throughput Flow Out Flow Variance	-.11
PS Throughput Flow Out Flow Normality	-.03
PS Throughput Flow Flow Surplus Mean	-.02
PS Throughput Flow Flow Surplus Skewness	-.28
PS Throughput Flow Flow Surplus Kurtosis	-.09
PS Throughput Distance In Flow Mean	-.05
PS Throughput Distance In Flow Kurtosis	.05
PS Throughput Distance Out Flow Kurtosis	.03
PS Throughput Distance Out Flow Normality	-.04
PS Throughput Distance Flow Sum Skewness	-.13
PS Cost Avg	-.08
PS Cost Variance	.03

(b) Scatter Plot



5.4.3 Algorithm Domination

The detailed results for algorithm domination estimation using different significance thresholds are shown in Table 5.8. Here it can be seen that in most cases, there is a clear winner right from the start. In some instances, the significance level has to be increased to arrive at such a clear cut decision. Finally, however, only very few instances remain for which there is a true alteration between algorithms during their optimization processes, where it really depends on the available computing resources.

In Table 5.9, the results of all classifiers that were used are shown for reference. All forty classifiers in Weka 3.6 (see Witten et al., 2011) that are available for classification were used. The table is sorted by performance from top to bottom in descending order. The best algorithm was support vector classification (see Vapnik, 1995) using the sequential minimal optimization algorithm for training (SMO) (see Platt, 1998). The next few algorithms, namely multilayer perceptron using backpropagation (see Rosenblatt, 1961), LogitBoost (see Landwehr et al., 2005) with simple logistic regression as base learners (see Hosmer and Lemeshow, 2000), and function trees (see Gama, 2004), also with logistic regression functions, have very similar similar performances. Even plain logistic regression performs quite well. All results were obtained using threefold cross validation All results were obtained using threefold cross validation. Moreover, the results for the individual folds deviate only by up to 2% from the averages shown in Table 5.9.

With this experiment it was not actually tried to find the best classifier for algorithm dominance prediction but rather to show the suitability of the data itself for prediction. With an average performance above 70% percent using default settings for many classification schemes this appears to be a reasonable claim.

Table 5.8. Domination Series at Different Significance Levels: While for many problems, the choice is obvious, for other the significance level has to be raised. In the end, only few instances remain undecided. Please note, VNS actually always starts out dominant as the first fitness measurement is made only after several evaluations. Therefore, in this table, the leading pseudo-dominance of VNS has been removed.

	Significance Level		
Problem Name	0.9	0.99	0.9999
bur26a	R	R	R
bur26b	V	V	V
bur26c	V	V	V
bur26d	V	V	V
bur26e	RV	V	V
bur26f	V	V	V
bur26g	V	V	V
bur26h	V	V	V
chr12a	R	R	R
chr12b	RVR	R	R
chr12c	VR	R	R
chr15a	VR	R	R
chr15b	VR	R	R
chr15c	VR	R	R
chr18a	VR	R	R
chr18b	VR	R	R
chr20a	VR	R	R
chr20b	VR	R	R
chr20c	V	V	V
chr22a	R	R	R
chr22b	RVR	R	V
chr25a	R	V	V
els19	V	V	V
had12	RV	RV	V
had14	V	V	V
had16	V	V	V
had18	RV	V	V
had20	RV	V	V
kra30a	R	R	V
kra30b	RVR	R	R
kra32	RVR	R	R
lipa20a	R	R	R
lipa20b	R	R	R
lipa30a	R	R	R
lipa30b	R	R	R
lipa40a	R	R	R
lipa40b	R	R	R
lipa50a	R	R	R
lipa50b	R	R	R
lipa60a	R	R	R
lipa60b	R	R	R
lipa70a	RVRV	RV	V
lipa70b	RVR	R	R
lipa80a	RV	RV	RV
lipa80b	RV	RV	V
lipa90a	RV	RV	RV
lipa90b	RV	RV	R
nug12	R	R	R
nug14	R	R	R
nug15	R	R	R
nug16a	R	R	R
nug16b	R	R	R
nug17	R	R	R
nug18	R	R	R
nug20	R	R	R
nug21	R	R	R
nug22	RVR	R	R
nug24	R	R	R

	Significance Level		
Problem Name	0.9	0.99	0.9999
nug25	R	R	R
nug27	R	R	R
nug28	R	R	R
nug30	R	R	R
rou12	R	R	R
rou15	R	R	R
rou20	R	R	R
scr12	R	R	R
scr15	RVR	R	R
scr20	R	R	R
sko100a	RV	V	V
sko100b	V	V	V
sko100c	VRV	V	V
sko100d	V	V	V
sko100e	V	V	V
sko100f	RV	V	V
sko42	R	R	R
sko49	R	R	R
sko56	R	R	R
sko64	R	V	V
sko72	RV	V	V
sko81	V	V	V
sko90	V	V	V
ste36a	R	R	V
ste36b	RVR	R	R
ste36c	RVR	R	R
tai100a	RV	RV	RV
tai100b	V	V	V
tai10a	R	R	R
tai10b	V	V	V
tai12a	R	R	R
tai12b	V	V	V
tai150b	RV	V	V
tai15a	R	R	R
tai15b	R	R	R
tai17a	R	R	R
tai20a	R	R	R
tai20b	V	V	V
tai256c	V	V	V
tai25a	R	R	R
tai25b	V	V	V
tai30a	R	R	R
tai30b	V	V	V
tai35a	R	R	R
tai35b	RVR	R	R
tai40a	R	R	R
tai40b	V	V	V
tai50a	R	R	R
tai50b	R	V	V
tai60a	R	R	R
tai60b	V	V	V
tai64c	V	V	V
tai80a	RV	RV	RV
tai80b	V	V	V
tho150	V	V	V
tho30	R	R	R
tho40	R	R	R
wil100	V	V	V
wil50	R	R	R

Table 5.9. Accuracy Values of Algorithm Domination Classifiers using threefold cross validation.

Data Set	all			binary			resampled			avg
Max Variable Correlation	1	.99	.9	1	.99	.9	1	.99	.9	
SMO	80.57	80.40	78.60	84.33	84.21	82.74	81.96	82.44	81.36	81.85
MultilayerPerceptron	77.55	77.58	77.81	81.66	81.45	81.54	79.44	80.07	79.53	79.63
SimpleLogistic	76.61	77.18	76.27	80.36	80.77	80.71	77.85	78.41	79.17	78.59
FT	76.44	77.66	77.04	80.42	80.48	80.65	78.04	77.81	78.06	78.51
LMT	76.30	76.84	76.21	79.98	80.80	80.60	77.78	78.48	78.72	78.41
IBk	76.41	76.84	75.98	80.40	79.84	79.33	77.63	78.04	78.78	78.14
RandomForest	76.30	75.90	75.36	78.42	78.65	78.60	77.30	78.48	77.44	77.38
Logistic	74.99	74.27	72.54	80.62	79.79	80.83	77.96	76.74	78.28	77.34
DTNB	74.62	74.22	74.19	79.13	79.46	79.28	76.04	76.22	76.44	76.62
IB1	73.19	73.62	75.16	76.68	77.56	79.06	76.93	77.96	78.58	76.53
DMNBtext	75.87	74.59	75.84	78.03	76.86	78.03	74.89	73.85	75.11	75.90
RBFNetwork	74.59	74.53	74.22	78.23	77.58	77.72	74.63	74.74	75.94	75.80
DecisionTable	74.05	74.22	74.93	77.13	78.51	78.72	73.44	72.70	73.58	75.25
J48	72.93	73.08	73.30	75.96	76.20	76.40	75.22	75.52	75.56	74.91
NBTree	72.59	72.65	73.65	76.27	76.53	75.67	74.19	75.07	75.50	74.68
SimpleCart	73.16	73.62	74.70	75.88	76.85	77.74	72.44	72.93	74.50	74.65
JRip	74.13	73.53	74.47	76.32	77.53	77.83	72.70	72.00	71.94	74.50
J48graft	72.85	72.93	73.13	75.88	75.94	75.85	74.30	74.44	75.00	74.48
NaiveBayes	73.90	73.05	72.25	77.65	78.29	76.92	72.93	73.33	71.83	74.46
BayesNet	66.44	66.24	68.18	80.64	80.37	79.58	75.70	75.78	76.94	74.43
RandomTree	72.54	72.59	72.19	76.35	75.83	76.49	73.74	73.93	75.14	74.31
PART	72.28	72.31	72.54	75.24	75.33	75.57	74.85	75.07	73.94	74.13
BFTree	72.68	71.94	72.74	75.23	76.08	76.74	73.00	73.11	73.61	73.90
LADTree	70.94	71.79	72.42	74.63	75.77	75.62	73.11	74.56	74.31	73.68
VFI	71.94	72.28	72.19	75.82	76.11	77.17	70.22	71.11	73.31	73.35
HyperPipes	71.40	72.19	73.30	74.44	75.23	76.29	70.63	71.81	72.69	73.11
REPTree	73.13	73.22	73.79	74.76	74.48	75.50	70.56	70.89	71.36	73.08
Ridor	72.05	71.94	71.82	75.62	73.93	74.85	70.85	71.30	71.61	72.66
KStar	71.88	72.19	70.71	73.08	73.41	72.85	71.89	72.33	73.14	72.39
NNge	71.97	71.77	72.39	73.00	73.17	74.69	70.81	71.67	71.44	72.32
LWL	71.08	70.94	70.94	75.21	75.68	75.50	70.52	70.44	70.36	72.30
DecisionStump	70.28	69.17	69.26	71.84	70.54	70.98	66.52	65.44	65.50	68.84
ConjunctiveRule	68.80	67.81	68.38	69.31	67.87	68.05	66.15	64.89	64.08	67.26
ZeroR	58.12	58.12	58.12	60.17	60.17	60.17	50.00	50.00	50.00	56.10

5.5 Vehicle Routing Problem

In this section some aspects of the vehicle routing problem are summarized that were discovered using the proposed analysis methods. While such an extensive algorithm performance test grid as for the quadratic assignment problem was not built, it was analyzed how fitness landscape analysis results relate to problem generators and how they relate to problem specific measurements.

5.5.1 Problem Specific Analysis

In Table 5.10, the problem specific characteristics that were used are shown. These characteristics are relatively simple to extract and are detailed in Section 4.3.3. As can be seen from this table, the different eccentricity values are quite similar to each other, so in practice, one of them is sufficient.

