A survey of techniques for characterising fitness landscapes and some possible ways forward

Katherine M. Malan *, Andries P. Engelbrecht

Department of Computer Science, University of Pretoria, South Africa

ARTICLE INFO

ABSTRACT

Real-world optimisation problems are often very complex. Metaheuristics have been successful in solving many of these problems, but the difficulty in choosing the best approach can be a huge challenge for practitioners. One approach to this dilemma is to use fitness landscape analysis to better understand problems before deciding on approaches to solving the problems. However, despite extensive research on fitness landscape analysis and a large number of developed techniques, very few techniques are used in practice. This could be because fitness landscape analysis in itself can be complex. In an attempt to make fitness landscape analysis techniques accessible, this paper provides an overview of techniques from the 1980s to the present. Attributes that are important for practical implementation are highlighted and ways of adapting techniques to be more feasible or appropriate are suggested. The survey reveals the wide range of factors that can influence problem difficulty, emphasising the need for a shift in focus away from predicting problem hardness towards measuring characteristics. It is hoped that this survey will invoke renewed interest in the field of understanding complex optimisation problems and ultimately lead to better decision making on the use of appropriate metaheuristics.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

Metaheuristics have become popular for solving complex optimisation problems where classical optimisation methods are either infeasible or perform poorly. Despite many success stories, it is well known that on some problems these techniques fail and that there is in fact very little understanding of which algorithms, or algorithm variants, are in general more suitable for solving which kinds of problems. It is also true that no one optimisation algorithm is at all times superior to the other. This was shown theoretically by Wolpert and Macready with their famous 'No-Free-Lunch' theorems for search/optimisation [87,88]. In the case of simple hill-climbing algorithms it is relatively straightforward to estimate which problems will be easy and which will be harder to solve. However, in the case of more complex metaheuristics, it is not as easy to predict the degree to which problems will present difficulties for algorithms. As expressed by Culberson [12]: *"The researcher trying to solve a problem is then placed in the unfortunate position of having to find a representation, operators and parameter settings to make a poorly understood system solve a poorly understood problem. In many cases he might be better served concentrating on the problem itself"*. This article focuses on ways of better understanding problems in the hope that practitioners and researchers will have better guidance in the use of appropriate algorithms.

Many attempts at characterising optimisation problems have focused on finding a measure that could divide problems into those that are easy and those that are hard to solve [30,48,23]. These attempts have not been very successful. In the

---

* Corresponding author. Tel.: +27 124203618.
  E-mail addresses: kmalan@cs.up.ac.za (K.M. Malan), engel@cs.up.ac.za (A.P. Engelbrecht).

literature, whenever a publication appears proposing some measure of problem hardness, a number of subsequent publications can usually be found with counter-examples for which the proposed hardness measure does not hold. Some authors even provide counter-examples to their own techniques, pre-empting the inevitable 'counter-paper'.

Much of the earlier work done on predicting problem hardness assumed genetic algorithms with the resulting notions of GA-hard and GA-easy problems [14,28,33,47]. As pointed out by Guo and Hsu [23] *"Any efforts like this are doomed to fail"*, because the class of GA algorithms is too broad. The same problem can change from a hard problem to an easy problem by changing the GA settings. If finding a GA hardness measure is infeasible, then surely finding a general problem hardness measure is infeasible? Even assuming such a general difficulty measure could be found, He et al. [24] have proved that a predictive version of such a measure, i.e. that runs in polynomial-time, cannot exist (unless P = NP or BPP[1] = NP). The general agreement in literature seems to be that no satisfactory problem difficulty measure for search heuristics has been found [30,23,24].

A possible reason for this is that although there are many factors (such as deception, ruggedness, and non-linear separability) that clearly affect problem difficulty, no one factor appears to be necessary or sufficient for characterising problem hardness. For example, modality, although an important consideration, cannot be used as the only estimate of complexity for search algorithms. Horn and Goldberg [28] show that there are problems with minimal modality (such as long path problems) that are hard for a GA to optimise and that there are problems with maximum modality (such as their one-max function with "bumps") that are easy for a GA to optimise. Kallel et al. [34] confirm that multimodality is neither necessary nor sufficient as a predictor of difficulty for both hill-climbers and genetic algorithms. Forrest and Mitchell [18] studied GA failure showing that some GA-deceptive problems are easy for a GA and that there are non-deceptive problems that are difficult for a GA. They conclude that GA-deception is only one factor that contributes to the difficulty of search for a GA. It has therefore become widely accepted, as expressed by Smith et al. [64] that *"No single measure or description can possibly characterise any high-dimensional heterogeneous search space"*.

Instead of trying to find one measure of hardness, a more realistic approach could be to determine the characteristics of a problem and then use these characteristics to determine which algorithm would be best suited to solving that problem. What is hard for a Particle Swarm Optimisation (PSO) algorithm to solve might not necessarily be hard for a Genetic Algorithm (GA) to solve, or even a PSO with different parameter settings. It is hoped that in analysing problems in more depth, it will become possible to distinguish problems based on their characteristics.

This paper addresses the topic of characterising optimisation problems. The aims are to, firstly, discuss characteristics of problems that could potentially make them hard to solve and, secondly, to provide an overview of existing techniques for analysing these problem characteristics. Section 2 starts with an overview of different views of fitness landscapes. Although the term 'fitness landscape' is used frequently in literature, it can have different meanings in different contexts and these are elaborated on in Section 2. Section 3 provides a summary of different features of optimisation problems that could potentially affect the difficulty in solving the problem. The most important contribution of this work is in Section 4 where a survey is provided of existing techniques to characterise optimisation problems from the 1980s to the present. Important features are highlighted such as the focus, the level of search independence, assumptions on which the technique is based, and the result produced by the technique. The paper concludes with suggestions on how research in this area can move forward.

2. Fitness landscapes

For most optimisation problems there is a fitness function[2] that reflects the objectives of the problem to be solved. (Problems that do not have a readily available fitness function are excluded from this study.) Potential solutions to a problem are compared based on their fitness values, which are determined using the fitness function. In some problem cases, the function is expressed so that the aim is to find the solution that maximises the fitness value and in other cases, the aim is to find the solution that minimises the fitness value.

There are many ways of analysing fitness functions, such as epistasis variance [14] and the density of states [62] and these are discussed further in Section 4. However, more interesting analyses can be performed, when a fitness function is extended into a fitness landscape, by introducing some form of topology onto the search space. Although the term 'fitness landscape' with its associated notions of 'peaks' and 'valleys' is widely used in many contexts and in academic writing, there is often a lack of understanding of what precisely is meant by the term. This section summarises some of the contributions towards understanding and formalising fitness landscapes.

Wright [89] introduced the notion of a fitness landscape (which he called a surface of selective values) for genetic evolution back in 1932, with further invited commentary on his seminal paper published 56 years later [90]. He proposed an abstract space where genotypes are packed, side by side, in a two-dimensional space in such a way that each is surrounded by genotypes that differ by only one gene replacement. He used contour lines to indicate fitness values and in this way illustrated the peaks and valleys in a two dimensional diagram, as shown in Fig. 1. Notice that the 'axes' of the diagram are not real axes, as there are no defined units or labels. In his own words, such a representation *"is useless for mathematical*

---

[1] BPP: bounded-error probabilistic, polynomial time [83].

[2] Note that this study does not restrict the notion of fitness and fitness function to the meaning of fitness in the evolutionary sense, but rather to the broader notion of an objective and objective function to be optimised by an algorithm.
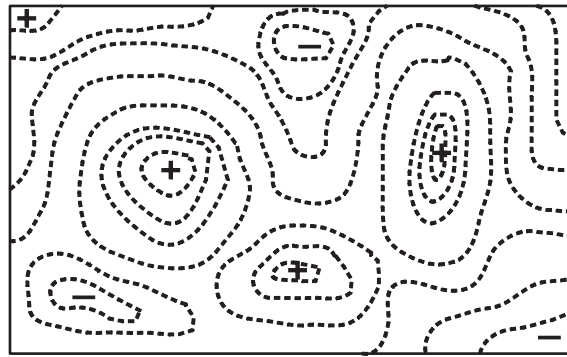
Fig. 1. Adaptation of Wright's (1932) fitness landscape, which he called a two-dimensional 'surface of selective values'.

purposes" [90]. Wright's aim with this representation was to provide an intuitive picture of evolutionary processes taking place in high dimensional space and not to provide any kind of formal model for analysis. Despite a lack of formal definition, this same basic fitness landscape metaphor with its associated 'valleys', 'peaks', 'ridges' and 'plateaus' has been used extensively within multiple disciplines to understand and explain complex systems.

An alternative, more formal view of a fitness landscape particular to search algorithms, is to define a landscape as a directed graph, where nodes correspond to solutions. Two nodes in the fitness landscape graph are neighbours if one solution can be reached from the other through a single step of a search operator (such as mutation or crossover in the case of a GA). Jones [32] introduced such a model and argued for the view of *"one operator, one landscape"*, where each search operator defines its own fitness landscape. In his model, the fitness value (or 'height' in the fitness landscape metaphor) is indicated as a label attached to each node in the graph and probabilities of the step occurring are attached to edges of the graph.

In Jones' [32] study of fitness landscapes it is assumed that the landscape is discrete (or combinatorial). Stadler [66] provides a more general view of landscapes as consisting of three elements:

1. A set X of configurations (solutions to the problem),
2. a notion $\mathcal{X}$ of neighbourhood, nearness, distance, or accessibility on X, and
3. a fitness function $f : X \rightarrow \mathbb{R}$.

