Noname manuscript No.
(will be inserted by the editor)

Characterising the searchability of continuous optimisation problems for PSO

K. M. Malan · A. P. Engelbrecht

Received: date / Accepted: date

Abstract The focus of research in swarm intelligence has been largely on the algorithmic side with relatively little attention being paid to the study of problems and the behaviour of algorithms in relation to problems. When a new algorithm or variation on an existing algorithm is proposed in the literature, there is seldom any discussion or analysis of algorithm weaknesses and on what kinds of problems the algorithm is expected to fail. Fitness landscape analysis is an approach that can be used to analyse optimisation problems. By characterising problems in terms of fitness landscape features, the link between problem types and algorithm performance can be studied. This article investigates a number of measures for analysing the ability of a search process to improve fitness on a particular problem (called *evolvability* in literature, but referred to as *searchability* in this study to broaden the scope to non-evolutionary based search techniques). A number of existing fitness landscape analysis techniques originally proposed for discrete problems are adapted to work in continuous search spaces. For a range of benchmark problems, the proposed searchability measures are viewed alongside performance measures for a traditional global best particle swarm optimisation (PSO) algorithm. Empirical results show that no single measure can be used as a predictor of PSO performance, but that multiple measures of different fitness landscape features can be used together to predict PSO failure.

Keywords Fitness landscape analysis · evolvability · particle swarm optimisation

1 Introduction

PSO algorithms have become popular for solving complex real-valued optimisation problems and have been applied successfully to a wide range of problems. Just like

K. M. Malan
Department of Computer Science, University of Pretoria, Pretoria, South Africa
Tel.: +2712-420-3618
Fax: +2712-362-5188
E-mail: kmalan@cs.up.ac.za

A. P. Engelbrecht
E-mail: engel@cs.up.ac.za

all other metaheuristics, however, PSO algorithms sometimes fail. The traditional PSO algorithm can easily be understood in terms of the basic elements of a swarm of solution vectors, velocities and global and personal guides, but due to stochastic elements and the dynamic swarm interactions, the resulting behaviour is not as easy to understand or predict. Given an unknown problem, a practitioner or researcher having to choose an appropriate PSO algorithm has almost no guidance with respect to algorithm configuration and parameter setting and the most common approach is to use trial-and-error to find a solution that hopefully works well.

Smith-Miles (2008) proposed applying the framework of the general algorithm selection problem (Rice, 1976) to optimisation. The basic idea is to use machine learning to train an algorithm performance predictor based on features of the problem. Contrary to the work of Smith-Miles, the goal of the current study is not to develop an algorithm selector, but to investigate whether searchability measures can be used to predict PSO performance. Such measures could then form part of the solution to solving the algorithm selection problem for PSO variants.

There are many problem features that have been shown to influence the difficulty of a problem for search (Malan and Engelbrecht, 2013c), such as the ruggedness of the fitness landscape, interdependency between variables, presence of multiple funnels in a fitness landscape, etc. In particular, Langdon and Poli (2007) showed how the features of a problem influence PSO search. Using genetic programming (GP), they evolved two-dimensional objective functions to maximise the difference between the performance of PSO, differential evolution and a covariance matrix adaptation evolution strategy (CMA-ES), as well as PSO with different parameter settings. They found, for example, that PSO struggled to maximize a fitness landscape with a ramp and "cliff edge" at the end of the ramp. They showed that GP was always able to find a landscape that was more suited to one algorithm or setting than another, confirming the link betweeen problem features and algorithm performance.

There are studies that have been successful in predicting performance or selecting algorithms for continuous optimisation based on problem features. For example, Bischl et al (2012) used low-level problem features based on exploratory landscape analysis (proposed by Mersmann et al (2011)) of the BBOB (2013) test suite and then successfully used one-sided support vector regression to predict which algorithms (from a portfolio of algorithms) would perform well. Muñoz et al (2012) developed a model using a neural network to predict performance of a CMA-ES algorithm. The inputs to the model were a number of landscape features as well as algorithm parameters. In this way the model could be used to select the best predicted algorithm configuration of the problem. These studies show that problem characteristics can be used to predict or select the most appropriate algorithm from a set of possible algorithms. However, rather than analysing individual features a predictors, data mining is used to predict performance based on combinations of different features.