5.5.2 Instance Characterization

In the following figures, the relation between problem-specific analysis and fitness landscape measures is examined exemplarily for extreme examples. While in Section 4.7.2 the general trend for correspondence between problem-specific values and fitness landscape measures is investigated, in this section, several concrete examples are examined.

For this purpose, a new method for the visualization for VRP problem instances was developed that visually summarizes several important aspects. On the one hand, distribution histograms are shown for distances and demands while on the other hand, the actual geographical distribution of customers is plotted. Moreover, the demands of the individual customers are shown on the maps by varying the size of the respective cities.

Figure 5.10 shows examples with very high or very low clustering. As experience shows, high clustering instances are in general easier to solve. On the fitness landscape analysis side, we can see that this highly clustered instance has average FLA measures. On the other hand, the very evenly distributed instances with low clustering show increased values for neutral and up-down sub-walk lengths. This could indicate a generally flatter and better connected landscape, where traversal to local optima takes longer.

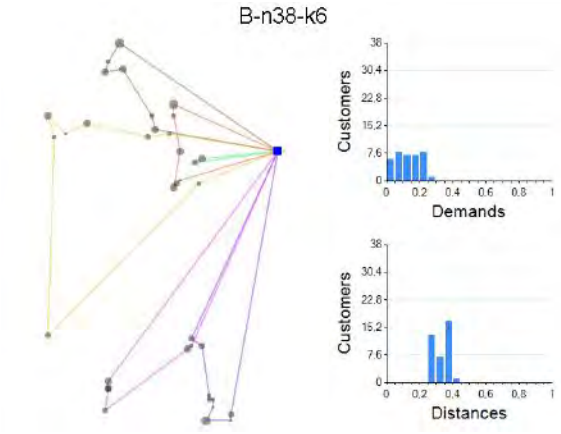
In Figure 5.11, instances with extremely high or low demands are compared. In these cases, also the FLA charts look very different again. While the instance with high demands has lower ruggedness values for random walks and longer neutral walks, the instance with low demands shows an increase in regularity.

Interestingly, as shown in Figure 5.12, the distribution of distances does not seem to be reflected very much in the FLA values. Finally, comparing the charts in Figure 5.13, there is, again, not such a clear distinction between the two scenarios with either high or low eccentricity.

These examples show that direct interpretation of fitness landscape analysis measures is rather difficult. While in some cases, they can give definitive clues about certain circumstances pertaining to particular problem instances, in other cases, while a difference can be observed, it seems almost impossible to create a direct interpretation just by looking at the raw values. In the future, therefore, efforts should go towards normalization of FLA values and their combination into useful concepts that are better interpretable.

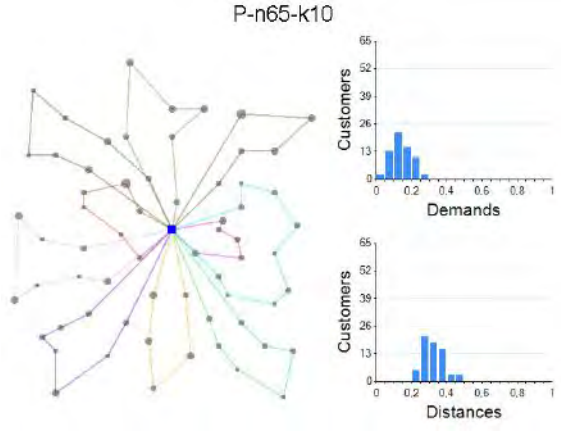
Table 5.10. Problem-Specific Values for Vehicle Routing Problems: This table shows the number of customers, the clustering index, mean and variance of both distances and demands as well as all depot eccentricity measures which are geographic, distance-based and demand-weighted distance-based.

			Distance		Demand		Eccentricity		
Instance	cust	CI	μ	σ	μ	σ	geo	dist	d&d
A-n32-k5	31	0.05	0.43	0.06	0.13	0.07	0.31	0.35	0.32
A-n33-k5	32	0.04	0.38	0.06	0.14	0.07	0.13	0.15	0.17
A-n33-k6	32	0.05	0.36	0.07	0.16	0.11	0.15	0.16	0.16
A-n34-k5	33	0.04	0.38	0.07	0.14	0.07	0.20	0.22	0.23
A-n36-k5	35	0.05	0.39	0.07	0.12	0.07	0.32	0.34	0.33
A-n37-k5	36	0.04	0.37	0.06	0.11	0.08	0.12	0.12	0.13
A-n37-k6	36	0.04	0.37	0.06	0.15	0.11	0.31	0.31	0.29
A-n38-k5	37	0.03	0.37	0.07	0.13	0.07	0.17	0.16	0.14
A-n39-k5	38	0.03	0.37	0.07	0.12	0.08	0.31	0.27	0.31
A-n39-k6	38	0.03	0.40	0.06	0.13	0.12	0.23	0.21	0.22
A-n44-k6	43	0.03	0.36	0.07	0.13	0.07	0.29	0.27	0.27
A-n45-k6	44	0.03	0.42	0.07	0.13	0.07	0.20	0.21	0.20
A-n45-k7	44	0.03	0.37	0.05	0.14	0.07	0.36	0.39	0.42
A-n60-k9	59	0.02	0.40	0.06	0.14	0.09	0.35	0.32	0.36
A-n61-k9	60	0.02	0.34	0.05	0.15	0.10	0.14	0.13	0.14
A-n62-k8	61	0.02	0.39	0.07	0.12	0.08	0.33	0.35	0.36
A-n63-k10	62	0.03	0.36	0.06	0.15	0.10	0.25	0.23	0.24
A-n63-k9	62	0.02	0.39	0.06	0.14	0.08	0.43	0.45	0.43
A-n64-k9	63	0.02	0.35	0.07	0.13	0.09	0.38	0.39	0.37
A-n65-k9	64	0.02	0.40	0.05	0.13	0.07	0.18	0.16	0.17
B-n31-k5	30	0.50	0.24	0.12	0.13	0.06	0.44	0.38	0.37
B-n34-k5	33	0.20	0.33	0.09	0.13	0.15	0.19	0.39	0.44
B-n35-k5	34	0.33	0.43	0.04	0.12	0.08	0.36	0.42	0.41
B-n38-k6	37	0.50	0.33	0.05	0.13	0.07	0.30	0.28	0.26
B-n39-k5	38	0.20	0.39	0.06	0.11	0.07	0.21	0.20	0.17
B-n41-k6	40	0.25	0.38	0.07	0.14	0.06	0.23	0.25	0.28
B-n43-k6	42	0.13	0.35	0.06	0.12	0.07	0.27	0.30	0.33
B-n44-k7	43	0.20	0.37	0.04	0.15	0.11	0.35	0.32	0.37
B-n45-k5	44	0.11	0.39	0.07	0.11	0.07	0.19	0.12	0.12
B-n45-k6	44	0.20	0.24	0.08	0.13	0.07	0.11	0.20	0.22
B-n56-k7	55	0.33	0.36	0.08	0.11	0.07	0.26	0.17	0.16
B-n57-k7	56	0.20	0.40	0.09	0.12	0.10	0.35	0.32	0.34
B-n57-k9	56	0.14	0.31	0.06	0.14	0.07	0.46	0.49	0.52
B-n63-k10	62	0.25	0.39	0.08	0.15	0.08	0.29	0.39	0.35
B-n64-k9	63	0.20	0.35	0.09	0.14	0.09	0.11	0.07	0.05
E-n22-k4	21	0.06	0.31	0.05	0.17	0.11	0.07	0.08	0.06
E-n23-k3	22	0.06	0.34	0.07	0.10	0.18	0.04	0.08	0.13
E-n30-k3	29	0.05	0.33	0.04	0.09	0.13	0.10	0.06	0.13
E-n33-k4	32	0.06	0.24	0.07	0.11	0.10	0.42	0.46	0.45
F-n45-k4	44	0.05	0.25	0.08	0.08	0.13	0.13	0.09	0.06
P-n16-k8	15	0.07	0.37	0.06	0.44	0.25	0.28	0.34	0.37
P-n19-k2	18	0.06	0.40	0.06	0.10	0.05	0.29	0.36	0.38
P-n20-k2	19	0.05	0.39	0.06	0.10	0.05	0.29	0.37	0.38
P-n21-k2	20	0.05	0.38	0.06	0.09	0.05	0.29	0.36	0.39
P-n22-k2	21	0.05	0.38	0.06	0.09	0.05	0.29	0.36	0.39
P-n22-k8	21	0.06	0.31	0.05	0.34	0.21	0.07	0.08	0.06
P-n23-k8	22	0.05	0.38	0.06	0.34	0.19	0.29	0.35	0.38
P-n40-k5	39	0.03	0.38	0.06	0.11	0.06	0.05	0.09	0.09
P-n45-k5	44	0.02	0.38	0.06	0.10	0.06	0.05	0.05	0.05
P-n60-k10	59	0.02	0.31	0.05	0.16	0.06	0.03	0.03	0.01
P-n60-k15	59	0.02	0.34	0.06	0.24	0.09	0.04	0.02	0.01
P-n65-k10	64	0.02	0.32	0.06	0.14	0.06	0.02	0.03	0.01