This description can be used in the case of both discrete and continuous landscapes. For example, in a discrete landscape, $\mathcal{X}$ could be described as a notion of neighbourhood specified using a crossover operator, or using a more generic notion of neighbourhood, such as Hamming distance. For a continuous landscape, $\mathcal{X}$ can be described in terms of a distance metric, such as Euclidean distance or using some gradient-based search strategy for determining accessibility to a continuous subset of configurations.

In many studies, the term fitness landscape is used to refer to a problem encoding in combination with a fitness function (elements 1 and 3 in Stadler's description above). In these cases it is usually assumed that the notion of neighbourhood/distance is based on some 'natural' notion of order/distance. For example, in binary-encoded problems, Hamming distance is often assumed as the neighbourhood relationship: any two points are neighbours if their Hamming distance is 1. In real-encoded problems, Euclidean distance is usually assumed as the metric on which the fitness landscape is defined. In some representations, it is not as obvious to define neighbourhood. For example, when solutions are in the form of trees, there is no obvious way of deciding when two tree solutions are neighbours. In these cases, neighbourhood is often defined in terms of a particular search operator/strategy: two solutions are regarded as neighbours if it is possible to move from one to the other via a single application of the search operator.

Because a fitness landscape is defined using a particular notion of neighbourhood/distance, the same fitness function can generate many different fitness landscapes. For example, in Wright's [89] fitness landscape, a genotype is a neighbour of another genotype if they differ by a single gene. If instead, neighbourhood was defined based on a k-bit-flip mutation operator, a very different fitness landscape may result. The number of different possible fitness landscapes is dependent on the fitness distribution [7]. A constant fitness function, for example, has only one possible fitness landscape. The fitness landscape is therefore not a feature of the problem *per se*, but rather a feature of the encoding of the problem, the fitness function, and of the notion of neighbourhood/distance used to define the landscape. Landscape theory serves as a reminder that other possibilities may exist, beyond the obvious ones, for defining landscapes and exploring and analysing optimisation problems.

3. Features of fitness functions and landscapes

This section summarises a number of features of optimisation problems that could influence the ability of algorithms to solve the problems. The features listed are not in any way exhaustive. There may be features not known or not mentioned

here, which could influence the behaviour of optimisation algorithms. The purpose is to summarise those features that are commonly discussed in literature. Measuring or quantifying these features is not always straight-forward and this is discussed further in Section 4. The first three features (degree of variable interdependency, noise and fitness distribution) are features of the fitness function alone (without any defined fitness landscape). The remaining features are based on fitness landscapes. Although divided into separate subsections, many of these features are related to one another.

3.1. Degree of variable interdependency (including epistasis)

In genetics, epistasis refers to the degree of dependency between genes in a chromosome for expression [14]. If genes contribute independently to the overall fitness of the chromosome then the system has low epistasis. On the other hand, if the fitness contributions of genes depends on the values of the other genes, the system has high epistasis. In general, for optimisation problems, this characteristic can be referred to as the degree of interdependency between variables (also known as non-linear separability). When variables in an optimisation problem are dependent on each other, this means that it is impossible to tune one variable to find the optimal value independently of the others. For example, if different variables in a mathematical expression of a fitness function are separated by addition, then the variables contribute independently to the fitness. However, if different variables are combined in a term through multiplication, then these variables must cooperate in order to contribute to fitness; if either variable has a low value, then the product may be low even if the other variable has a high value. It is seldom as simple as in this example. In complex problems, the interactions between variables can take many different forms. Studies have shown that linearly separable functions are easier for genetic algorithms to solve than non-linearly separable functions [63,13]. Naudts and Naudts [49] argue that it is the type of interaction (functions with first and second order dependencies) rather than the amount of higher order interaction that influences the difficulty of the problem for search algorithms. Measures for quantifying epistasis include epistasis variance [14] (Technique 5 in the survey), the site-wise optimisation measure [49], and bit-wise epistasis [16] (Technique 12 in the survey).

3.2. Noise

Noisy objective functions are common in many real-world optimisation problems. Levitan and Kauffman [39] studied the effect of noise on hill-climbing algorithms and found that although certain types and levels of noise had a negative effect on the ability of the algorithm to search well, small amounts of noise could help the algorithm to perform better than in the absence of noise. It is a common belief that evolutionary algorithms work well in noisy environments, but Beyer [6] has shown that this is not necessarily the case. A common way of reducing the effects of noise during search is to resample data points and average over a number of fitness evaluations [58]. Similarly, to detect noise in a fitness function, data points can be resampled. Some measure of difference (such as variance or standard deviation) could then be used to quantify the level of noise in multiply sampled data points.

3.3. Fitness distribution

A statistical analysis of fitness function values can provide some information on the problem at hand. For example, the distribution of fitness values (the frequency with which each fitness value occurs) can be used to provide a profile of the problem, as is done with the density of states technique [62] (Technique 9 in the survey). In most cases the fitness distribution of a problem cannot be exactly determined and has to be estimated, based on some sampling and grouping strategy.

3.4. Fitness distribution in search space

Given a fitness landscape of a problem, a simple way of characterising the problem is to measure, in some way, how the fitness values are distributed across the search space. This differs from simple fitness value statistics, because the position of fitness values within the search space is taken into account. Techniques for quantifying fitness distribution layout in binary landscapes include the HDIL (Hamming Distance In a Level) and HDBL (Hamming Distance Between a Level) measures [4] (Technique 13 in the survey).

3.5. Modality and the landscape structure of optima

Unimodal functions have only one local optimum, which is also the global optimum. Multimodal functions have more than one local optimum. Horn and Goldberg [28] define a local optimum as a point or region (a set of interconnected points with equal fitness) with fitness function value greater than those of all its nearest neighbours. This definition would consider flat plateaus and ridges as single optima. Local optima are obstacles for local search algorithms in finding the global optimum because there is a lack of information in the neighbourhood to direct search out of the local optima.

Other than the number of optima, the distribution of basin sizes and the depth (or height) of the basins is a factor that may be more important in determining landscape difficulty [34]. Local optima with relatively small basins of attraction are called isolated. An extreme example of an isolated landscape could be a needle-in-a-haystack binary encoded maximisation problem, where the fitness value is 1 for one arbitrary bit string and 0 elsewhere. A less extreme example would be a

landscape where the local optima have large basins of attraction and the global optima has a smaller basin of attraction. Rana [57] studied the effect of multimodality on GA performance and found that although the number of local optima did not always affect GA behaviour, highly fit local optima, particularly with large basins of attraction, did present a problem for GA search. Kinnear [38] found that in the case of genetic programming, landscape basin depths showed a good correlation with problem difficulty over a range of problems.

Techniques related to modality and landscape structure of optima include: Garnier and Kallel's [19] technique for estimating the number and distribution of local optima and Merz's [45] escape rate measure for estimating the sizes of basins of local optima in the fitness landscapes of combinatorial problems. In addition, Ochoa et al. [53] have proposed a technique for compressing the essential landscape features for combinatorial optimisation problems into a graph called a local optima network. This graph-based model serves as a characterisation of the structure of a landscape and the distribution of local optima.

3.6. Information to guide search and deception

Some problems result in fitness landscapes that are structured in such a way that they guide search algorithms more easily towards the global optima. Both the quantity and quality of information available is important [8]. In other words, for an algorithm to perform well, the fitness landscape should not only provide sufficient information to guide the search, but the information should also guide the search in the right direction. The presence of misleading information is sometimes known as deception. Deception is clearly related to the landscape structure of optima. The positions of sub-optima in relation to the global optimum and the presence of isolation will have an effect on the level of deception. Deception only has meaning with reference to a particular search algorithm. A problem that is deceptive to a GA would not necessarily be deceptive to a PSO algorithm. Measures of deception include GA-deception [20,21,15] (Technique 1 in the survey) and the deceptiveness coefficient [31]. Xin et al. [91] studied the notion of deception for PSOs and concluded that the relative size of basins of attraction (local versus global) was the most important factor related to deception for PSOs.

3.7. Global landscape structure (funnels)

A funnel in a landscape is a global basin shape that consists of clustered local optima [67]. Fig. 2 shows two one-dimensional minimisation benchmark problems. Fig. 2a shows the Rastrigin function as an example of a single-funnel landscape. Although Rastrigin is clearly multimodal, there is a distinct underlying unimodal structure, indicating the presence of a single funnel. Fig. 2b illustrates the Schwefel 2.26 function, which is an example of a multi-funnel landscape. The exact number of funnels in Schwefel 2.26 would depend on the precise definition of a funnel. Multi-funnel landscapes can present problems for search, particularly in the case of algorithms that rely on local information, as they may become trapped in sub-optimal funnels [67,91]. A technique for estimating the presence of funnels in a fitness landscape is Lunacek and Whitley's dispersion metric [42] (Technique 19 in the survey).