Given an unknown problem, the aim of the current study is to approximate problem features based on samples from the search space to form a characterisation vector of the problem. In previous work (Malan and Engelbrecht, 2013a) measures were implemented for quantifying ruggedness, funnels and gradients and the link to PSO performance was investigated. This paper extends that work by investigating ways of measuring searchability. Empirical results show that although there is evidence of some correlation of the proposed searchability measures to PSO performance, no single measure is a good predictor of hardness for PSO. However, when the proposed searchability measures are considered with other fitness landscape measures, fairly accurate failure prediction models can be deduced for different PSO algorithms.

A further consideration regarding the related studies discussed above are that the prediction models are black boxes that do not help in understanding the link between problems and algorithms. A distinguishing feature of the current study is that the aim is not only to predict PSO algorithm failure or performance, but also to understand PSO better. By using decision tree induction, the kinds of problems that PSO struggles with are highlighted and the resulting models therefore provide insights into the algorithms themselves. The proposed measures are based on fitness landscape concepts, such as ruggedness and searchability (rather than low-level statistical features), which makes it possible to reason about the problem features in fitness landscape terms in relation to algorithm performance.

The article is organised as follows: Section 2 describes the notion of evolvability, defines searchability, and describes existing techniques for quantifying or visualising searchability. For each existing technique, modifications are proposed to be applicable to PSO. The proposed techniques are tested on one-dimensional benchmarks in Section 3, while Section 4 investigates the link between the proposed searchability measures and actual PSO performance on higher dimensional problems. Finally, Section 5 shows how the proposed searchability measures can be combined with other fitness landscape measures to predict PSO failure.

2 Evolvability / Searchability

Evolvability can be loosely defined as the capacity to evolve (Turney, 1999). Altenberg (1994) describes evolvability with particular reference to genetic algorithms as the ability of a population to produce offspring that are fitter than their parents. Evolvability is defined in this study as the ability of a given search process to move to a place in the landscape of better objective function value and is referred to as *searchability* to broaden the scope of evolvability beyond evolutionary based algorithms.

Some problem analysis measures were originally conceived as a way of quantifying problem difficulty, but can be viewed as measures of searchability with respect to local search. Two such examples are fitness distance correlation (Jones and Forrest, 1995) and the information landscape hardness measure (Borenstein and Poli, 2005b). Both these techniques require knowledge of the global optima, and so cannot be used as predictive measures of algorithm performance on unknown problems when used in their original form. An alternative is to base the measure on a sample of the search space and to use the best point from the sample in the place of the global optimum. This implies a shift in focus away from measuring hardness to measuring searchability, since the aim is no longer to quantify how well or badly the problem guides search towards the optimum, but rather to quantify how well or badly the problem guides search towards a place in the search space where objective function values improve.

There are also techniques that were specifically designed to visualise or quantify evolvability for evolutionary algorithms. Three such techniques include fitness evolvability portraits (Smith et al, 2002), fitness clouds (Verel et al, 2003) and fitness-probability clouds (Lu et al, 2011). Although visual plots are potentially useful for human analysis, numerical output is more useful for facilitating automated analysis of problem features for performance prediction. The negative slope coefficient (Vanneschi et al, 2004) is a numerical output measure that quantifies the evolvability of a fitness landscape and is based on fitness clouds. Similarly, the accumulated escape probability (Lu et al, 2011) is a numerical output measure that quantifies evolvability based on a fitness-probability cloud, but is restricted to problems with a discrete representation.

This section describes fitness distance correlation (Jones and Forrest, 1995), information landscape hardness (Borenstein and Poli, 2005b), fitness clouds (Verel et al, 2003) and the negative slope coefficient (Vanneschi et al, 2004), and proposes ways of adapting these techniques to measure searchability of continuous problems in the context of PSO algorithms.

2.1 Fitness distance correlation

Fitness distance correlation (FDC) (Jones and Forrest, 1995) was introduced as a way of predicting the performance of a genetic algorithm (GA) on problems with known global optima and measures the correlation between the objective value of solutions and the distance to the nearest global optimum.

2.1.1 Original formulation

Given a set of n points with associated objective function values $F = \{f_1, \ldots, f_n\}$ and distances of each point to the nearest global optimum in search space $Dist = \{d_1, \ldots, d_n\}$, the FDC is calculated as the covariance of F and $Dist$ divided by the product of the standard deviation of F and standard deviation of $Dist$, or:

$$FDC = \frac{\frac{1}{n}\sum_{i=1}^{n}(f_i - \overline{f})(d_i - \overline{d})}{\sigma(F)\sigma(Dist)} \tag{1}$$

where $\overline{f}$, $\overline{d}$, $\sigma(F)$ and $\sigma(Dist)$ are the means of F and $Dist$ and the standard deviations of F and $Dist$, respectively.