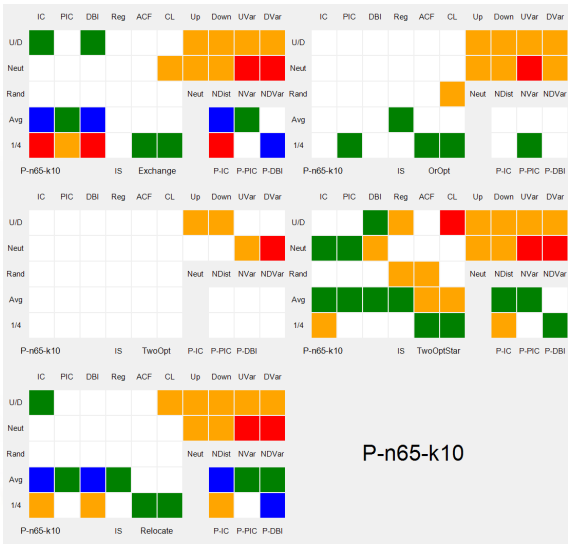


(a) High Clustering Problem Specific Analysis

(b) High Clustering Fitness Landscape Analysis

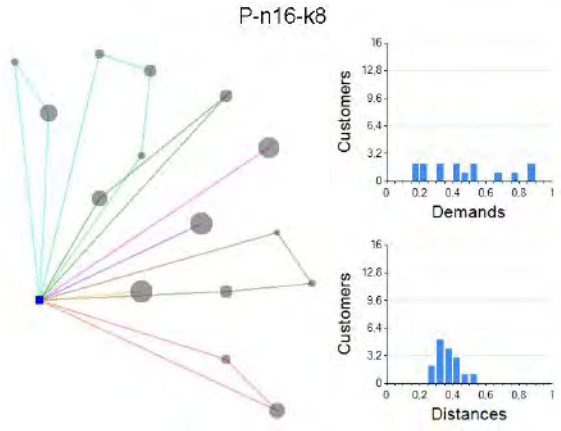


(c) Low Clustering Problem Specific Analysis

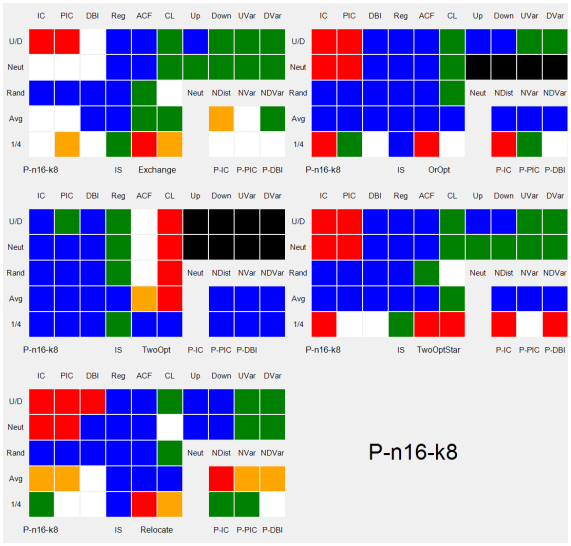


(d) Low Clustering Fitness Landscape Analysis

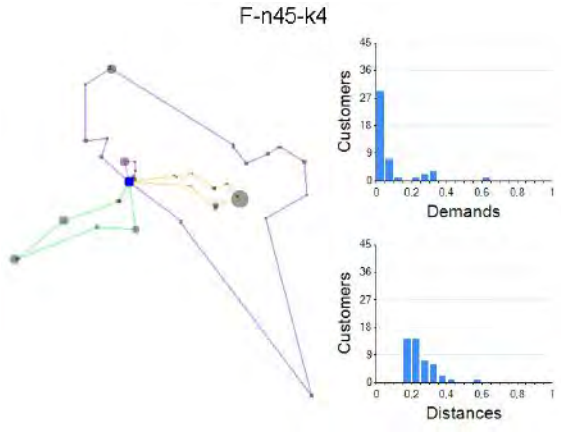
Figure 5.10. VRP Instances with Extreme Clustering Indexes: In these cases, problem specific properties can clearly be discerned also in the fitness landscape results.



(a) High Demands Problem Specific Analysis



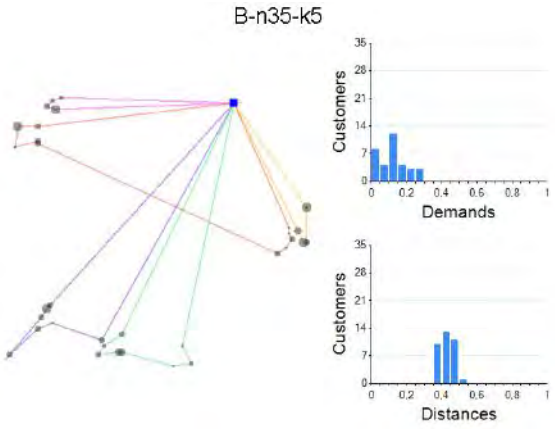
(b) High Demands Fitness Landscape Analysis



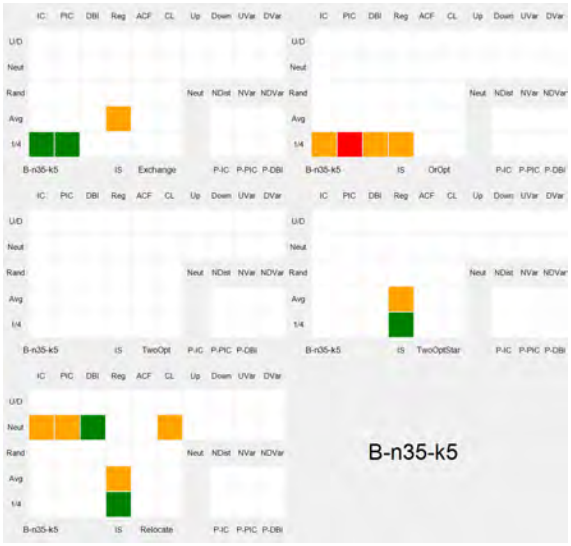
(c) Low Demands Problem Specific Analysis

(d) Low Demands Fitness Landscape Analysis

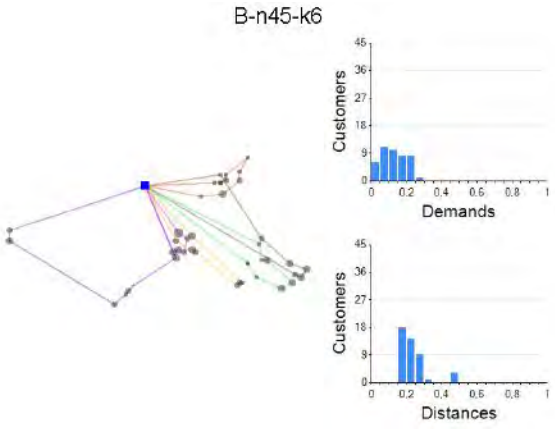
Figure 5.11. VRP Instances with Extreme Demands: Also in this case, the fitness landscape analysis results show a strong difference between the instances with different problem specific properties.



(a) High Distances Problem Specific Analysis



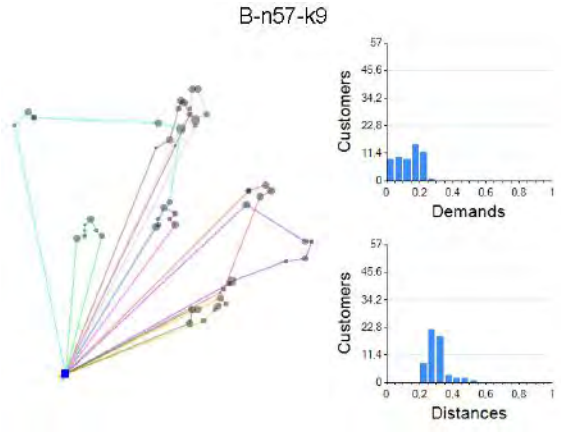
(b) High Distances Fitness Landscape Analysis



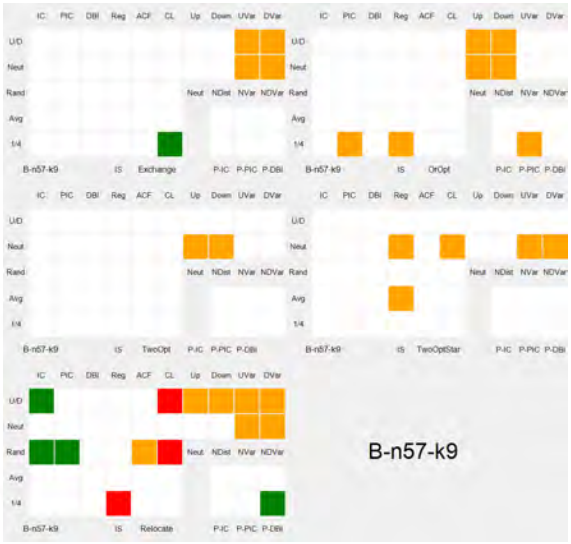
(c) Low Distances Problem Specific Analysis

(d) Low Distances Fitness Landscape Analysis

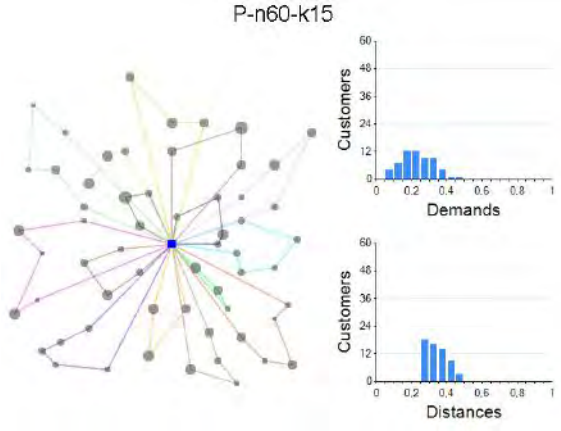
Figure 5.12. VRP Instances with Extreme Distances: In this case the conclusion possible from fitness landscape analysis is ambiguous at best and wrong at worst. In other words, no clear conclusions are possible in this case.



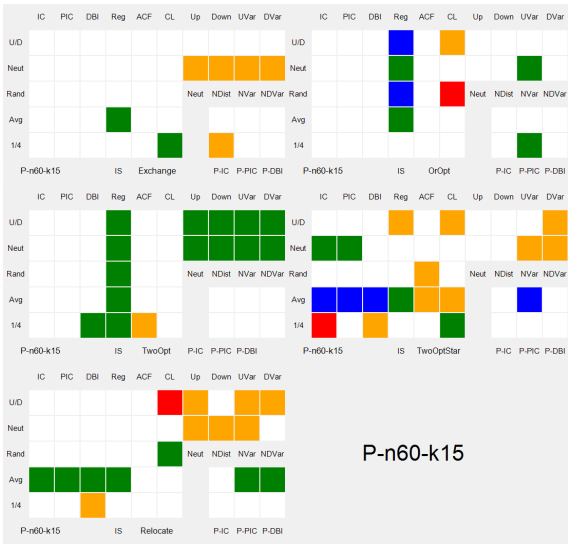
(a) High Eccentricity Problem Specific Analysis



(b) High Eccentricity Fitness Landscape Analysis



(c) Low Eccentricity Problem Specific Analysis



(d) Low Eccentricity Fitness Landscape Analysis

Figure 5.13. VRP Instances with Extreme Eccentricity: Also in this case, while we could assume a certain influence of sub-walk characteristics, any conclusions are not well founded.