3.8. Ruggedness and smoothness

Ruggedness refers to the number and distribution of local optima. It therefore has to do with the level of variation in fitness values in a fitness landscape. If neighbouring points have very different fitness values, then the result is a rugged landscape. The opposite of a very rugged landscape would be a landscape with a single large basin/peak of attraction or a flat landscape with no features. In general, search algorithms struggle to optimise very rugged landscapes, because the
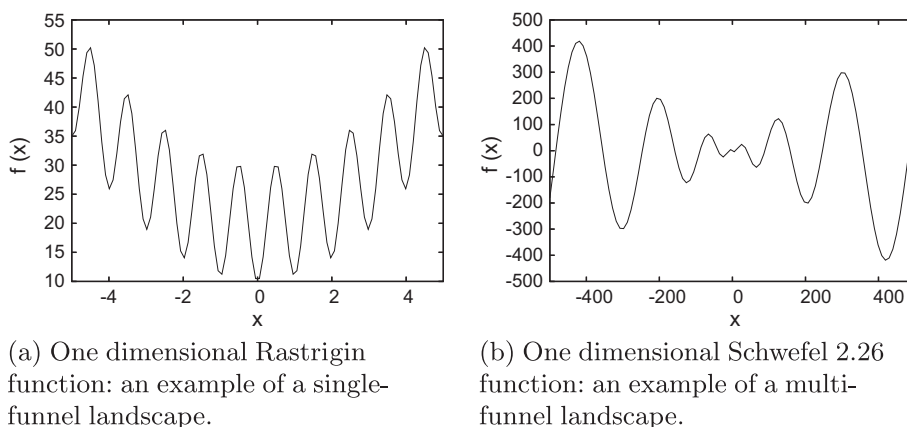

(a) One dimensional Rastrigin function: an example of a single-funnel landscape.

(b) One dimensional Schwefel 2.26 function: an example of a multi-funnel landscape.

Fig. 2. Two sample minimisation benchmark problem landscapes.

algorithms can get trapped in local optima. Kauffman [35,36] introduced a model of binary fitness landscapes with tunable ruggedness, called NK landscapes, where the value of N specifies the number of variables and K can be set to determine the level of ruggedness. The NK landscapes have been used extensively in studies of ruggedness and the link to problem difficulty. Techniques for measuring ruggedness include adaptive walks [37], autocorrelation measures [84,44] (Techniques 2 and 3 in the survey), correlation length [40] (Technique 4 in the survey), entropic measures [78,79,43] (Technique 10 in the survey) and amplitude spectra [27] (Technique 11 in the survey).

A smooth landscape is one where neighbouring points have nearly the same fitness value [36]. Smoothness also relates to the size of the basins of attraction. A landscape is smooth if the number of optima is low and the optima have large basins of attraction [79]. A technique for quantifying landscape smoothness is the second entropic measure by Vassilev et al. [78,79] (Technique 10 in the survey).

3.9. Neutrality

Neutrality is present in a landscape when neighbouring points have equal fitness values. A discrete landscape is regarded as neutral if a substantial fraction of adjacent pairs of solutions are neutral [60]. A neutral landscape therefore does not imply a flat landscape (where the function is constant), but rather the presence of successive neutrality, which can manifest in features such as plateaus and ridges in a landscape. Neutrality can also feature in continuous fitness landscapes as regions of equal or nearly equal fitness (for an investigation into this topic of neutrality in continuous domains see [29]). Neutrality is a feature which is often ignored, but can have a profound effect on the number and distribution of local optima [17] and on the success of search algorithms [64]. During search when a population moves through a neutral portion of a fitness landscape, this could be misinterpreted as convergence on a local optimum. Since the fitness values are not changing, it may seem as if the population is stagnating, when in fact the population is moving across a neutral area. In a study of neutral landscapes Beaudoin et al. [3] found that neutrality had a smoothing effect on problem difficulty in that adding neutrality to a deceptive landscape made the problem easier, whereas adding neutrality to an easy landscape made it harder (as measured by the fitness distance correlation difficulty metric [33]). Techniques that measure landscape neutrality include neutral walks [60] (Technique 14 in the survey) and neutral network analysis [73,75] (Technique 20 in the survey). Verel et al. [81] show how local optima networks [53] can be extended to analyse the structure of neutral combinatorial fitness landscapes, but this approach currently requires a full enumeration of the search space.

3.10. Symmetry

Symmetry in a fitness function or landscape leads to multiple points with the same fitness values, so in some way partitions the search space into large equivalence classes [85]. There are many different forms of symmetry. For example, if a fitness landscape is symmetrical with respect to one of the axes this is known as axial bias. A fitness landscape is symmetrical with respect to an optimum if the fitness value of all points a set distance away from the optimum is the same regardless of the direction of the point. Some forms of symmetry are a feature of the fitness function alone. Van Hoyweghen and Naudts [69] define simple types of symmetry for discrete representations, such as symmetry on string positions (where a permutation on string positions results in no change to the fitness) and symmetry on the alphabet (e.g. spin-flip symmetry, where a binary string and the binary complement have the same fitness value). There are conflicting studies on the effect of symmetry on search. Whitley et al. [85] note several research findings where the presence of symmetry in functions results in failure for certain genetic algorithms. Naudts and Naudts [49] also show that the presence of symmetry can have a negative effect on the ability of a simple GA to converge. This could be due to the phenomenon where two dissimilar good (symmetrical) solutions, crossed over, result in inferior children. Other studies have shown that genetic algorithms show improved performance on landscapes with axial biases [13] and that a rotation of the coordinate system for such problems (resulting in the loss of symmetry) causes severe algorithmic performance loss [63].

3.11. Evolvability/Searchability

Evolvability can be loosely defined as the capacity to evolve [68]. Altenberg [1] describes evolvability with particular reference to genetic algorithms as the ability of a population to produce offspring that are fitter than their parents. Although the notion of evolvability is related to the algorithm's ability to evolve the population and is therefore primarily a performance measure of an algorithm, it can also be viewed as a characteristic of a fitness landscape in terms of a particular search operator/strategy. The evolvability of a fitness landscape is defined in this study as the ability of a given search process to move to a place in the landscape of better fitness and is henceforth referred to as *searchability*. Note that this definition broadens the scope of evolvability beyond evolutionary based algorithms to encompass any search process. Searchability is a characteristic of problems that only has meaning with reference to a particular search strategy. A problem that has high searchability in terms of one algorithm, may exhibit low searchability with reference to another algorithm. Fitness landscape analysis techniques that focus on evolvability include fitness evolvability portraits [64] (Technique 15 in the survey), fitness clouds [80] (Technique 16 in the survey), fitness-probability clouds [41] (Technique 21 in the survey) and accumulated escape probability [41] (Technique 22 in the survey).

3.12. Discussion

In the subsections above, a number of characteristics of fitness functions and landscapes were discussed. Many of these characteristics are related to each other. For example, modality and the structure of optima are clearly related to ruggedness, smoothness and neutrality of the landscape. Also, if a function has a high degree of variable interdependency, then this will probably affect the ruggedness of an associated landscape. Vassilev et al. [79] claim that the ruggedness, smoothness and neutrality alone can fully characterise a fitness landscape. Although this may be true, there could still be value in viewing a problem through a deception 'lense', or through a funnel 'lense' or through any other viewpoint that could shed light on the nature of the problem to be solved. The aim of this study is to work with a number of these characteristics together to form a more comprehensive view of the problem rather than limiting the focus to one viewpoint.

4. Measures and techniques for analysing fitness landscapes

For low dimensional problems, the associated fitness landscape could be visualised. A graphical representation could then give some indication of the features of the problem to be solved. Two problem landscapes could be compared in terms of ruggedness, deception, neutrality, etc. simply through visual inspection. In reality, however, problems are too complex to be visualised, so some other way of analysing problem characteristics is needed. The ideal would be to have a single numerical measure of difficulty for every problem. Given the issue with problem 'hardness' as described in Section 1, it is unrealistic to find such a single measure. Instead, it is proposed that problems should be characterised through multiple viewpoints and that this hopefully will provide sufficient insight into the problem so as to facilitate informed decisions regarding the approach used to solve the problem.

This section provides an overview of techniques used to characterise optimisation problems from the 1980s to the present. The techniques are summarised in Table 1, sorted in chronological order. Before presenting the table, the kinds of measures not included in this survey are described.

4.1. Types of measures not included in this survey

There are many reasons for characterising optimisation problems. Some measures are performed during execution of algorithms with the aim of adapting the algorithms on the fly. In other cases problems are studied and characterised to try to explain unexpected algorithmic behaviour in retrospect. In yet other cases the motivation is to divide problems into theoretical complexity classes. Jansen [30] distinguishes between two types of classifications of fitness functions: descriptive and analytical. A descriptive classification is one where classes of fitness functions are defined with some common property, whereas an analytical classification is a technique that takes a fitness function and produces a classifying attribute as output. For example, Naudts and Kallel [48] provide a precise definition for the class of site-wise optimisable fitness functions (a descriptive classification) and also define the site-wise optimisation measure as a measure of epistasis (an analytical classification). All techniques for classifying problems considered in this study are analytical. The aim is to obtain *a priori* information on the problem to help guide the choice of appropriate (or possibly not inappropriate) algorithms to solve the problem, in a less computationally-intensive way than actually solving the problem. In this section, three types of measures not included in this study are discussed. These are termed *theoretical measures*, *dynamic measures* and *retrospective measures*.

4.1.1. Theoretical measures

There are some estimators of problem complexity or difficulty that are theoretical in nature. Such measures, which cannot be practically implemented, are not discussed in this overview. An example of this is Kolmogorov complexity (KC). KC, also known as algorithmic information theory [22], is a measure of an object that relates to the complexity of the computer program required to produce that object and then halt. A discrete fitness function defined over a finite space can be described by a single binary string consisting of all possible output values of the function. The KC of this string is expected to capture the difficulty of the function [10]. Although this approach to using KC to quantify function complexity has been used extensively in theoretical studies and proofs, particularly in relation to the no-free-lunch theorems for search/optimisation, the KC of a problem cannot be computed [10], and is therefore not studied further.