The FDC measure takes on values from −1 (perfect anti-correlation) to +1 (perfect correlation), where for maximisation problems, low values (≤ -0.15) are regarded as easy, values around 0 difficult and higher values (≥ 0.15) misleading. In the original study (Jones and Forrest, 1995), Hamming distance was used as the measure of distance, but a number of subsequent studies have proposed alternatives, such as the use of crossover to determine distance (Altenberg, 1997) and distance metrics between trees for genetic programming problems (Vanneschi, 2004). Jones and Forrest (1995) showed empirically that the FDC measure is a reliable indicator of GA performance on a wide range of problems.

A significant limitation of the FDC technique is that the optimal solution(s) must be known beforehand. It has also been shown that it does not reliably predict the difficulty of optimising the problem (Altenberg, 1997; Jansen, 2001; Naudts and Kallel, 1998, 2000; Quick et al, 1998; Reeves, 1999).

2.1.2 Fitness distance correlation as a searchability measure

The following adapted FDC measure is proposed for continuous problems. Given a sample of n points, $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, from the search space, with associated objective function values $F = \{f_1, \ldots, f_n\}$, the best point in the sample is determined and denoted $\mathbf{x}^*$. The Euclidean distances of every point $\mathbf{x}_i$ from $\mathbf{x}^*$ are calculated and denoted as $Dist^* = \{d_1^*, \ldots, d_n^*\}$. The fitness distance correlation searchability (FDC_s) measure is defined as the covariance of F and $Dist^*$ divided by the product of the standard deviation of F and standard deviation of $Dist^*$. Since these are samples, this can be estimated and simplified to

$$\text{FDC}_s = \frac{\sum_{i=1}^{n}(f_i - \overline{f})(d_i^* - \overline{d^*})}{\sqrt{\sum_{i=1}^{n}(f_i - \overline{f})^2}\sqrt{\sum_{i=1}^{n}(d_i^* - \overline{d^*})^2}} \tag{2}$$

where $\overline{f}$ and $\overline{d^*}$ are the means of F and $Dist^*$, respectively.

2.2 Information landscape measures

Borenstein and Poli (2005a,b) introduced the concept of an *information landscape*: a matrix of all possible comparisons between solutions based on objective function values.

2.2.1 Original formulation

Given an objective function f of a minimisation problem and a set X of discrete solutions, an information matrix M is defined as having $|X| \times |X|$ entries $m_{i,j} = t(x_i, x_j)$, where

$$t(x_i, x_j) = \begin{cases} 1 & \text{if } f(x_i) < f(x_j) \\ 0.5 & \text{if } f(x_i) = f(x_j) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Only a subset of the elements in the information matrix are necessary to define the information landscape. Ignoring duplicated entries due to symmetry, entries on the diagonal and the row and column of the optimal solution, the information matrix can be reduced to a vector $\mathbf{v} = (v_1, v_2, \ldots, v_m)$, where $|\mathbf{v}| = m = (|X| - 1)(|X| - 2)/2$. Given two problems represented by information landscape vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, the *distance* between $\mathbf{v}_1$ and $\mathbf{v}_2$ is defined as

$$D(\mathbf{v}_1, \mathbf{v}_2) = \frac{1}{m}\sum_{i=1}^{m} |v_{1i} - v_{2i}| \tag{4}$$

Borenstein and Poli proposed a measure of GA hardness based on the distance between the information landscape of a problem and the information landscape of an 'optimal' landscape (referred to as $\mathbf{v}_{max}$). An optimal landscape is one which is known to be easy for a given search algorithm, or on which the algorithm will perform maximally.