Chapter 6

Conclusion

In this work we have stepped on relatively new ground, combining powerful existing methods with new ones in large scale studies for the suitability and practical applicability of fitness landscape analysis. However, while concrete goals are withing reach, there is still a lot of work to do. Therefore, this conclusion consists of a short summary of the presented work and a longer summary of open issues and further ideas for research.

6.1 Summary

Fitness landscape analysis was devised as a general method for problem understanding. As such many different aspects of fitness landscapes have been investigated. A comprehensive survey of existing methods showed the diversity but also the disparity of different methods each elucidating different aspects of optimization problems.

Based on a thorough review of existing techniques several new methods for fitness landscape analysis have been introduced. Besides a few minor enhancements and tweaks of existing methods, an efficient methods for the in-depth study of basins of attraction and fitness barriers as well a method for their visualization has been introduced. Moreover, a new approach to the investigation of isotropy was presented. Finally, a popular testbed landscape has been extended to create isotropic effects as a proof of concept for the newly introduced isotropy analysis.

Using the basin analysis method, it is possible to obtain a clearer understanding of the influence of certain shapes in the fitness landscape. In particular, the shape of basins and the intermediate fitness barriers have been studied, and it was shown that they their characteristics can become more important than just the number of local optima for the determination of problem difficulty.

Using a new, more pragmatic approach to isotropy analysis, it was possible to measure obvious cases of isotropy but also to observe some surprising effects. While on real vector landscapes, the individual walk lengths could be easily increased to also increase the level of confidence of the isotropy measures. In combinatorial landscapes that are typically more tightly interconnected, however, the choice of walk lengths must be made very carefully to find the right balance between both locality and statistical significance.

Combining both existing and new analysis methods, a comprehensive analysis suite was implemented and applied to several problem classes. This allowed to establish a frame of reference for the fitness landscape analysis values and provided convenient instance characterization “at a glance”, using concise box charts. Moreover, these values have been used to derive problem similarities that have been quite successful for the selection of algorithm parameters, for problem

hardness prediction and particularly for algorithm selection, based on sophisticated performance comparison using algorithm dominance series.

In summary, first steps in the practical and large-scale application of fitness landscape analysis methods have been performed which show promising results and suggest the general suitability of theses methods. However, a good deal of work is still ahead of us.

6.2 Open Issues

While previous works have opened the door already for automatized problem understanding. Here, the first few steps using and combining these new possibilities have been made and great potential was discovered. However, several big issues remain to be solved, and many details have to be elaborated. Nevertheless, the goal is within reach. In the following, new directions for exploration and concrete lines of research are proposed to complete the vision of automatic algorithm selection and parameterization as well as increase and automatize structural problem understanding.

6.2.1 Recombination Analysis

One of the bigger open problems is how to practically tackle recombination spaces. While, with the help of P-structures, as detailed in Section 3.12, a first methodology for the formulation of tangible neighborhoods was introduced, further analysis and, in particular, further measures are still missing.

One important aspect that is currently not mentioned is the influence of a population, which is usually involved in recombinatory optimization methods. While P-structures are able to capture recombination itself, they have no notion of the characteristics of the second partner and its relation to the first, such as the fact that typically both parters are from a common population where all individuals are drawn from a shared distribution of fitness values. Morevoer, often these parters will be "in the vicinity" of each other in the fitness landscape, as the population as a whole converges. Therefore, a new recombination analysis could extend the formulation P-structures by including the base fitness values and, therefore, actually limiting the size of the recombination matrix or assign appropriate probabilities according to fitness compatibility.

Another aspect, that has not been analyzed in general terms, is the recombinatory trajectory analysis or genealogical analysis that follows through the development of several generations of recombiatory steps. Here, important answers could be found, not only concerning the problem at hand but also about the examined optimization schemes. Therefore, one could argue that this analysis is actually already outside the scope of pure fitness landscape analysis and should not be carried out in general but rather focus on different algorithmic developments.

Nevertheless, the problem structure as explored by recombination definitely contains important pieces of information as proved by the success of recombinatory methods. Therefore, it seems to be an interesting, albeit difficult, line of research for further investigations of fitness landscapes.

6.2.2 Interfacing Algorithms and Problems

While, for some examples, it has been shown that fitness landscape analysis reveals important information relevant for instance characterization and problem comparison that can be used to understand problem characteristics or to select algorithms or parameters. However, currently a lot of up-front effort has to be invested to arrive at these conclusions. Therefore, another important future direction is not the generalization of these methods, but a more automatic application.

The methods themselves are already reasonably general, however, sufficiently difficult to prevent completely automatic application.

One idea to facilitate this issue, is to introduce additional knowledge to enable more automatic interfacing. For example, for some algorithms certain combinations of parameters might create an easily comprehensible concept, such as selection pressure in genetic algorithms that is controlled by both the selection mechanism and the population size. A comparable concept could be the size of the considered neighborhood in a trajectory based optimization scheme or, for example in the tabu search, in combination with the tabu list length. On the other side, fitness landscape measures could constitute other concepts built by consolidating several measures. For example, ruggedness or isotropy have several concrete measures that could be combined into a single conceptual measure.

Finally, once on both sides these more general and comprehensible concepts have been introduced by the respective experts, i.e algorithm designers define relevant compound properties while fitness landscape experts combine their measures to more expressive notions, these concepts can be mapped by methods similar to those shown in Chapter 4 and relate algorithmic concepts to problem characteristics more naturally than direct matching of the raw values could achieve.

Figure 6.1 illustrates this idea, where instead of blindly correlating arbitrary aspects of both problems and algorithms, more elaborate and carefully chosen concepts are analyzed. While this requires further help form the respective experts, the obtained results would also be more profound and more general. For example, if it was found which problem concept would relate to selection pressure, it would be easy to correctly parametrize new algorithms based on their corresponding notion of selection pressure.

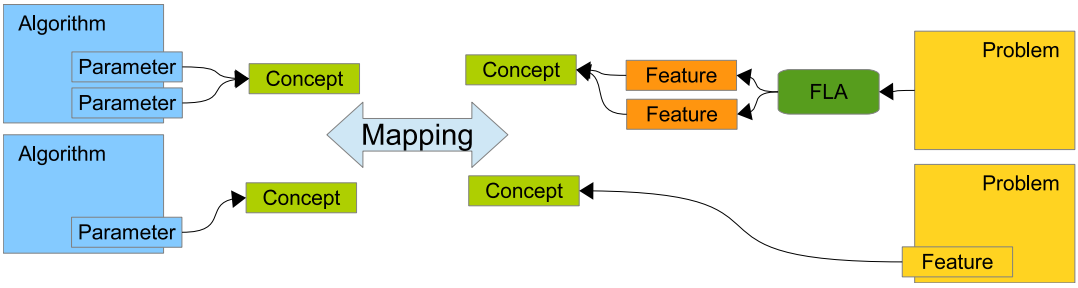


Figure 6.1. Concept Mapping: Instead of directly correlation algorithm and problem properties, intermediate knowledge of algorithm and problem analysis experts can be consolidated into concepts that can then be mapped in more general terms.

6.2.3 Extension and Normalization

Many measures introduced in Chapters 3 and 4 can easily be extended or adapted to yield new slightly angled perspectives. For example, fitness distance correlation could be extended to a multi-dimensional analysis, to relate solution candidates not only to the global optimum but also to several best known solutions. As another example, the fitness cloud could also be extended to perform a general evolvability analysis of the whole neighborhood expansion and not just a certain selection scheme. Also, the generalized formulation of evolvability in Section 3.5.3 provides many leverage points for further analysis. An interesting idea, for example, would be to use the techniques of up-down analysis to explore the whole slopes of local optima in a sort of basin sampling, where a walk starts in both directions up and down until it reaches the extremal point. Then the total number of steps and the distance between the final points can be used as fast estimate of basin sizes and slopes and the corresponding anisotropy.

Another important open issue is the normalization of fitness landscape analysis values. As was

shown, many values correlate strongly with other values. This could mean that they measure, to a certain extend, similar phenomena. In these cases, these methods or their results should be orthogonalized which means that each one should concentrate on a certain aspect that is not also analyzed by another measures. One prominent example is the auto correlation coefficient which correlates almost perfectly with problem size. It might be more interesting to analyze the deviation of this correlation than the actual value itself. Similar effects have been observed for the isotropy measures and their corresponding base measures. One idea for a general and automatic scheme of such normalizations is given in the following section.

6.2.4 Comparability

While fitness landscape analysis is, in principle, very general and can be applied to any problem class regardless of the internal representation, one important aspect has received relatively little attention. Most often, the task of fitness landscape analysis is to provide both a comprehensive but also succinct and tangible description of a problem instance. However, the actual measurements are tied to the particular neighborhood that was used for the analysis and can be very dependent on this neighborhood structure as illustrated in Section 2.2.5.

Different landscape variants, which are the result of examining the same problem instances with different neighborhood definitions, provide very different results at first sight. However, if these measurements are aligned to a frame of reference that has been established for this particular neighborhood relation, the results are quite comparable. This can be seen from Figure 4.17 in Section 4.4.1, where the same analyses are repeated for different landscape variants on the same problems. As these box charts are normalized using the distributions of the whole QAPLIB, a very good frame of reference is established and the indicator colors that reflect the magnitude of each of the measurements within this frame of reference are mostly the same across landscape variants.

In fact, the same method can be used to compare problem instances belonging to different problem classes. However, as a prerequisite, a *universal landscape assay* has to be prepared. This universal assay needs to factorize the effects of different neighborhood structures.

- One easy possibility would be to average over all currently known neighborhoods and provide a unified measurement. As demonstrated in Section 4.4.1, while the actual numbers are different, very often they move in parallel and, therefore, could be meaningfully combined. However, for certain problem classes the number and extent of neighborhoods could be different, wherefore this combination method could lead to a skewed comparison. If care is taken, and the neighborhoods are matched to each other, such that different neighborhood sizes correspond between problem classes, it should be possible and relatively easy to compare across problem classes.