4.1.2. Dynamic measures

Dynamic measures are those that are measured during execution of an optimisation algorithm and are typically used as a basis for adapting the search algorithm on the fly. Examples of such measures include the following:

- The correlation coefficient by Manderick et al. [44], which measures the correlation between two populations during execution.
- Riopka's average bit certainty measure [61], which is used to modify the behaviour of a GA relative to the landscape being searched.
- Generation Rate of Better Solutions (GRBS) measure by Waeselynck et al. [82] that monitors convergence, and
- Merz's escape rate measure [45] performed during the run of a memetic algorithm.

Table 1
Techniques for characterising fitness functions and landscapes.

Technique 1	**GA-deception** by Goldberg [20] with extensions [21,15]
Year	1987
Focus	Deception with respect to a GA (Section 3.6)
Search Independence	Medium: based on genetic operators and schemata, applicable to recombinative algorithms
Assumptions	Assumes knowledge of global optima. Assumes a binary representation
Description	A binary fitness function is expressed as a Walsh polynomial. The Walsh coefficients are then used to calculate schema average fitness values. A set of schema with relatively high fitness are determined and the effect of genetic operators on the fitness of the schema are analysed
Result	Decision on level of GA-deception (strictly deceptive, deceptive, simple, strictly simple)
Technique 2	**Autocorrelation function** by Weinberger [84] with extensions [44,26]
Year	1990
Focus	Ruggedness (Section 3.8)
Search Independence	High: based on random walks through a binary fitness landscape
Assumptions	Assumes a discrete landscape and that the landscape is statistically isotropic, meaning that the statistics of a random walk on a landscape will be the same, regardless of the starting position
Description	From a sequence of fitness values, obtained from a random walk through the fitness landscape, calculate the correlation with the same sequence of values a small distance away. Do this for all possible landscapes
Result	Plot of autocorrelation $\rho(s)$ against step size s (distance between sequences being correlated). The value of $\rho(s)$ is in the range $(-1,1)$ where $|\rho(s)| = 1$ indicates maximal correlation and a value close to 0 indicates almost no correlation
Technique 3	**Correlation length** by Weinberger [84] with extensions [44,65,26]
Year	1990
Focus	Ruggedness (Section 3.8)
Search Independence	High: based on random walks through a binary fitness landscape
Assumptions	As for Technique 2. Also assumes that the autocorrelation function is a decaying exponential
Description	Using the autocorrelation function $\rho(s)$ for step size s, calculate the correlation length using the formula: $\tau = -1/\ln(\rho(1))$
Result	A single value (the distance beyond which the majority of points become uncorrelated: a smaller value indicates a more rugged landscape)
Technique 4	**Correlation length** by Lipsitch [40]
Year	1991
Focus	Ruggedness (Section 3.8)
Search Independence	High: based on random walks through a binary fitness landscape
Assumptions	Assumes the problem has a binary representation
Description	Given 600 random initial points in the search space, calculate the standard correlation coefficient (c_i) between the fitness of points and the fitness of each of 30 i-mutant neighbours of the points. The correlation length is one less than the value of i at which c_i first becomes non-positive
Result	A single value (from 0 to 30, inclusive), where smaller values are indicative of a more rugged landscape
Technique 5	**Epistasis variance** by Davidor [14] with extensions [59,50,47]
Year	1991
Focus	Epistasis (Section 3.1)
Search Independence	Complete: based on fitness function alone
Assumptions	Assumes a binary representation
Description	A measurement of epistasis is calculated based on a linear composition of a string solution from its bits. The level of inaccuracy (the epistasis variance) of the linear decomposition of the function is used as an estimate of the amount of non-linearity in the function
Result	A single value (from 0 to a non-normalised positive number), where 0 indicates no dependency between genes
Technique 6	**Formae variance** by Radcliffe and Surry [56]
Year	1994
Focus	Fitness variance of formae (Section 3.3)
Search Independence	Medium: based on evolutionary notion of formae
Assumptions	Assumes a discrete representation
Description	Given a discrete fitness function and a sample of randomly generated formae (generalised schemata) at each order, calculate the variance of fitness values for each forma order. The premise is that lower variance will provide more exploitable information for evolutionary search algorithms
Result	Plot of fitness variance of formae against forma order, where a plot in which variance falls more quickly is indicative of more exploitable information
Technique 7	**Fitness distance correlation** and scatter plots by Jones and Forrest [33] with extensions [32,2,70]
Year	1995
Focus	Deception with respect to local search (Section 3.6)
Search Independence	High: uses Hamming distance as basis of measure
Assumptions	Requires knowledge of global optima. Assumes the existence of a measure of distance between solutions
Description	Given a random sample of points in the search space, each point i generates a pair (f_i, d_i), where f_i is the fitness of point i and d_i is the distance of point i to the nearest global optimum. The fitness distance correlation is calculated as the correlation coefficient of this set of (fitness, distance) pairs
Result	A single correlation value r (between -1 and $+1$, inclusive), where for maximisation problems, low values ($r \leqslant -0.15$) are easy, values around 0 ($-0.15 < r < 0.15$) are difficult and higher values ($r \geqslant 0.15$) are misleading. A scatter plot of fitness against distance is used when r is insufficient as a measure of the relationship between fitness and distance

Technique 8	**Static-ϕ metric** by Whitley et al. [86] with extensions [25,57]
Year	1995
Focus	GA deception (Section 3.6)
Search Independence	Medium: based on schemata, applicable to recombinative algorithms
Assumptions	Requires knowledge of global optima. Restricted to binary representations
Description	Given a binary fitness function with all hyperplane partitions, the static-ϕ metric calculates the degree of consistency between a ranking of schemata within hyperplane partitions based on average fitness values and a ranking based on the distance from the global optimum (using a form of match counting)
Result	A single value, from 0 to a positive value (could be normalised)

Technique 9	**Density of states** by Rosé et al. [62]
Year	1996
Focus	Fitness distribution (Section 3.3)
Search Independence	Complete: based on fitness function alone
Assumptions	None
Description	Given a sample of points (the Boltzmann ensemble method of sampling was used in the original study), the density of states quantifies the number of solutions with a given fitness value. The shape of the density of states graph (when plotted over a range of fitness values) can serve as a classifier of fitness functions. For example, maximisation problems with a fast decay of the density of states graph is indicative of the fast decay of the probability of finding a better solution, so should be harder to solve [62]
Result	A plot of the number of solutions per fitness value

Technique 10	**FEM and SEM**: First and second entropic measures by Vassilev et al. [76,78,77,79] with extensions [43]
Year	1997
Focus	Ruggedness and Smoothness with respect to neutrality (Sections 3.8 and 3.9)
Search Independence	High: based on a random walk through the fitness landscape
Assumptions	Assumes a discrete representation
Description	Based on a random walk, a sequence of three-point objects are generated. These objects are classified as rugged, smooth or neutral, based on the change in fitness values between neighbouring points. The ruggedness/ smoothness of the landscape is estimated using a measure of entropy with respect to the probability distribution of the rugged/non-rugged elements within the sequence
Result	A graph illustrating how ruggedness/smoothness changes with an increase in landscape neutrality. Ruggedness/ Smoothness values are in the range [0,1] where 1 indicates maximal ruggedness/smoothness

Technique 11	**Amplitude Spectra** by Hordijk and Stadler [27]
Year	1998
Focus	Ruggedness (Section 3.8)
Search Independence	High: based on any notion of neighbourhood
Assumptions	Assumes a discrete representation
Description	Using a form of Fourier analysis, the fitness landscape is decomposed into elementary landscapes. The resulting amplitude spectrum provides a summary of the properties of a landscape
Result	A graph of amplitude values for different interaction orders

Technique 12	**Bit-wise epistasis** by Fonlupt et al. [16]
Year	1998
Focus	Epistasis (Section 3.1)
Search Independence	Complete: based on fitness function alone
Assumptions	Assumes a binary representation
Description	For each bit position i calculate the variance of the fitness differences at that position by comparing the fitness values of the genotypes with 0 in bit position i and 1 in bit position i, with the other bit position values staying the same. The computation is based on a full enumeration of the search space if feasible. If not, bit-wise epistasis is approximated on a sample of schemata
Result	A plot of bit-wise epistasis values (in range [0,1]) for each bit position, where a value of 0 for all bit positions indicates no dependency between variables

Technique 13	**HDIL and HDBL** by Belaidouni and Hao [5]
Year	2000
Focus	Fitness distribution layout (Section 3.4).
Search Independence	High: uses Hamming distance as basis of measure
Assumptions	Assumes a binary representation
Description	Iso-cost levels are defined (sets of solutions with the same fitness values). The HDIL (Hamming Distance In a Level) measures the similarity of solutions within a given iso-cost level, based on the average Hamming distance between solutions in the set corresponding to that iso-level. The HDBL (Hamming Distance Between a Level) quantifies the distance between two iso-cost level sets C and C', based on the average Hamming distance required for solutions from C to reach any solution in set C'
Result	A single HDIL value for each iso-cost level, where a low value indicates that solutions with the same fitness value are clustered together in the search space and a high value that the solutions are spread out. A single HDBL value for each pair of iso-cost levels, where a low value indicates that on average a small Hamming distance has to be covered to move from a solution in one iso-cost level to a solution in the other iso-cost level