2.2.2 Information landscape negative searchability measure

The information landscape hardness measure is based on the difference between the information landscape vector of a problem and a reference landscape vector (called $\mathbf{v}_{max}$ in the original study, but denoted using $\mathbf{v}_r$ in this study). The well-known Spherical function in D dimensions $\left(f(\mathbf{x}) = \sum_{i=1}^{D} x_i^2\right)$ can serve the purpose of such a reference landscape for the following reasons:

- The Spherical function is an 'optimal' landscape in that it presents no negative information for search: if any point $\mathbf{x}_i$ has a lower objective function value than another point $\mathbf{x}_j$, then $\mathbf{x}_i$ will be closer to the optimum than $\mathbf{x}_j$.
- The Spherical function can be defined up to any dimension and is defined for all values of $\mathbf{x}$, so the domain can be set to match the domain of any real-encoded problem.
- The Spherical function can be shifted so that the optimum is positioned anywhere in the search space, so that it coincides with the estimated optimum of the problem landscape.

Given the above, it is proposed that the information landscape hardness measure be adapted to an information landscape negative searchability measure (IL_{ns}) using the approach outlined in Algorithm 1. The measure is referred to as a negative searchability measure because high values are indicative of bad information for search.

Algorithm 1 Algorithm for computing the IL_{ns} (information landscape negative searchability) measure for a minimisation problem.

1: Generate a sample of n random points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from a uniform distribution of the search space of problem p with dimension D.
2: Determine the position of the best solution in the sample, $\mathbf{x}^*$.
3: Construct vector $\mathbf{v}_p$ representing the information matrix of the problem using Equation 3.
4: Define reference function f_r as $f_r(\mathbf{x}) = \sum_{i=1}^{D}(x_i - x_i^*)^2$.
5: Using the same sample of points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, and based on f_r, construct vector $\mathbf{v}_r$ representing the information matrix of the reference landscape.
6: Compute IL_{ns} as the difference between $\mathbf{v}_p$ and $\mathbf{v}_r$ using Equation 4.

2.3 Fitness cloud

Verel et al (2003) introduced fitness clouds for visualizing the evolvability of evolutionary search. The fitness cloud is a scatterplot showing the relationship between objective function values of parents and offspring. For each solution $\mathbf{x}$ in the search space (where $\mathbf{x}$ is a binary string in the original formulation), a neighbour of $\mathbf{x}$, called $\mathbf{x}'$, is determined based on some genetic search operator. The fitness cloud is then a scatterplot of all points $(f(\mathbf{x}), f(\mathbf{x}'))$ where f is the objective function. The line $f(\mathbf{x}) = f(\mathbf{x}')$ in the scatterplot forms the division between points with good evolvability and points with bad evolvability, with points falling on the line being indicative of neutrality in a fitness landscape.

2.3.1 Determining neighbours for fitness clouds using PSO

Constructing a fitness cloud for a given problem requires a sample of solutions and neighbours of those solutions. In the original fitness cloud publication (Verel et al, 2003) two solutions are regarded as neighbours if there is "*a transformation related to a local search heuristic or an operator which allows it to pass*" from one solution to another. For PSO algorithms, the search operators are in the form of position update equations. Three update models with differing levels of exploration/exploitation are proposed for calculating neighbours to be used as the basis for generating fitness clouds. These are the cognitive-only model (using only the personal best as a guide), the social-only model (using only the global best as a guide) and the traditional model (combining the personal and global bests as guides).

Given a problem in multi-dimensional real space, the position of each particle i at iteration t of the algorithm can be represented as $\mathbf{x}_i(t)$. The swarm of particles at $t = 0$ are initialised with random positions and at each iteration of the algorithm, the positions of particles are updated as follows:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{5}$$

where $\mathbf{v}_i(t+1)$ is the velocity of particle i at time $(t+1)$. The three velocity update models used for determining neighbours are as follows:

1. Cognitive-only PSO update (Kennedy, 1997):

$$\mathbf{v}_i(t+1) = w \cdot \mathbf{v}_i(t) + c_1 \cdot \mathbf{r}_1(t) \odot (\mathbf{y}_i(t) - \mathbf{x}_i(t)) \tag{6}$$

   where w is the inertia weight, c_1 is the cognitive acceleration constant, $\mathbf{r}_1(t) \sim U(0,1)^D$ where D is the dimension of the problem, $\odot$ denotes element-by-element vector multiplication, $\mathbf{y}_i(t)$ refers to particle i's personal best position, and $\mathbf{y}_i(0) \neq \mathbf{x}_i(0)$.
2. Social-only PSO update (Kennedy, 1997):

$$\mathbf{v}_i(t+1) = w \cdot \mathbf{v}_i(t) + c_2 \cdot \mathbf{r}_2(t) \odot (\hat{\mathbf{y}}(t) - \mathbf{x}_i(t)) \tag{7}$$