- The second option is to use either a *standard* or a *minimal* neighborhood for all analyses to establish comparability. For example, for many combinatorial optimization problems that use a permutation encoding, the minimal operation could be an exchange of two elements as the smallest possible change and, therefore, the finest grained connected neighborhood. For a real vector encoded problem, the standard neighborhood could be a normally distributed mutation in all dimensions. On the other hand, for some problem instances, the minimal neighborhoods might not be admissible or not even possible in which case the measurements could be skewed again.

- Finally, the most difficult but most unbiased method would be to use only measures which have been standardized with respect to the neighborhood, for example, by dividing through the neighborhood size with respect to the solution space size. The difficulty of this method is twofold:

- On the one hand, the measures themselves have to be analyzed in what way they relate to the neighborhood size and, therefore, in what way they should be normalized.
- On the other hand, the neighborhood size itself has to be determined. Usually, this can be derived analytically with relatively little effort, however, ideally it should be automatically derived. Therefore, an interesting new direction for fitness landscape analysis could be to not only analyze the *fitness* part of fitness landscapes but also focus more on the *landscape* part, i.e. how the landscape is connected. In fact, this lack of general interest in the connectedness is actually reflected in the typically definition of a fitness landscape, where the connectedness $\mathcal{X}$ is actually mentioned but left quite open to interpretation.

This last proposed method, where measurements are standardized using other measurements of the neighborhood structure could then be directly applied to the analysis of arbitrary new problems, where both the fitness structure in relation to the neighborhood structure can be examined. This could then provide the stepping stone for a completely automatic analysis, where nothing has to be known about the internal structure of problems and all relevant aspects can be obtained by stochastic sampling to provide a first shot at an appropriate optimization method.

In the end, as shown in (Poli and Graff, 2009), while there might not be a free lunch, it seems quite possible that we might be able to score a free appetizer, as it was called in (Droste et al., 1999), and get computerized help from fitness landscape analysis in the areas of problem understanding, algorithm selection and algorithm tuning.

Bibliography

Ackley, D. H. (1987). *A connectionist machine for genetic hillclimbing*. Kluwer, Boston.

Affenzeller, M. and Wagner, S. (2005). Offspring selection: A new self-adaptive selection scheme for genetic algorithms. In Ribeiro, B., Albrecht, R., Dobnikar, A., Pearson, D., and Steele, N., editors, *Adaptive and Natural Computing Algorithms*, pages 218–221. Springer Vienna.

Affenzeller, M., Winkler, S., Wagner, S., and Beham, A. (2009). *Genetic Algorithms and Genetic Programming – Modern Concepts and Practical Applications*. Numerical Insights. CRC Press.

Aita, T., Hayashi, Y., Toyota, H., Husimi, Y., Urabe, I., and Yomo, T. (2007). Extracting characteristic properties of fitness landscape from in vitro molecular evolution: A case study on infectivity of fd phage to e.coli. *Journal of Theoretical Biology*, 246:538–550.

Altenberg, L. (1994a). *Advances in Genetic Programming*, chapter The Evolution of Evolvability in Genetic Programming, pages 47–74. M.I.T. Press.

Altenberg, L. (1994b). Evolving better representations through selective genome growth. In *Proceedings of the 1st IEEEConference on Evolutionary Computation*, volume 1, pages 182–187.

Altenberg, L. (1997). *The Handbook of Evolutionary Computation*, volume 1–2, chapter NK Fitness Landscapes, pages B2.7.2:1–11. Oxford University Press.

Angel, E. and Zissimopoulos, V. (1998a). Autocorrelation coefficient for the graph bipartitioning problem. *Theoretical Computer Science*, 191:229–243.

Angel, E. and Zissimopoulos, V. (1998b). On the quality of the local search for the quadratic assignment problem. *Discrete Applied Mathematics*, 82:15–25.

Augerat, P., Belenguer, J., Benavent, E., Corberán, A., Naddef, D., and Rinaldi, G. (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem. Research Report 949-M, Universite Joseph Fourier, Grenoble, France.

Bäck, T. (1996). *Evolutionary algorithms in theory and practice*. Oxford University Press.

Barnett, L. (1997). *Tangled Webs: Evolutionary Dynamics on Fitness Landscapes with Neutrality*. MSc dissertation, School of Cognitive Sciences, University of East Sussex, Brighton.

Barnett, L. (1998). Ruggedness and neutrality—the NKp family of fitness landscapes. In *ALIFE: Proceedings of the sixth international conference on Artificial life*, pages 18–27, Cambridge, MA, USA. MIT Press.

Bateson, W. (1909). *Mendel's principles of heredity*. Cambridge University Press.

Battiti, R., Brunato, M., and Mascia, F. (2008). *Reactive Search and Intelligent Optimization*, volume 45 of *Operations research/Computer Science Interfaces*. Springer Verlag.

Beasley, J. E. (1998). Heuristic algorithms for the unconstrained binary quadratic programming problem. Technical report, Management School, Imperial College, London, UK.

Beham, A., Pitzer, E., and Affenzeller, M. (2011). A new metric to measure distances between solutions to the quadratic assignment problem. In *Proceedings of the IEEE 3rd International Symposium on Logistics and Industrial Informatics (LINDI 2011)*.

Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies - A comprehensive introduction. *Natural Computing*, 1(1):3–52.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Borenstein, Y. and Poli, R. (2005a). Information landscapes. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1515–1522, New York, NY, USA. ACM.

Borenstein, Y. and Poli, R. (2005b). Information landscapes and problem hardness. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1425–1431, New York, NY, USA. ACM.

Borenstein, Y. and Poli, R. (2007). Decomposition of fitness functions in random heuristic search. In *FOGA'07: Proceedings of the 9th International Conference on Foundations of Genetic Algorithms*, pages 123–137, Berlin, Heidelberg. Springer-Verlag.

Box, G. E. P., Jenkins, G. M., and Reinsel, G. (1970). *Time Series Analysis: Forecasting & Control*. Holden Day.

Brandt, H. and Dieckmann, U. (1999). Correlation analysis of fitness landscapes. Working papers, International Institute for Applied Systems Analysis.

Burkard, R. E., Karisch, S. E., and Rendl, F. (1997). QAPLIB - A quadratic assignment problem library. *Journal of Global Optimization*, 10(4):391–403.

Chicano, F. and Alba, E. (2011). Elementary landscape decomposition of the 0-1 unconstrained quadratic optimization. *Journal of Heuristics*, 1:1–18.

Chicano, F., Luque, G., and Alba, E. (2010). Elementary landscape decomposition of the quadratic assignment problem. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO'10, pages 1425–1432, New York, NY, USA.

Chicano, F., Whitley, L. D., and Alba, E. (2011). A methodology to find the elementary landscape decomposition of combinatorial optimization problems. *Evolutionary Computation*, 19(4):597–637.

Christofides, N. and Eilon, S. (1969). An algorithm for the vehicle dispatching problem. *Operational Research Quarterly*, 20(3):309–318.

Christofides, N., Mingozzi, A., and Toth, P. (1979). *Combinatorial Optimization*, chapter The vehicle routing problem, pages 315–338. Wiley.

Collard, P., Verel, S., and Clergue, M. (2004). Local search heuristics: Fitness cloud versus fitness landscape. In Mántaras, R. L. D. and Saitta, L., editors, *European Conference on Artificial Intelligence*, pages 973–974. IOS Press.

Cordeau, J.-F., Gendreau, M., Hertz, A., Laporte, G., and Sormany, J.-S. (2005). New heuristics for the vehicle routing problem. In Langevin, A. and Riopel, D., editors, *Logistics Systems: Design and Optimization*, pages 279–297. Springer US.

Cordell, H. J. (2002). Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Human Molecular Genetics*, 11:2463–2468.

Czech, Z. J. (2008). Statistical measures of a fitness landscape for the vehicle routing problem. In *IEEE International Symposium on Parallel and Distributed Processing, 2008*, pages 1–8.

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.

Davidor, Y. (1990). Epistasis variance: suitability of a representation to genetic algorithms. *Complex Systems*, 4:369–383.

de Abreu, N. M. M., Netto, P. O. B., Querido, T. M., and Gouvea, E. F. (2002). Classes of quadratic assignment problem instances: isomorphism and difficulty measure using a statistical approach. *Discrete Applied Mathematics*, 124:103–116.

de Jong, K. A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, Computer and Communication Sciences, University of Michigan.

Deb, K. and Goldberg, D. E. (1993). Analyzing deception in trap functions. In *Foundations of Genetic Algorithms*, volume 2, pages 93–108.

Deb, K. and Goldberg, D. E. (1994). Sufficient conditions for deceptive and easy binary functions. *Annals of Mathematics and Artificial Intelligence*, 10:385–408.

Deb, K., Horn, J., and Goldberg, D. E. (1993). Multimodal deceptive functions. *Complex Systems*, 7:131–153.

Denzinger, J., Fuchs, M., and Fuchs, M. (1997). High performance ATP systems by combining several AI methods. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI '97)*, pages 102–107.

Derrida, B. (1980). Random energy model: Limit of a family of disordered models. *Physical Review Letters*, 45:79–82.

Droste, S., Jansen, T., and Wegener, I. (1999). Perhaps not a free lunch but at least a free appetizer. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., and Smith, R. E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 833–839. Morgan Kaufmann.

Eiben, A. E. and Smith, J. E. (2003). *Introduction to Evolutionary Computation*. Natural Computing Series. Springer.

Eksioglu, B., Vural, A. V., and Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483.

Fisher, M. (1994). Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research*, 42(4):626–642.

Fonlupt, C., Robilliard, D., and Preux, P. (1997). Fitness landscape and the behavior of heuristics. In *Proceedings of Evolution Artificielle 1997*, pages 321–329.

Fontana, W., Stadler, P. F., Bornberg-Bauer, E. G., Griesmacher, T., Hofacker, I. L., Tacker, M., Tarazona, P., Weinberger, E. D., and Schuster, P. (1993). RNA folding and combinatory landscapes. *Physical Review E*, 47(3):2083–2099.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Gama, J. (2004). Functional trees. *Machine Learning*, 55(3):219–250.

Garrett, D. and Dasgupta, D. (2008). *Learning and Intelligent Optimization*, chapter Multiobjective Landscape Analysis and the Generalized Assignment Problem, pages 110–124. Springer.