Technique 14	**Neutral walk** by Reidys and Stadler [60]
Year	2001
Focus	Neutrality (Section 3.9)
Search Independence	High: based on generic notion of neighbourhood
Assumptions	Assumes a discrete representation
Description	From a random starting position x_0 in the search space, perform a neutral walk as follows: generate all neutral neighbours of x_0. Find one neutral neighbour for which the total distance from the starting point will increase with the step. This process is continued until there are no neutral neighbours that result in the total distance increasing
Result	A single value (the number of steps in the neutral walk)

Technique 15	**Fitness evolvability portraits** by Smith et al. [64]
Year	2002
Focus	Evolvability (Section 3.11)
Search Independence	Medium: quantifies evolvability of a solution with reference to a particular operator (mutation in the original study)
Assumptions	Assumes a discrete representation
Description	For all solutions in a sample, calculate the evolvability metrics (such as the expected fitness of the top C_{th} percentile of offspring fitnesses). Determine the average metrics for solutions with the same (or similar) fitness values
Result	Plots of average evolvability metrics against fitness

Technique 16	**Fitness cloud** by Verel et al. [80] with extensions [72]
Year	2003
Focus	Evolvability (Section 3.11)
Search Independence	Medium: illustrates evolvability with reference to a particular search operator
Assumptions	Assumes the existence of a neighbourhood function
Description	For every solution x in the search space S of all possible solutions, determine a neighbour $x' \in S$ based on some search operator and plot the points $(f(x), f(x'))$ for every $x \in S$, where f is the fitness function
Result	Scatterplot showing the relationship between fitness values of parents and offspring

Technique 17	**Negative slope coefficient** by Vanneschi et al. [70,72] with extensions [74,55]
Year	2004
Focus	Evolvability (Section 3.11)
Search Independence	Medium: based on evolvability with reference to a particular search operator
Assumptions	Assumes the existence of a neighbourhood function
Description	Given a fitness cloud (Technique 16) partitioned into discrete bins, line segments are defined between the centroids of adjacent bins. The negative slope coefficient is the sum of all negative slopes between segments
Result	A single value (in the range $(-\infty, 0]$, where 0 indicates an easy problem and smaller values indicate more difficult problems)

Technique 18	**Information landscape hardness measure** by Borenstein and Poli [8,9] with extensions [11]
Year	2005
Focus	Deception in terms of difference from a landscape with perfect information for search (Section 3.6)
Search Independence	High: based on a comparison to an optimal landscape, which assumes the same neighbourhood structure
Assumptions	Requires knowledge of global optima. Assumes a discrete representation
Description	Given a discrete problem, compute the information landscape (matrix of probabilities of superiority of every solution with respect to every other solution). Determine an optimal information landscape, which presents perfect information to guide search. Calculate the distance between the optimal information landscape and the information landscape of the problem
Result	A single value in the range [0,1], where a value of 0 indicates no misleading information and 1 indicates maximal misleading information (difference from the optimal information landscape)

Technique 19	**Dispersion metric** by Lunacek and Whitley [42]
Year	2006
Focus	Global topology or presence of funnels (Section 3.7)
Search Independence	High: requires the calculation of distances in the solution space
Assumptions	Assumes the existence of a measure of distance between solutions
Description	Given a sample of points below a fitness threshold: if a decrease in threshold (assuming a minimisation problem) results in an increase in the dispersion of the points from the sample that are below the threshold, then this indicates the presence of multiple funnels in the landscape. Dispersion is calculated as the average pairwise distance in solution space between all points in a sample. The dispersion metric is calculated as the dispersion of a sample of points subtracted from the dispersion of a subset of the fittest points from the same sample
Result	A single value where smaller values (negative values) indicate a simpler global topology and larger values (positive values) indicate the presence of funnels. The magnitude of the dispersion metric is dependent on the scale of the distances in the search space

Technique 20	**Measures on neutral networks** by Vanneschi et al. [73] with extensions [75]
Year	2006
Focus	Neutrality (Section 3.9)
Search Independence	Medium: based on a notion of neighbourhood as defined by a search operator
Assumptions	Assumes a discrete representation
Description	Given a discrete fitness landscape, determine the set of all neutral networks (plateaus formulated as connected graphs of solutions with equal fitness neighbours). Measures are defined based on this set: average neutrality ratio, average fitness gain, non-improvable and "non-worsenable" solutions ratios
Result	Scatterplots of measures with respect to fitness values of neutral networks

Technique 21	**Fitness-probability cloud** by Lu et al. [41]
Year	2011
Focus	Evolvability (Section 3.11)
Search Independence	Medium: based on evolvability with reference to a particular search operator
Assumptions	Restricted to problems with a discrete representation since the technique is based on the notion of an escape rate [45], which assumes discrete steps through the search space
Description	Using a sample of n solution points (Metropolis–Hastings sampling used in the original study) with associated fitness values $f_1, \ldots, f_n$, generate a sample set of neighbours for each point through one application of a given search operator. Calculate the proportion P_i of neighbours with improved fitness for each f_i. The fitness-probability cloud is the set of (f_i, P_i) points
Result	A plot of (f_i, P_i) pairs, where f_i is a fitness value and P_i is the estimated escape probability of the sampled point i
Technique 22	**Accumulated escape probability** by Lu et al. [41]
Year	2011
Focus	Evolvability (Section 3.11)
Search Independence	Medium: based on evolvability with reference to a particular search operator
Assumptions	As for Technique 21
Description	Given a fitness-probability cloud as defined in Technique 21: $fpc = (f_1, P_1), \ldots, (f_n, P_n)$, the accumulated escape probability is defined as the mean of all P_i values in fpc
Result	A single value in the range [0,1], where a higher value indicates higher evolvability

4.1.3. Retrospective measures

There are some measures of problems that involve the actual execution of an optimisation algorithm. By attempting to solve the problem (possibly using a number of approaches or iterations of an algorithm), characteristics of the problem can be deduced in retrospect. Examples of retrospective measures include the following:

- Kauffman and Levin's adaptive walks [37]: This is a measure for estimating the ruggedness of a landscape and involves determining the lengths of hill-climbing walks.
- Ochoa's consensus sequence plots [51], which involves running a GA on the problem multiple times with a decreasing mutation rate.
- Garnier and Kallel's [19] method for estimating the number and distribution of local optima, which involves performing a steepest ascent search from a random sample of starting positions.

The measures considered in more detail in this study are all computed *a priori* and although some measures are based on theories applicable to specific algorithms or on particular search operators, the purpose is to obtain information on the problem without actually executing a particular search algorithm.

4.2. Introduction to the survey

The aim of this survey was to obtain a better understanding of existing techniques for characterising optimisation problems. For a survey of techniques to be useful, distinguishing characteristics needed to be highlighted, but it was not clear what these distinguishing characteristics should be. Naudts and Kallel [48] distinguish between exact and approximate measures. A measure is *exact* if it is computed using all solutions in the search space, whereas an approximate measure is computed using a sample of the search space. He et al. [24] further distinguish between predictive and non-predictive measures. They define a predictive difficulty measure as one where the algorithm's worst-case running time is bounded by a polynomial in n (the problem size). Exact computation of many of the difficulty measures is in general exponential with respect to the problem size [30], so although many of the techniques were originally defined as exact measures, they are in practice used as approximate measures. Since the aim was not to divide techniques into classes, but rather to understand techniques to be of practical use, more descriptive distinguishing features are highlighted. These are described and motivated below and correspond to the attributes used in Table 1.

1. *Technique (with unique number)*: The first attribute gives the name of the technique and the reference to the authors that proposed the technique. The techniques in the table appear in chronological order by the year of the first reference to the technique. The reason for organising the survey in this way was to facilitate an understanding of how the techniques have evolved over the last two decades. Where a technique was adapted in different ways by subsequent studies by the same or different authors, citations to significant complementary research on the original technique are listed as "extensions".
2. *Year*: The year that the technique was first introduced in published form.
3. *Focus*: The overall focus of the technique is given as the second attribute. This refers to what is measured or predicted by the technique. To explicitly tie each technique to the features discussed in Section 3, the relevant subsection discussing the feature is stated in parentheses.
4. *Search independence*: This descriptor refers to the level with which a technique is bound to a particular search algorithm. Four categories are used:
   (a) Complete: A technique for characterising a problem is regarded as having complete search independence when the technique is based on a fitness function alone and not on any notion of neighbourhood/nearness between solutions. In other words, there is no fitness landscape involved in the technique.

(b) High: A technique is regarded as having high search independence when it is based on some generic or neutral notion of neighbourhood/distance between solutions, such as Hamming distance or Euclidean distance, which defines the fitness landscape. An example of a technique with high search independence is one which is based on a random walk through the landscape, without any significant biased direction.

(c) Medium: A technique is regarded as having medium search independence when the sampling or analysis is based on, and therefore biased by, some theory or notion particular to a given search algorithm.

(d) Low: A technique is regarded as having low search independence if it is based on a sample generated by the actual execution of an optimisation algorithm. The survey does not include any techniques with low search independence, as these would be classified as retrospective measures (Section 4.1.3).

5. *Assumptions*: Where there are significant assumptions on which the technique is based, these are mentioned.
6. *Brief summary*: A brief summary of how the technique works is provided.
7. *Result*: There are many different forms of output produced by the techniques outlined in this survey. For example, some result in a single numerical output value, while others produce visual output in the form of scatterplots, graphs or charts. The *Result* attribute describes the output produced by the technique.