   where c_2 is the social acceleration constant, $\mathbf{r}_2(t) \sim U(0,1)^D$, and $\hat{\mathbf{y}}(t)$ refers to the global best position at time step t, being the best solution from the personal best positions of all particles.
3. Traditional PSO model (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995):

$$\begin{aligned}\mathbf{v}_i(t+1) = w \cdot \mathbf{v}_i(t) &+ c_1 \cdot \mathbf{r}_1(t) \odot (\mathbf{y}_i(t) - \mathbf{x}_i(t)) \\ &+ c_2 \cdot \mathbf{r}_2(t) \odot (\hat{\mathbf{y}}(t) - \mathbf{x}_i(t))\end{aligned} \tag{8}$$

   The velocity update of the traditional PSO model (assuming a star neighbourhood topology) combines the cognitive and social velocity updates into one velocity update that has the effect of pulling each particle in a direction somewhere between the global and personal best particle positions (Van Den Bergh and Engelbrecht, 2006).

The approach used for constructing the fitness cloud of a minimisation problem using PSO updates for determining neighbours is given in Algorithm 2. The basic idea is to perform two PSO updates on the initial swarm of solutions to determine the neighbours. The reason for two updates is that the initial velocity of all particles is zero. This means that the inertia term $(w \cdot \mathbf{v}_i(t))$ will be zero for the first update and will only come into effect on the second iteration of the algorithm. Note that steps 3 to 8 of Algorithm 2 ensure that the personal best particle is not the same as the current particle in the first iteration. If this was not done, then both the first and second terms of Equation 6 will be zero, resulting in no particles moving. The strategy used to prevent this is to generate a new random solution a small distance from the initial solution (by adding Gaussian noise) and to swap these points if the new solution is not better than the initial solution.

Algorithm 2 Algorithm for constructing a fitness cloud of a minimisation problem using PSO updates to determine neighbours of a sample.

1: Generate a sample swarm of n random solution vectors for iteration 0: $\mathbf{x}_1(0), \ldots, \mathbf{x}_n(0)$ from a uniform distribution of the search space of the problem.
2: Determine the objective function values of all solutions, $f(\mathbf{x}_1(0)), \ldots, f(\mathbf{x}_n(0))$.
3: **for** each solution, $\mathbf{x}_i(0)$, generate a personal best position, $\mathbf{y}_i(0)$ as follows:
4: Generate a new position vector, $\mathbf{z}$, a small distance from $\mathbf{x}_i(0)$, but still in the domain of the problem, by adding Gaussian noise with a standard deviation equal to 10% of the range of the problem to each component of $\mathbf{x}_i(0)$.
5: Determine the objective function value of the new position vector, $f(\mathbf{z})$.
6: If $f(\mathbf{z}) < f(\mathbf{x}_i(0))$, then set $\mathbf{y}_i(0) = \mathbf{z}$.
7: Else set $\mathbf{y}_i(0) = \mathbf{x}_i(0)$ and set $\mathbf{x}_i(0) = \mathbf{z}$.
8: **end for**
9: Determine the global best solution of iteration 0, $\hat{\mathbf{y}}(0)$, selected from the personal best positions $\mathbf{y}_i(0)$.
10: Set all initial velocities $\mathbf{v}_i(0)$ to zero.
11: For each $\mathbf{x}_i(0)$, determine the velocity update $\mathbf{v}_i(1)$ using the specific PSO update equation and calculate the iteration 1 positions: $\mathbf{x}_1(1), \ldots, \mathbf{x}_n(1)$, repairing any positions outside the bounds of the search space on any dimension to be set on the boundary for that dimension.
12: Determine the iteration 1 personal best solutions $\mathbf{y}_1(1), \ldots, \mathbf{y}_n(1)$ and the global best solution $\hat{\mathbf{y}}(1)$, selected from the set of personal best positions $\mathbf{y}_i(1)$.
13: For each $\mathbf{x}_i(1)$, determine repaired positions $\mathbf{x}_i(2)$ based on calculated $\mathbf{v}_i(2)$ velocity updates (as in step 11).
14: Determine the objective function values of all iteration 2 positions, $f(\mathbf{x}_1(2)), \ldots f(\mathbf{x}_n(2))$.
15: Normalise the objective function values of all initial points, $f(\mathbf{x}_1(0)), \ldots, f(\mathbf{x}_n(0))$, and final neighbours, $f(\mathbf{x}_1(2)), \ldots f(\mathbf{x}_n(2))$, to the range $[0, 1]$, where 0 is the worst objective function value and 1 is the best objective function value, and generate the fitness cloud from the normalised objective function values.