Gavrilets, S. (1999). *Evolutionary Dynamics: Exploring the Interplay of Selection, Accident, Neutrality and Function*, chapter Evolution and Specitation in a Hyperspace: The Roles of Neutrality, Selection, Mutation and Random Drift, pages 135–162. Oxford University Press.

Geared, N., Wiles, J., Hallinan, J., Tonkes, B., and Skellet, B. (2002). A comparison of neutral landscapes – NK, NKp and NKq. In *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 205–210. IEEE Press.

Gibbs, W. (1906). *Scientific Papers of J. Willard Gibbs*. Longmans, Green and Company.

Gitchoff, P. and Wagner, G. P. (1996). Recombination induced hypergraphs: A new approach to mutation-recombination isomorphism. *Complexity*, 2(1):37–43.

Glover, F. and Laguna, M. (1997). *Tabu Search*. Kluwer.

Goldberg, D. E. (1987). *Genetic algorithms and simulated annealing*, chapter Simple genetic algorithms and the minimal deceptive problem, pages 74–88. Pitman.

Goldberg, D. E., Deb, K., Kargupta, H., and Harik, G. (1993). Rapid, accurate optimization of difficult problems using fast messy genetic algorithms. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 56–64. Morgan Kaufmann.

Goldberg, D. E., Korb, B., and Deb, K. (1989). Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3:493–530.

Goldberger, J., Roweis, S., Hinton, G., and Salakhutdinov, R. (2005). Neighbourhood components analysis. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, Cambridge, MA.

Golden, B., Raghavan, S., and Wasil, E. (2008). *The vehicle routing problem: latest advances and new challenges*. Operations Research/Computer Science Interfaces. Springer, Dordrecht.

Gouvêa, F. Q. (1993). *p-adic Numbers: An Introduction*. Springer.

Grefenstette, J. J. (1992). Deception considered harmful. In *Foundations of Genetic Algorithms 2*, pages 75–91. Morgan Kaufmann.

Grefenstette, J. J. (1999). Evolvability in dynamic fitness landscapes: A genetic algorithm approach. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 2031–2038.

Gut, A. (2009). *An Intermediate Course in Probability*. Springer, 2 edition.

H. Mühlenbein, D. S. and Born, J. (1991). The parallel genetic algorithm as function optimizer. *Parallel Computing*, 17:619–632.

Hansen, P., Mladenovic, N., and Perez, J. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175:367–407.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Holland, J. H. (1988). *Evolution, Learning and Cognition*, chapter The Dynamics of Searches Directed by Genetic Algorithms, pages 111–127. World Scientific.

Hoos, H. H. and Stützle, T. (2004). *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann/Elsevier.

Hordijk, W. (1996). A measure of landscapes. *Evol. Comput.*, 4(4):335–360.

Hordijk, W. and Stadler, P. F. (1998). Amplitude spectra of fitness landscapes. *J. Complex Systems*, 1:39–66.

Hosmer, D. and Lemeshow, S. (2000). *Applied Logistic Regression*. Wiley.

Hunter, R. S. (1948). Photoelectric color-difference meter. In *Proceedings of the Thirty-Third Annual Meeting of the Optical Society of America*, volume 48, pages 1092–1092. OSA.

Huynen, M. A. (1996). Exploring phenotype space through neutral evolution. *Journal of Molecular Evolution*, 43:165–169.

Huynen, M. A., Stadler, P. F., and Fontana, W. (1996). Smoothness within ruggedness: The role of neutrality in adaptation. *Proceedings of the National Academy of Sciences of the United States of America*, 93:397–401.

Jarque, C. M. and Bera, A. K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 6(3):255–259.

Jolliffe, I. (2002). *Principal Component Analysis*. Springer, 2 edition.

Jones, T. (1995). *Evolutionary Algorithms, Fitness Landscapes and Search*. PhD thesis, University of New Mexico, Albuquerque, New Mexico.

Jones, T. and Forrest, S. (1995). Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 184–192. Morgan Kaufmann.

Judge, G. G., Hill, R. C., Griffiths, W. E., Lütkepohl, H., and Lee, T.-C. (1988). *Introduction to the Theory and Practice of Econometrics*. Wiley.

Kallel, L., Naudts, B., and Schoenauer, M. (1999). On functions with a given fitness-distance relation. In *Proceedings of the 1999 Congress on Evolutionary Computation, 1999*, volume 3, pages 1910–1916.

Karp, R. M. (1972). *Complexity of Computer Computations*, chapter Reducibility Among Combinatorial Problems, pages 85–103. Plenum Press.

Katada, Y. (2006). Estimating the degree of neutrality in fitness landscapes by the Nei's standard genetic distance – an application to evolutionary robotics. In *2006 IEEE Congress on Evolutionary Computation*, pages 483–490.

Katada, Y., Ohkura, K., and Ueda, K. (2004). The Nei's standard genetic distance in artificial evolution. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pages 1233–1239.

Katznelson, Y. (2004). *An Introduction to Harmonic Analysis*. Cambridge University Press, 3rd edition.

Kauffman, S. (1989). Adaptation on rugged fitness landscapes. In Stein, D., editor, *Lectures in the Sciences of Complexity*, volume 1, pages 527–618. Addison-Wesley Longman.

Kauffman, S. A. (1993). *The Origins of Order*. Oxford University Press.

Kesten, H. (2006). What is... Percolation. *Notices of the AMS*, pages 572–573.

Kimura, M. (1968). Evolutionary rate at the molecular level. *Nature*, 217:624–626.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Knowles, J. D. and Corne, D. W. (2002). *Soft Computing Systems: Design, Management and Application*, chapter Towards Landscape Analyses to Inform the Design of a Hybrid Local Search for the Multiobjective Quadratic Assignment Problem, pages 271–279. IOS Press.

Knuth, D. (1997). *The Art of Computer Programming*, chapter Sorting and Searching, pages 144–155. Addison-Wesley, 3rd edition.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

Landwehr, N., Hall, M., and Frank, E. (2005). Logistic model trees. *Machine Learning*, 59(1–2):161–205.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *Operational Research*, 59:345–358.

Larsen, R. J. and Marx, M. L. (2011). *Introduction to Mathematical Statistics and Its Applications*. Pearson, 5 edition.

Lawler, E. L., Lenstra, J. K., Kan, A. H. G. R., and Shmoys, D. B., editors (1985). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley.

Le, M. N., Ong, Y.-S., Jin, Y., and Sendhoff, B. (2009). Lamarckian memetic algorithms: local optimum and connectivity structure analysis. *Memetic Comp.*, 1:175–190.

Liepins, G. E. and Vose, M. D. (1991). Deceptiveness and genetic algorithm dynamics. In Rawlins, G. J. E., editor, *Foundations of Genetic Algorithms (FOGA/CS-90)*, volume 1, pages 36–50.

Lipsitch, M. (1991). Adaptation on rugged landscapes generated by iterated local interactions of neighboring genes. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 128–135.

Ljung, G. M. and Box, G. E. P. (1978). On a measure of a lack of fit in time series models. *Biometrika*, 65(2):297–203.

Lloyd, S. P. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

Locatelli, M. (2003). A note on the griewank test function. *Journal of Global Optimization*, 25(2):169–174.

Manderick, B., de Weger, M. K., and Spiessens, P. (1991). The genetic algorithm and the structure of the fitness landscape. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 143–150.

Marrow, P. (1999). Evolvability: Evolution, computation, biology. In Wu, A., editor, *Proceedings of the 1999 Genetic and Evolutionary Compuation Conference Workshop Program*, pages 30–33.

Mathias, K. and Whitley, L. D. (1992). Genetic operators, the fitness landscape and the traveling salesman problem. *Parallel Problem Solving From Nature*, 2:219–228.

Mattfeld, D. C., Bierwirth, C., and Kopfer, H. (1999). A search space analysis of the sjob shop scheduling problem. *Annals of Operations Research*, 86:441–453.

Merz, P. (2004). Advanced fitness landscape analysis and the performance of memetic algorithms. *Evol. Comput.*, 12(3):303–325.

Merz, P. and Freisleben, B. (1998). Memetic algorithms and the fitness landscape of the graph bi-partitioning problem. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, pages 765–774. Springer.

Merz, P. and Freisleben, B. (2000). Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Transactions on Evolutionary Computation*, 4(4):337–352.

Merz, P. and Freisleben, B. (2001). Memetic algorithms for the traveling salesmen problem. *Complex Systems*, 13(4):297–345.

Mitchell, M. and Forrest, S. (1991). What is deception anyway?: And what does it have to do with gas?: Some concerns inspired by the tanese functions. Technical report, Department of Computer Science, College of Engineering, University of New Mexico.

Moré, J., Garbow, B., and Hillstrom, K. (1981). Testing unconstrained optimization software. *ACM Transactions on Mathematical Software*, 7:136–140.

Moscato, P. (1989). On evolution, search, optimization, genetic algorithms andmartial arts: Towards memetic algorithms. Technical Report C3P Report 826, Caltech Concurrent Computation Program, California Institue of Technology.

Moscato, P. (1999). *New Ideas in Optimization*, chapter Memetic Algorithms: A Short Introduction, pages 219–234. McGraw-Hill, London.

Moscato, P. and Cotta, C. (1991). A competitive and cooperative approach to complex combinatorial search. In *Proceedings of the 20th Informatics and Operations Research Meeting, Buenos Aires*, pages 3.15–3.29.

Munkres, J. R. (2000). *Topology*. Prentice Hall.

Naudts, B. (1998). *Measuring GA-hardness*. PhD thesis, University of Antwerp.

Naudts, B. and Kallel, L. (2000). A comparison of predictive measures of problem difficulty in evolutionary algorithms. *IEEE Transactions on Evoluationary Computation*, 4(1):1–15.

Naudts, B. and Verschoren, A. (1999). Epistasis and deceptivity. *Bulletin of the Belgian Mathematical Society*, 6(1):147–154.

Naus, I. I. (1968). An extension of the birthday problem. *The American Statistician*, 22:27–29.

Nei, M. (1972). Genetic distance between populations. *Amer. Naturalist*, 106:283–292.

Newman, M. E. J. and Engelhardt, R. (1998). Effects of selective neutrality on the evolution of molecular species. *Proc. R. Soc. Lond. B*, 265:1333–1338.