5. Discussion

The aim of this study was to make sense of the body of work outlined in Table 1 in order to better utilise these techniques in practical ways. This section highlights what the survey reveals: where the focus has been, where the gaps are and possible ways in which techniques can be adapted to be more usable or relevant. The main points of the discussion in this section are summarised as possible ways forward in Table 2.

5.1. The focus of techniques

Scanning the Focus attribute in Table 1 reveals how the techniques for characterising problems have evolved over time. Studies starting in the late 1980s through to the 90s had a strong focus on ruggedness, with some focus on other themes including deception, epistasis and fitness variance/distribution. The late 90s saw the emergence of neutrality as one of the new focus areas, with a number of studies highlighting the fact that there were problems that were not rugged and yet were hard to solve and many of these had high neutrality. The 2000s see evolvability emerge as a new focus of many techniques, with other themes including global topology and fitness statistics.

Table 2
Summary of some possible ways forward.

Way forward 1	New techniques for features not covered by existing techniques
Description	There are features that are known to influence problem difficulty, but for which there are no known predictive measures that can be used to obtain *a priori* information on the problem
Examples	Symmetry (Section 3.10) and variable interdependency for continuous functions (Section 3.1)
Way forward 2	Shifting focus from optimality to searchability (or evolvability)
Description	Techniques which measure deception assume knowledge of the global optima, which is not known for unseen problems. These techniques can be converted to instead measure searchability (or evolvability), by basing the analysis or calculation on the fittest solution from a sample, instead of the global optimum
Examples	This approach could apply to GA-deception (Technique 1), fitness distance correlation (Technique 7), static-ϕ metric (Technique 8), or information landscape hardness measure (Technique 18)
Way forward 3	Generalising the notion of neighbourhood
Description	Techniques with medium search independence can be adapted to work with more general notions of neighbourhood and in this way be adapted to techniques with high search independence
Examples	Fitness cloud (Technique 16) and associated negative slope coefficient (Technique 17) could be adapted to work with a generic distance measure as a neighbourhood function
Way forward 4	Specialising the notion of neighbourhood
Description	Techniques with high search independence can be adapted to have medium independence by working with more specific notions of neighbourhood for a given search algorithm
Example	Correlation length (Technique 3), based on random walks, can be adapted to measure ruggedness of a search path of a particular search algorithm
Way forward 5	Adapting techniques for different representations
Description	Techniques that are defined for one representation (e.g. discrete) can possibly be adapted to be used for problems with a different representation (e.g. continuous representation)
Example	Information landscape hardness measure (Technique 18) defined for discrete problems could possibly be adapted for continuous problems based on a random sample of solutions
Way forward 6	Scalarizing visual outputs
Description	Techniques that produce plots or graphs can form the basis for new numerical measures to facilitate automated analysis
Examples	Density of states (Technique 9), which results in a visual plot, could be condensed into a single measure that in some way captures the shape of the graph

The many different factors on which the techniques focus highlight the wide range of features that can influence problem difficulty. Each factor is clearly important to some degree and it opens the question: Are there characteristics which are also important, but are missing from the list of available techniques? For example, symmetry is known to influence problem difficulty [63,85,49,13], but to the authors' knowledge there are no known techniques for measuring symmetry in fitness landscapes. Another example is the degree of variable interdependency. Although there are techniques for measuring epistasis that appear in the survey, these all only apply to discrete representations. These and other potential factors point to possible areas for future work.

There are four techniques in Table 1 that focus on measuring deception, Goldberg's GA-deception (Technique 1), Jones and Forrest's fitness distance correlation (Technique 7), Whitley et al.'s static-ϕ metric (Technique 8), and Borenstein and Poli's information landscape hardness measure (Technique 18). These four techniques are also the only ones listed that require knowledge of the global optima. This is because it only makes sense to talk of deception in reference to finding the optimal solution(s). Characterising a problem based on deception is not useful in practice for two reasons:

- If the aim is to obtain *a priori* information on the problem, the global optima will not be known.
- In many cases it may be infeasible to expect an algorithm to find a global optimum and if a problem guides an algorithm to a reasonable solution, then this may be sufficient.

Assuming a broader notion of success and failure than finding an optimal solution, an alternative aim could be to measure the ease or difficulty with which a search process will progress towards a place of better fitness. This is equivalent to a shift in focus from optimality to searchability (or evolvability as it is more commonly known). If this position is taken, it becomes possible to use techniques such as fitness distance correlation (Technique 7) even when the global optimum is not known. The fitness distance correlation measure was based on the premise that *if the fitness function correlates well with the distance to the optimum, then search will be easier* (assuming a minimisation problem). If the measure is changed to focus on searchability, rather than deception or problem difficulty, and the premise is re-stated as: *if the fitness function correlates well with the distance to a position of higher fitness, then search will progress more easily*, then the technique can be used with the most fit value from a sample in place of the optimum. For example, given an unknown problem, a random sample of solutions can be generated and the fitness values determined. From this sample, the most fit solution is determined and used as the basis for the fitness distance correlation calculation of Technique 7. The result would no longer be a measure of deception for local search, but would instead be a measure of how easy or hard it would be for a local search algorithm to progress to a place of better fitness. In this way, a technique for measuring deception or problem difficulty is converted into a technique for measuring problem searchability.

5.2. Search independence

Each technique in Table 1 is characterised as having complete, high, or medium search independence. Depending on the purpose and context, different kinds of techniques will be more suitable. On the one hand, where the choice of algorithm is set, an appropriate technique with medium search independence could be used to better understand the given problem with reference to that algorithm. For example, assuming an evolutionary algorithm is being used, the accumulated escape probability (Technique 22) could be used to guide the choice of appropriate parameters for the algorithm on the given problem. On the other hand, where the purpose is to choose an appropriate algorithm for a given problem, techniques for characterising the problem with complete or high search independence will be more useful.

In some cases a technique that is based on a particular search operator and therefore regarded as having medium search independence could be adapted to use a generic notion of neighbourhood, so that the analysis could apply to different algorithms. For example, the negative slope coefficient (Technique 17) is described as having medium search independence because the neighbourhood is defined in terms of a particular search operator (subtree mutation for genetic programming in the original study). If instead, the neighbourhood was defined using some generic notion of distance, such as Euclidean distance for a continuous problem, then the technique will be used with high search independence. Conversely, a technique with high search independence can be adapted into a technique with medium search independence. For example, the correlation length technique (Technique 3) originally based on random walks, could be instead based on the trajectory of a particle within a PSO swarm. In this way, the technique would be used to measure ruggedness from the particular viewpoint of a PSO search process.

5.3. Further proposed work

Many of the techniques outlined in the survey assume the fitness function (f) is a mapping from the binary space to real space ($f : \{0, 1\}^n \rightarrow \mathbb{R}$), or from some discrete alphabet to real space. In some cases this is a restriction, because there is no obvious way of using the technique for other representations, such as continuous representations (where $f : \mathbb{R}^n \rightarrow \mathbb{R}$). In other cases, although the technique is described in terms of one representation, this is not necessarily a restriction. For example, the information landscape hardness measure (Technique 18) is defined for discrete representations and involves constructing a matrix of fitness superiority values of all solutions with respect to all other solutions. This approach could be

adapted to a continuous representation by using a random sample of solutions. Without knowledge of the global optimum the technique would also have to be adapted in the way described in Way forward 2 of Table 2.

The Result attribute of Table 1 also presents opportunities for further work. Some of the techniques produce plots or graphs as results. While visual output is useful for human analysis, numerical output is more useful for facilitating automated analysis. There are examples of existing numerical measures that are based on other techniques that produce visual output, such as:

- Vanneschi et al.'s negative slope coefficient (Technique 17), which is a numerical output measure based on Verel et al.'s fitness cloud scatterplot (Technique 16).
- Malan and Engelbrecht's [43] single ruggedness measure, which is based on the first entropic measure output graph by Vassilev et al. (Technique 10).

In similar ways, other measures with non-numerical output could form the basis of new numerical measures. For example, the result of the density of states technique (Technique 9) is a plot of the number of solutions against fitness. The tail end of the graph closer to optimal fitness values is the more significant part in terms of assessing the difficulty for search. A possible single measure of fitness distribution could in some way quantify the proportion of solutions at better fitness values in contrast to the number of solutions at other less fit fitness values.

Other than developing new techniques or adapting existing techniques for fitness landscape analysis, there is also further research required in analysing the link between problem characteristics and algorithm performance. Early studies in this direction include: using the negative slope coefficient to choose the most appropriate genetic programming configuration to solve real life applications [71]; using fitness distance correlation and correlation length to analyse heuristic search spaces [52]; the use of correlation length and fitness distance correlation to design memetic algorithms [46]; and analysing the correlation between problem difficulty measures and hybrid evolutionary algorithms [54]. Further work is required in analysing a wider range of measures and in finding effective ways to guide the problem solving process based on these measured problem characteristics. History has shown the inadequacy of trying to predict algorithm performance based on a single characteristic. What seems to be needed is a combined approach, where a number of different characteristics are analysed together to determine the suitability of a given search algorithm to a given problem. This in itself is a complex multi-dimensional problem where there is interdependency between the variables (in this case the characteristics of a problem). Given a suitable dataset of known benchmark problems with generated numerical characteristics, and a subset of algorithms, the actual performance of the algorithms on the set of problems can be determined by solving the problems using the algorithms. Some form of data mining could then be used to determine if a mapping can be found between problem characteristics and algorithm performance. If a mapping can be found and it is sufficiently general, the mapping could be used to predict algorithm performance on unseen problems. This is a promising area of future research.

6. Conclusion

This paper provides a survey of existing techniques for characterising problems. Each technique is described in terms of the focus (what is measured), the level of search independence, assumptions on which the technique is based, and the result produced. The survey reveals how the focus has changed over the last two decades. Some characteristics, such as ruggedness, are the focus of many different techniques, but others, such as symmetry, are not well represented. Suggestions are made for ways in which existing techniques can be adapted to be more usable or relevant. Techniques that require knowledge of the global optima can be used without this knowledge by shifting the focus from optimality to searchability. The same fitness analysis technique can also be used in multiple ways by changing the search independence and in effect analysing different fitness landscapes for the same problem.

Further work is needed in developing new ways of characterising problems, adapting existing techniques, and investigating the links between problem characteristics and algorithm performance. It is hoped that this paper will invoke renewed interest in the field of understanding complex optimisation problems and ultimately lead to better decision making on the use of algorithms.

References

[1] L. Altenberg, The evolution of evolvability in genetic programming, in: K. Kinnear (Ed.), Advances in Genetic Programming, MIT Press, Cambridge, 1994, pp. 47–74.
[2] L. Altenberg, Fitness distance correlation analysis: an instructive counterexample, in: Proceedings of the Seventh International Conference on Genetic Algorithms, Morgan Kaufmann, 1997, pp. 57–64.
[3] W. Beaudoin, S. Verel, P. Collard, C. Escazut, Deceptiveness and neutrality: the ND family of fitness landscapes, in: GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2006, pp. 507–514.
[4] M. Bélaidouni, J.K. Hao, An analysis of the configuration space of the maximal constraint satisfaction problem, in: Parallel Problem Solving from Nature – PPSN VI, Lecture Notes in Computer Science, vol. 1917/2000, Springer-Verlag, 2000, pp. 49–58.
[5] M. Bélaidouni, J.K. Hao, Landscapes of the maximal constraint satisfaction problem, in: Artificial Evolution, Lecture Notes in Computer Science, vol. 1929/2000, Springer-Verlag, 2000, pp. 244–255.
[6] H.G. Beyer, Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice, Comput. Methods Appl. Mech. Eng. 186 (2000) 239–267.

 [7] Y. Borenstein, R. Poli, Fitness distributions and GA hardness, in: Parallel Problem Solving from Nature – PPSN VIII, Springer, Berlin/Heidelberg, 2004, pp. 11–20.
 [8] Y. Borenstein, R. Poli, Information landscapes, in: GECCO '05: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, ACM Press, New York, NY, USA, 2005, pp. 1515–1522.
 [9] Y. Borenstein, R. Poli, Information landscapes and problem hardness, in: GECCO '05: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, ACM Press, New York, NY, USA, 2005, pp. 1425–1431.
[10] Y. Borenstein, R. Poli, Kolmogorov complexity, optimization and hardness, in: IEEE Congress on Evolutionary Computation, 2006, pp. 112–119.
[11] Y. Borenstein, R. Poli, Decomposition of fitness functions in random heuristic search, in: FOGA'07: Proceedings of the 9th International Conference on Foundations of Genetic Algorithms, Lecture Notes in Computer Science, vol. 4436/2007, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 123–137.
[12] J.C. Culberson, On the Futility of Blind Search, Technical Report, Department of Computing Science, University of Alberta, Edmonton, Canada, 1996.
[13] A. Czarn, C. MacNish, K. Vijayan, B. Turlach, The detrimentality of crossover, in: AI'07: Proceedings of the 20th Australian Joint Conference on Advances in Artificial Intelligence, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 632–636.
[14] Y. Davidor, Epistasis variance: a viewpoint on GA-hardness, Found. Genetic Algorithms (1991) 23–35.
[15] K. Deb, D.E. Goldberg, Sufficient conditions for deceptive and easy binary functions, Ann. Math. Artif. Intell. 10 (1994) 385–408.
[16] C. Fonlupt, D. Robilliard, P. Preux, A bit-wise epistasis measure for binary search spaces, in: PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature, Springer-Verlag, London, UK, 1998, pp. 47–56.
[17] W. Fontana, P.F. Stadler, E.G. Bornberg-Bauer, T. Griesmacher, I.L. Hofacker, M. Tacker, P. Tarazona, E.D. Weinberger, P. Schuster, RNA folding and combinatory landscapes, Phys. Rev. E 47 (1993) 2083–2099.
[18] S. Forrest, M. Mitchell, What makes a problem hard for a genetic algorithm? Some anomalous results and their explanation, Mach. Learn. 13 (1993) 285–319.
[19] J. Garnier, L. Kallel, How to detect all maxima of a function, in: Theoretical Aspects of Evolutionary Computing, Springer-Verlag, London, UK, 2001, pp. 343–370.
[20] D.E. Goldberg, Simple genetic algorithms and the minimal deceptive problem, in: L. Davis (Ed.), Genetic Algorithms and Simulated Annealing, Pitman, London, 1987, pp. 74–88.
[21] D.E. Goldberg, Genetic Algorithms, Genetic algorithms and Walsh functions: part II, deception and its analysis, Complex Sys. 3 (1989) 153–171.
[22] P.D. Grünwald, P.M.B. Vitányi, Algorithmic information theory, in: P. Adriaans, J. van Benthem (Eds.), Philosophy of Information, Elsevier, 2008, pp. 281–317.
[23] H. Guo, W.H. Hsu, GA-hardness revisited, in: Genetic and Evolutionary Computation – GECCO-2003, Lecture Notes in Computer Science, vol. 2724/2003, Springer-Verlag, Berlin, 2003, pp. 1584–1585.
[24] J. He, C. Reeves, C. Witt, X. Yao, Note on problem difficulty measures in black-box optimization: classification, realizations and predictability, Evol. Comput. 15 (2007) 435–443.
[25] R.B. Heckendorn, D. Whitley, S. Rana, Nonlinearity, hyperplane ranking and the simple genetic algorithm, in: Foundations of Genetic Algorithms (FOGA) 4, Morgan Kaufmann, 1997, pp. 181–202.
[26] W. Hordijk, A measure of landscapes, Evol. Comput. 4 (1996) 335–360.
[27] W. Hordijk, P.F. Stadler, Amplitude spectra of fitness landscapes, Adv. Complex Syst. 1 (1998) 39–66.
[28] J. Horn, D.E. Goldberg, Genetic algorithm difficulty and the modality of fitness landscapes, in: L.D. Whitley, M.D. Vose (Eds.), Foundations of Genetic Algorithms 3, Morgan Kaufmann, San Francisco, CA, 1995, pp. 243–269.
[29] E. Izquierdo-Torres, Evolving Dynamical Systems: Nearly Neutral Regions in Continuous Fitness Landscapes, Master's thesis, University of Sussex, 2004.
[30] T. Jansen, On classifications of fitness functions, in: Theoretical aspects of evolutionary computing, Springer-Verlag, London, UK, 2001, pp. 371–385.
[31] M. Jelasity, B. Tóth, T. Vinkó, Characterizations of trajectory structure of fitness landscapes based on pairwise transition probabilities of solutions, in: Proceedings of the 1999 Congress on Evolutionary Computation (CEC99), IEEE, IEEE Press, 1999, pp. 623–630.
[32] T. Jones, Evolutionary Algorithms, Fitness Landscapes and Search, Phd thesis, The University of New Mexico, 1995.
[33] T. Jones, S. Forrest, Fitness distance correlation as a measure of problem difficulty for genetic algorithms, in: Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kaufmann, 1995, pp. 184–192.
[34] L. Kallel, B. Naudts, C.R. Reeves, Properties of fitness functions and search landscapes, in: L. Kallel, B. Naudts, A. Rogers (Eds.), Theoretical Aspects of Evolutionary Computing, Springer, Berlin, 2001, pp. 175–206.
[35] S.A. Kauffman, Adaptation on rugged fitness landscapes, in: E. Stein (Ed.), Lectures in the Sciences of Complexity, Addison-Wesley, Reading, 1989, pp. 527–618.
[36] S.A. Kauffman, The Origins of Order: Self-Organization and Selection in Evolution, Oxford University Press, USA, 1993.
[37] S.A. Kauffman, S. Levin, Towards a general theory of adaptive walks on rugged landscapes, J. Theor. Biol. 128 (1987) 11–45.
[38] K.E. Kinnear, Fitness landscapes and difficulty in genetic programming, in: Proceedings of the First IEEE Conference on Evolutionary Computing, IEEE Press, 1994, pp. 142–147.
[39] B. Levitan, S. Kauffman, Adaptive walks with noisy fitness measurements, Mol. Divers. 1 (1995) 53–68.
[40] M. Lipsitch, Adaptation on rugged landscapes generated by iterated local interactions of neighboring genes, in: R.K. Belew, L.B. Booker (Eds.), Proceedings of the 4th International Conference on Genetic Algorithms, Morgan Kaufmann, San Diego, CA, USA, 1991, pp. 128–135.
[41] G. Lu, J. Li, X. Yao, Fitness-probability cloud and a measure of problem hardness for evolutionary algorithms, in: Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science, vol. 6622/2011, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 108–117.
[42] M. Lunacek, D. Whitley, The dispersion metric and the CMA evolution strategy, in: GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2006, pp. 477–484.
[43] K.M. Malan, A.P. Engelbrecht, Quantifying Ruggedness of Continuous Landscapes using Entropy, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2009, pp. 1440–1447.
[44] B. Manderick, M.K. de Weger, P. Spiessens, The genetic algorithm and the structure of the fitness landscape, in: R.K. Belew, L.B. Booker (Eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, 1991, pp. 143–150.
[45] P. Merz, Advanced fitness landscape analysis and the performance of Memetic algorithms, Evol. Comput. 12 (2004) 303–325.
[46] P. Merz, B. Freisleben, Fitness landscape analysis and Memetic algorithms for the quadratic assignment problem, Trans. Evol. Comput. 4 (2000) 337–352.
[47] B. Naudts, L. Kallel, Some Facts About So Called GA-Hardness Measures, Technical Report 379, CMAP, Ecole Polytechnique, France, 1998.
[48] B. Naudts, L. Kallel, A comparison of predictive measures of problem difficulty in evolutionary algorithms, Trans. Evol. Comput. 4 (2000) 1.
[49] B. Naudts, J. Naudts, The effect of spin-flip symmetry on the performance of the Simple GA, in: PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature, Springer-Verlag, London, UK, 1998, pp. 67–76.
[50] B. Naudts, D. Suys, A. Verschoren, Epistasis as a basic concept in formal landscape analysis, in: Proceedings of the 7th International Conference on Genetic Algorithms, Morgan Kaufmann, 1997, pp. 65–72.
[51] G. Ochoa, Consensus sequence plots and error thresholds: tools for visualising the structure of fitness landscapes, in: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, PPSN VI, Springer-Verlag, London, UK, 2000, pp. 129–138.
[52] G. Ochoa, R. Qu, E.K. Burke, Analyzing the landscape of a graph based hyper-heuristic for timetabling problems, in: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, GECCO '09, ACM, New York, NY, USA, 2009, pp. 341–348.
[53] G. Ochoa, M. Tomassini, S. Vérel, C. Darabos, A Study of NK Landscapes' Basins and Local Optima Networks, in: Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2008, pp. 555–562.