Ochoa, G., Qu, R., and Burke, E. K. (2009). Analyzing the landscape of a graph based hyper-heuristic for timetabling problems. In *GECCO '09*, pages 341–348. ACM.

Ohta, T. (1992). The nearly neutral theory of molecular evolution. *Annual Review of Ecology and Systematics*, 23:263–286.

Ohta, T. (1996). The neutral theory is dead. the current significance and standing of neutral and nearly neutral theories. *BioEssays*, 18(8):673–677.

Ohta, T. (1998). Evolution by nearly-neutral mutations. *Genetica*, 102/103:89–90.

Ozcan, E., Bilgin, B., and Korkmaz, E. E. (2008). A comprehensive analysis of hyper-heuristics. *Intelligent Data Analysis*, 12(1):3–23.

Pitzer, E. and Affenzeller, M. (2011). *Recent Advances in Intelligent Engineering Systems*, chapter A Comprehensive Survey on Fitness Landscape Analysis, pages 167–196. Springer.

Pitzer, E., Affenzeller, M., and Beham, A. (2010). A closer look down the basins of attraction. In *UK Conference on Computational Intelligence*. in press.

Pitzer, E., Affenzeller, M., Beham, A., and Wagner, S. (2011). Comprehensive and automatic fitness landscape analysis using heuristiclab. In *Proceedings of the 13th International Conference on Computer Aided Systems Theory, EuroCAST 2011*.

Pitzer, E., Beham, A., and Affenzeller, M. (2012a). Generic hardness estimation using fitness and parameter landscapes applied to the robust taboo search and the quadratic assignment problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) '12*. ACM.

Pitzer, E., Beham, A., and Affenzeller, M. (2013a). *Advanced Methods and Applications in Computational Intelligence*, chapter Correlation of Problem Hardness and Fitness Landscapes in the Quadratic Assignment Problem, pages 163–192. Springer.

Pitzer, E., Beham, A., and Affenzeller, M. (2013b). Automatic algorithm selection for the quadratic assignment problem using fitness landscape analysis. In Martin Middendorf, C. B., editor, *Evolutionary Computation in Combinatorial Optimization*, Lecture Notes in Computer Science. accepted.

Pitzer, E., Vonolfen, S., , Beham, A., Affenzeller, M., Bolshakov, V., and Merkurieva, G. (2012b). Structural analysis of vehicle routing problems using general fitness landscape analysis and problem specific measures. In *Proceedings of the 14th International Asia Pacific Conference on Computer Aided System Theory*.

Platt, J. (1998). *Advances in Kernel Methods - Support Vector Learning*, chapter Fast Training of Support Vector Machines using Sequential Minimal Optimization, pages 185–208. MIT Press.

Poli, R. and Graff, M. (2009). There is a free lunch for hyper-heuristics, genetic programming and computer scientists. In *Proceedings of the 12th European Conference on Genetic Programming EuroGP 2009*, pages 195–207. Springer.

Provine, W. B. (1989). *Sewall Wright and Evolutionary Biology*. University Of Chicago Press.

Quick, R. J., Rayward-Smith, V. J., and Smith, G. D. (1998). Fitness distance correlation and ridge functions. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, pages 77–86. Springer.

Rechenberg, I. (1973). *Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog.

Reeves, C. R. and Rowe, J. E. (2003). *Genetic Algorithms*. Kluwer.

Reeves, C. R. and Wright, C. C. (1995). Epistasis in genetic algorithms: An experimental design perspective. In *Proc. of the 6th International Conference on Genetic Algorithms*, pages 217–224. Morgan Kaufmann.

Reidys, C. M. and Stadler, P. F. (1998). Neutrality in fitness landscapes. *Applied Mathematics and Computation*, 117(2-3):321–350.

Reidys, C. M. and Stadler, P. F. (2002). Combinatorial fitness landscapes. *SIAM Review*, 44:3–54.

Reinelt, G. (1991). Tsplib–a traveling salesman problem library. *ORSA Journal on Computing*, 3(4):376–384.

Rochet, S. (1997). Epistasis in genetic algorithms revisited. *Inf. Sci.*, 102(1-4):133–155.

Rochet, S., Venturini, G., Slimane, M., and Kharoubi, E. M. E. (1998). A critical and empirical study of epistasis measures for predicting GA performances: A summary. In *Artificial Evolution: Third European Conference, AE '97*, pages 275–286. Springer.

Rockmore, D., Kostelec, P., Hordijk, W., and Stadler, P. F. (2002). Fast fourier transform for fitness landscapes. *Applied and Computational Harmonic Analysis*, 12(1):57–76.

Rosenblatt, F. (1961). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books.

Rosenbrock, H. H. (1960). An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184.

Rowe, W., Platt, M., Wedge, D. C., Day, P. J., Kell, D. B., and Knowles, J. (2010). Analysis of a complete dna-protein affinity landscape. *J R Soc Interface*, 7(44):397–408.

Saitou, N. and Nei, M. (1987). The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4(4):406–425.

Schwefel, H.-P. (1995). *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology. Wiley.

Shakhnarovich, G., Darrell, T., and Indyk, P., editors (2005). *Nearest-Neighbor Methods in Learning and Vision*. MIT Press.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656.

Sherrington, D. and Kirkpatrick, S. (1975). Solvable model of a spin-glass. *Physical Review Letters*, 35(26):1792–1795.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2011). Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13.

Smith, T., Husbands, P., Layzell, P., and O'Shea, M. (2002). Fitness landscapes and evolvability. *Evol. Comput.*, 10(1):1–34.

Stadler, P. and Wagner, G. (1998). The algebraic theory of recombination spaces. *Evol.Comp.*, 5:241–275.

Stadler, P. F. (1994). Linear operators on correlated landscapes. *J.Phys.I France*, 4:681–696.

Stadler, P. F. (1995). *Lecture Notes in Physics*, volume 461, chapter Towards a Theory of Landscapes, pages 78–163. Springer.

Stadler, P. F. (1996). Landscapes and their correlation functions. *J.Math.Chem*, 20:1–45.

Stadler, P. F. (2002). *Biological Evolution and Statistical Physics*, chapter Fitness Landscapes, pages 187–207. Springer.

Stadler, P. F. and Grüner, W. (1993). Anisotropy in fitness landscapes. *J. Theor. Biol.*, 165(3):373–388.

Stadler, P. F. and Happel, R. (1995). Random field models for fitness landscapes. Working Papers 95-07-069, Santa Fe Institute, Santa Fe, NM.

Stadler, P. F. and Happel, R. (1999). Random field models for fitness landscapes. *J.Math.Biol.*, 38:435–478.

Stadler, P. F. and Schnabl, W. (1992). The landscape of the travelling salesmand problem. *Phys. Lett. A*, 161:337–344.

Stadler, P. F., Seitz, R., and Wagner, G. P. (2000). Evolvability of complex characters: Population dependent fourier decomposition of fitness landscapes over recombination spaces. *Bull.Math.Bio.*, 62:399–428.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. In *Proc. 3rd Int'l Conference on Genetic Algorithms*.

Taillard, E. D. (1991). Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17:443–455.

Taillard, E. D. (1995). Comparison of iterative searches for the quadratic assingnment problem. *Location Science*, 3:87–105.

Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. Wiley.

Tavares, J., Pereira, F., and Costa, E. (2008). Multidimensional knapsack problem: A fitness landscape analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(3):604–616.

Thompson, R. K. and Wright, A. H. (1996). Additively decomposable fitness functions. Technical report, University of Montan, Computer Science Dept., Missoula, MT, USA.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58(1):267–288.

Törn, A. and Zilinskas, A. (1989). Global optimization. In *Lecture Notes in Computer Science*, volume 350. Springer.

Turney, P. (1999). Increasing evolvability considered as a large-scale trend in evolution. In Wu, A., editor, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference Workshop Program*, pages 43–46.

Uludağ, G. and Uyar, A. Ş. (2009). Fitness landscape analysis of differential evolution algorithms. In *Fifth International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, IEEE*.

Vanneschi, L., Clergue, M., Collard, P., Tomassini, M., and Vérel, S. (2004). Fitness clouds and problem hardness in genetic programming. In *GECCO '04*, volume 3103 of *Lecture Notes in Computer Science*, pages 690–701. Springer.

Vanneschi, L., Tomassini, M., Collard, P., and Vérel, S. (2006). Negative slope coefficient: A measure to characterize genetic programming fitness landscapes. In Collet, P., Tomassini, M., Ebner, M., Gustafson, S., and Ekárt, A., editors, *EuroGP*, volume 3905 of *Lecture Notes in Computer Science*, pages 178–189. Springer.

Vanneschi, L., Tomassini, M., Collard, P., Vérel, S., Pirola, Y., and Mauri, G. (2007). A comprehensive view of fitness landscapes with neutrality and fitness clouds. In Ebner, M., O'Neill, M., Ekárt, A., Vanneschi, L., and Esparcia-Alcázar, A., editors, *GeneticProgramming, 10 European Conference, EuroGP 2007*, LNCS 4445, pages 241–250.

Vanneschi, L., Vérel, S., Tomassini, M., and Collard, P. (2009). Nk landscapes difficulty and negative slope coefficient: How sampling influences the results. In *EvoWorkshops*, volume 5484 of *Lecture Notes in Computer Science*, pages 645–654. Springer.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer.

Vassilev, V. K., Fogarty, T. C., and Miller, J. F. (2000). Information characteristics and the structure of landscapes. *Evol. Comput.*, 8(1):31–60.

Vollmann, T. E. and Buffa, E. S. (1699). The facilities layout problem in perspective. *Management Science*, 12(10):450–468.

Wagner, A. (2005). Robustness, evolvability, and neutrality. *FEBS Letters*, 579:1772–1778.

Wagner, A. (2008). Robustness and evolvability: a paradox resolved. *Proc. R. Soc. B*, 275:91–100.

Wagner, G. and Stadler, P. (1998). Complex adaptations and the structure of recombination spaces. In C.L. Nehaniv, M. I., editor, *Proceedings of the Conference on Semi-Groups and Algebraic Engineering*, pages 96–115. World Scientific.

Wagner, G. P. and Altenberg, L. (1996). Complex adaptations and the evoluation of evolvability. *Evolution*, 50(3):967–976.

Wagner, S. (2009). *Heuristic Optimization Software Systems - Modeling of Heuristic Optimization Algorithms in the HeuristicLab Software Environment*. PhD thesis, Johannes Kepler University, Linz, Austria.

Wagner, S., Kronberger, G., Beham, A., Winkler, S., and Affenzeller, M. (2008). Modeling of heuristic optimization algorithms. In *Proceedings of the 20th European Modeling and Simulation Symposium*, pages 106–111. DIPTEM University of Genova.

Wang, S., Zhu, Q., and Kang, L. (2006). Landscape properties and hybrid evolutionary algorithm for optimum multiuser detection problem. In *International Conference on Computational Science*, pages 340–347.

Ward, Jr., J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 48:236–244.

Wegener, I. and Witt, C. (2005). On the analysis of a simple evolutionary algorithm on quadratic pseudo-boolean functions. *Journal of Discrete Algorithms*, 3:61–78.

Weinberger, E. (1990). Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63(5):325–336.

Weinberger, E. D. (1991). Local properties of kauffman's n-k model, a tuneably rugged energy landscape. *Physical Review A*, 44(10):6399–6413.

Weinberger, E. D. (1996). Np completeness of kauffman's n-k model, a tuneable rugged fitness landscape. Working Papers 96-02-003, Santa Fe Institute.

Welch, B. L. (1947). The generalization of "student's" problem when several different population variances are involved. *Biometrika*, 34(1–2):28–35.

Whittle, P. (1951). *Hypothesis Testing in Time Series*. PhD thesis, University of Michigan.

Williams, J. W. J. (1964). Algorithm 232 - heapsort. *Communications of the ACM*, 7(6):347–348.

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd edition.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution. In *Proceedings of the Sixth International Congress of genetics*, volume 1, pages 356–366.

List of Figures

List of Tables

List of Algorithms

Europass Curriculum Vitae

Personal information

Surname(s) / First name(s)	**Dipl.-Ing. (FH) Erik Pitzer**
Nationality(-ies)	Austrian
Date of birth	7. February 1981
Gender	Male

Current Employment

Dates	February 2009 onwards
Position	research associate
Main Topic	heuristic algorithms, geo databases
Employer	University of Applied Sciences Upper Austria
Dates	February 2009 onwards
Position	teaching assistant
Main Topic	programming paradigms, math, statistics
Employer	University of Applied Sciences Upper Austria

Previous Work Experience

Dates	December 2007 – February 2009
Position	research associate
Main Topic	sequence & assay data stores, bioinformatics, cancer genetics
Employer	Harvard Medical School, Decisions Systems Group, Brigham and Women's Hospital, Boston, MA, USA
Dates	October 2007 – December 2007
Position	teaching assistant
Main Topic	math
Employer	University of Applied Sciences Upper Austria
Dates	July 2004 – August 2007
Position	research associate
Main Topic	bioinformatics, sequence alignment, di novo mass spectrometry
Employer	University of Applied Sciences Upper Austria

Guest Lectures

2011/12	Introduction to Heuristic Optimization, Riga Technical University
2011/12	HeuristicLab Tutorial, Riga Technical University

Research Projects

since 2012	"BioBoost", EU FP7, simulation and optimization of European-wide routing and logistics network for renewable fuels
2011	"NPR", Bridge, simulation-based production optimization in terms of sustainability
2009–2010	"Heureka!", Josef-Ressel-Center, FFG, heuristic optimization and fitness landscape analysis
2007–2009	"BRCA", NIH, USA, large scale data re-integration of micro array measurements, a platform for re-annotation and cross-platform comparison
2005-2007	"PepSeq", FH OÖ, new algorithms for deNovo peptide sequencing
2003-2005	"Biomis", FH OÖ, algorithms for bio databases and sequence analysis

Reviewing

EuroCAST 2013	14th International Conference on Computer Aided Systems Theory
GECCO-UP 2012	1st Workshop on Understanding Problems, Genetic and Evolutionary Computation Conference 2012
EvoSoft 2012	Evolutionary Computation Software Systems, Genetic and Evolutionary Computation Conference 2012
EuroCAST 2011	13th International Conference on Computer Aided Systems Theory
BioSystems	Journal Article Review, 2010
LINDI 2009	2rd IEEE International Symposium on Logistics and Industrial Informatics

Education

Dates	2004 onwards
Title	PhD
Main Subjects	optimization knowledge database,fitness landscape analysis, meta-heuristics
Institution	Johannes Kepler University Linz
Dates	2000 – 2004
Title	Dipl.-Ing. (FH) with distinction [MSc equivalent]
Main Subjects	software engineering
Institution	University of Applied Sciences Upper Austria
Dates	1990 – 1999
Title	High School Certificate (with distinction)
Institution	BRG Ried i.I. (Federal Scientific Grammar School)

Spoken Languages

Mother tongue(s): **German**

	Understanding		Speaking		Writing
Self-assessment *European level*(*)	Listening	Reading	Spoken interaction	Spoken production	
English+	C2 Proficient user	C2 Proficient user	C2 Proficient user	C2 Proficient user	C2 Proficient user
Italian	B2 Independent user	C1 Proficient user	B1 Independent user	B1 Independent user	B2 Independent user
Russian	A2 Basic user	A1 Basic user	A2 Basic user	A2 Basic user	A1 Basic user

(*) *Common European Framework of Reference (CEF) level*

+near native fluency (American English)

Software Engineering Skills and Experience

Database Technologies & Vendors	MSSQL, MySQL, SQLite, Oracle, PostreSQL, PostGIS, Progress, Magma, ObjectStore, db4o
Programming Languages	Python, C/C++, C#, Java, Scheme, R, Smalltalk, SQL, Clojure, Io, Scala, Perl, Assembler, Shell Programming, Ruby, Haskell, Lisp, Prolog, Progress 4GL, Eiffel, Promela, Occam, Ocaml, PL/SQL
Web Development Frameworks	J2EE, Ruby on Rails, Seaside, BEA Weblogic, Django
Operating Systems	UNIX (GNU/Linux [Gentoo, Red Hat, Mandrake, Debian, Ubuntu], GNU/Hurd [Debian], BSD [FreeBSD]), Microsoft Windows (95, 98, NT4.0, 2000, XP, Vista, 7)
Programming Paradigms	Object Oriented, Generic, Procedural, Functional, Logic, Prototype-Based, Declarative, Aspect Oriented
Version Control Tools	Git, Subversion, CVS, RCS
Computer Algebra Systems, Program Verification Tools, Proof Assistants	Mathematica, MatLab, Axiom, Maxima, PVS, ESC/Java2, JML, Theorema, CASL/CATS

Misc.	Free and Open-Source Desktop Software, Emacs, Vim, LaTeX, TeX, Apache, IIS, JabberD, Postfix, Spamassassin, Eclipse, gcc, gdb, valgrind, Cyrus Imapd, Linux Kernel Patching and Server Administration, Microsoft Visual Studio, SNiFF, Squeak, OpenOffice, Microsoft Office

Selected Publications

[1] E. Pitzer, A. Beham, and M. Affenzeller, "Automatic algorithm selection for the quadratic assignment problem using fitness landscape analysis," in *Evolutionary Computation in Combinatorial Optimization* (C. B. Martin Middendorf, ed.), Lecture Notes in Computer Science, 2013. accepted.

[2] E. Pitzer, A. Beham, and M. Affenzeller, *Advanced Methods and Applications in Computational Intelligence*, ch. Correlation of Problem Hardness and Fitness Landscapes in the Quadratic Assignment Problem, pp. 163–192. Springer, 2013. in press.

[3] E. Pitzer, A. Beham, and M. Affenzeller, "Generic hardness estimation using fitness and parameter landscapes applied to robust taboo search and the quadratic assignment problem," in *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion*, GECCO Companion '12, (New York, NY, USA), pp. 393–400, ACM, 2012.

[4] E. Pitzer, S. Vonolfen, , A. Beham, M. Affenzeller, V. Bolshakov, and G. Merkurieva, "Structural analysis of vehicle routing problems using general fitness landscape analysis and problem specific measures," in *Proceedings of the 14th International Asia Pacific Conference on Computer Aided System Theory*, 2012.

[5] E. Pitzer, A. Beham, M. Affenzeller, H. Heiss, and M. Vorderwinkler, "Production fine planning using a solution archive of priority rules," in *Logistics and Industrial Informatics (LINDI), 2011 3rd IEEE International Symposium on*, pp. 111 –116, aug. 2011.

[6] E. Pitzer and M. Affenzeller, *Recent Advances in Intelligent Engineering Systems*, ch. A Comprehensive Survey on Fitness Landscape Analysis, pp. 161–190. Springer, 2011.

[7] E. Pitzer, M. Affenzeller, A. Beham, and S. Wagner, "Comprehensive and automatic fitness landscape analysis using heuristiclab," in *Proceedings of the 13th International Conference on Computer Aided Systems Theory, EuroCAST 2011*, 2011.

[8] E. Pitzer and M. Affenzeller, "A closer look down the basins of attraction," in *2010 UK Workshop on Computational Intelligence*, 2010.

[9] E. Pitzer, J. Kim, K. Patel, P. A. Galante, , and L. Ohno-Machado, "Positionmatcher: A fast custom-annotation tool for short dna sequences," in *AMIA Summits Transl Sci Proc*, pp. 25–29, 2010.

[10] E. Pitzer, R. Lacson, C. Hinske, J. Kim, P. A. F. Galante, and L. Ohno-Machado, "Towards large-scale sample annotation in gene expression repositories.," *BMC Bioinformatics*, vol. 10 Suppl 9, p. S9, 2009.

[11] R. Lacson, E. Pitzer, C. Hinske, P. Galante, and L. Ohno-Machado, "Evaluation of a large-scale biomedical data annotation initiative.," *BMC Bioinformatics*, vol. 10 Suppl 9, p. S10, 2009.

[12] E. Pitzer, A. Masselot, and J. Colinge, "Assessing peptide de novo sequencing algorithms performance on large and diverse data sets.," *Proteomics*, vol. 7, pp. 3051–3054, Sep 2007.

[13] E. Pitzer, "Acceleration of Progressive Multiple Sequence Alignment by Parallelization and Complexity Reduction of Existing Algorithms," Master's thesis, University of Applied Sciences Hagenberg, Austria, 2004.