[54] M. Pelikan, NK landscapes, problem difficulty, and hybrid evolutionary algorithms, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO '10, ACM, New York, NY, USA, 2010, pp. 665–672.

[55] R. Poli, L. Vanneschi, Fitness-proportional negative slope coefficient as a hardness measure for genetic algorithms, in: GECCO '07: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2007, pp. 1335–1342.

[56] N. Radcliffe, P. Surry, Fitness variance of formae and performance prediction, in: L. Whitley, M. Vose (Eds.), Foundations of Genetic Algorithms III, Morgan Kaufmann, 1994, pp. 51–72.

[57] S. Rana, Examining the role of local optima and schema processing in genetic search, Ph.D. thesis, Colorado State University, USA, 1999. Adviser: Whitley, Darrell.

[58] S. Rana, L.D. Whitley, R. Cogswell, Searching in the presence of noise, in: PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, Springer-Verlag, 1996, pp. 198–207.

[59] C.R. Reeves, C.C. Wright, Epistasis in genetic algorithms: an experimental design perspective, in: Proc. of the 6th International Conference on Genetic Algorithms, Morgan Kaufmann, 1995, pp. 217–224.

[60] C.M. Reidys, P.F. Stadler, Neutrality in fitness landscapes, Appl. Math. Comput. 117 (2001) 321–350.

[61] T.P. Riopka, Adapting to complexity during search in combinatorial landscapes, in: EvoWorkshops'03: Proceedings of the 2003 International Conference on Applications of Evolutionary Computing, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 311–321.

[62] H. Rosé, W. Ebeling, T. Asselmeyer, The density of states – a measure of the difficulty of optimisation problems, in: PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, Springer-Verlag, London, UK, 1996, pp. 208–217.

[63] R. Salomon, Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms, BioSystems 39 (1996) 263–278.

[64] T. Smith, P. Husbands, P. Layzell, M. O'Shea, Fitness landscapes and evolvability, Evol. Comput. 10 (2002) 1–34.

[65] P.F. Stadler, Towards a theory of landscapes, in: R. Lopéz-Peña, R. Capovilla, R. Garcıa-Pelayo, H. Waelbroeck, F. Zertuche (Eds.), Complex Systems and Binary Networks, vol. 461, Springer Verlag, Berlin, New York, 1995, pp. 77–163.

[66] P.F. Stadler, Fitness landscapes, in: A.V. Michael Lässig (Ed.), Biological Evolution and Statistical Physics, Lecture Notes in Physics, vol. 585, Springer-Verlag, Berlin, Heidelberg, 2002, pp. 183–204.

[67] A.M. Sutton, D. Whitley, M. Lunacek, A. Howe, PSO and multi-funnel landscapes: how cooperation might limit exploration, in: GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM, New York, NY, USA, 2006, pp. 75–82.

[68] P.D. Turney, Increasing Evolvability Considered as a Large-Scale Trend in Evolution, in: Proceedings of 1999 Genetic and Evolutionary Computation Conference Workshop Program (GECCO-99 Workshop on Evolvability), pp. 43–46.

[69] C. Van Hoyweghen, B. Naudts, Symmetry in the search space, in: Proceedings of the 2000 Congress on Evolutionary Computation, 2000, volume 2, pp. 1072 – 1078.

[70] L. Vanneschi, Theory and Practice for Efficient Genetic Programming, Phd thesis, Faculty of Sciences, University of Lausanne, 2004.

[71] L. Vanneschi, Investigating problem hardness of real life applications, in: R.L. Riolo, T. Soule, B. Worzel (Eds.), Genetic Programming Theory and Practice V, Genetic and Evolutionary Computation, Springer, Ann Arbor, 2007, pp. 107–125.

[72] L. Vanneschi, M. Clergue, P. Collard, M. Tomassini, S. Verel, Fitness Clouds and Problem Hardness in Genetic Programming, in: GECCO (2), 2004, pp. 690–701.

[73] L. Vanneschi, Y. Pirola, P. Collard, A quantitative study of neutrality in GP boolean landscapes, in: GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2006, pp. 895–902.

[74] L. Vanneschi, M. Tomassini, P. Collard, S. Verel, Negative slope coefficient: a measure to characterize genetic programming fitness landscapes, in: Genetic Programming, Lecture Notes in Computer Science, vol. 3905/2006, Springer, Berlin/ Heidelberg, 2006, pp. 178–189.

[75] L. Vanneschi, M. Tomassini, P. Collard, S. Verel, Y. Pirola, G. Mauri, A comprehensive view of fitness landscapes with neutrality and fitness clouds, in: Genetic Programming, Lecture Notes in Computer Science, vol. 4445/2007, Springer, Berlin/ Heidelberg, 2007, pp. 241–250.

[76] V.K. Vassilev, An Information Measure of Landscapes, in: Proceedings of the Seventh International Conference on Genetic Algorithms, 1997, pp. 49–56.

[77] V.K. Vassilev, Fitness Landscapes and Search in the Evolutionary Design of Digital Circuits, Phd thesis, Napier University, 2000.

[78] V.K. Vassilev, T.C. Fogarty, J.F. Miller, Information characteristics and the structure of landscapes, Evol. Comput. 8 (2000) 31–60.

[79] V.K. Vassilev, T.C. Fogarty, J.F. Miller, Smoothness, ruggedness and neutrality of fitness landscapes: from theory to application, in: Advances in Evolutionary Computing: Theory and applications, Springer-Verlag New York, Inc., 2003, pp. 3–44.

[80] S. Verel, P. Collard, M. Clergue, Where are bottlenecks in NK fitness landscapes? in: The 2003 Congress on Evolutionary Computation, CEC '03, vol. 1, 2003, pp. 273–280.

[81] S. Vérel, G. Ochoa, M. Tomassini, Local optima networks of NK landscapes with neutrality, IEEE Trans. Evol. Comput. 15 (2011) 783–797.

[82] H. Waeselynck, P. Thévenod-Fosse, O. Abdellatif-Kaddour, Simulated annealing applied to test generation: landscape characterization and stopping criteria, Empir. Softw. Eng. 12 (2007) 35–63.

[83] I. Wegener, Complexity Theory – Exploring the Limits of Efficient Algorithms, Springer, Berlin, New York, 2005.

[84] E. Weinberger, Correlated and uncorrelated fitness landscapes and how to tell the difference, Biol. Cybern. 63 (1990) 325–336.

[85] D. Whitley, S. Rana, J. Dzubera, K.E. Mathias, Evaluating evolutionary algorithms, Artif. Intell. 85 (1996) 245–276.

[86] L.D. Whitley, K.E. Mathias, L.D. Pyeatt, Hyperplane ranking in simple genetic algorithms, in: Proceedings of the 6th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 231–238.

[87] D.H. Wolpert, W.G. Macready, No Free Lunch Theorems for Search, Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995.

[88] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, Trans. Evol. Comput. 1 (1997) 67–82.

[89] S. Wright, The roles of mutation, inbreeding, crossbreeding, and selection in evolution, in: Proceedings of the Sixth International Congress on Genetics, 1932, pp. 356–366.

[90] S. Wright, Surfaces of selective value revisited, Am. Nat. 131 (1988) 115–123.

[91] B. Xin, J. Chen, F. Pan, Problem difficulty analysis for particle swarm optimization: deception and modality, in: GEC '09: Proceedings of the first ACM/ SIGEVO Summit on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2009, pp. 623–630.