Steps 9 to 13 involve the two PSO update operations. In the position update steps, particles are constrained at the boundary of the search space to avoid the fitness cloud containing objective function values corresponding to points outside the domain of the problem. This is necessary for PSO, because there is a good chance that particles will leave the search space, even in the first iteration of the algorithm. This was proved theoretically to be the case even when initial velocities are set to zero, particularly in the case of high-dimensional search spaces (Helwig and Wanka, 2008) and has been supported by empirical evidence (Engelbrecht, 2012).

In step 15, the objective function values of all initial points and neighbours are normalised to the range $[0, 1]$, where 0 is the worst objective function value and 1 is the best objective function value (note that the best and worst objective function values are as encountered during execution of the algorithm, so that the algorithm can be run on unknown problems). Normalising the objective function values in this way effectively converts the minimisation problem into a maximisation problem for the purposes of the fitness cloud visualisation and allows for comparisons between fitness clouds of different problems. If the fitness cloud scatterplot is drawn with the original objective function values of a minimisation problem, it will have to be interpreted in the opposite way to the original fitness cloud approach. Points below the diagonal would be regarded as having good searchability, rather than bad searchability. This 'upside down' fitness cloud causes problems later with the computation of the negative slope coefficient measure, since a negative slope would indicate good searchability, rather than bad searchability. To avoid this confusion and need to redefine terminology, the objective function values are converted to behave as for a maximisation problem.

2.3.2 Proposed measures based on fitness clouds

The first proposed single-valued measure based on a fitness cloud is simply the proportion of points in the cloud for which the objective function value improved. This is termed the fitness cloud index (FCI) and is calculated as follows: Given a minimisation problem with an objective function f and a sample $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of n points with associated neighbours $\{\mathbf{x}'_1, \ldots, \mathbf{x}'_n\}$, the FCI measure is defined as:

$$FCI = \frac{\sum_{i=1}^{n} g(\mathbf{x}_i)}{n} \tag{9}$$

where

$$g(\mathbf{x}_i) = \begin{cases} 1 \text{ if } f(\mathbf{x}'_i) < f(\mathbf{x}_i) \\ 0 \text{ otherwise} \end{cases} \tag{10}$$

Note that the FCI measure is based on a simple comparison of neighbouring objective function values, and so does not require normalisation of the objective function values or conversion to a maximisation problem (step 15 of Algorithm 2). By definition of being a proportion, the FCI measure is normalised to the range $[0, 1]$, where 0 indicates the worst possible searchability and 1 indicates perfect searchability of the problem with respect to the given search operator.

2.4 Negative slope coefficient

Vanneschi et al (2004) proposed a measure of problem difficulty called the negative slope coefficient (NSC), based on the fitness cloud. The NSC is calculated by partitioning the fitness cloud into discrete bins. Line segments are defined between the centroids of adjacent bins and the NSC is calculated as the sum of all negative slopes between segments. Vanneschi et al hypothesised that the NSC measure could be used as a predictive difficulty measure for problems: if NSC=0 the problem is

Algorithm 3 Algorithm for calculating the NSC measure based on a fitness cloud.

1: Partition the horizontal axis of the fitness cloud (the $f(x)$ axis) into m segments $I_1, I_2, \ldots, I_m$.
2: Partition the vertical axis of the fitness cloud (the $f(x')$ axis) into m segments $J_1, J_2, \ldots, J_m$ so that each segment J_i contains all the $f(x')$ values corresponding to the $f(x)$ values in I_i.
3: For each partition I_i, compute the average objective function value, M_i.
4: For each partition J_i, compute the average objective function value, N_i.
5: Define segments $S_1, S_2, \ldots, S_{m-1}$, such that S_i is the segment from point (M_i, N_i) to point (M_{i+1}, N_{i+1}).
6: For each segment S_i, calculate the slope P_i using $P_i = \frac{N_{i+1}-N_i}{M_{i+1}-M_i}$
7: Calculate the negative slope coefficient measure as: $NSC = \sum_{i=1}^{m-1} c_i$, where $\forall i \in [1, m)$: $c_i = \begin{cases} P_i & \text{if } P_i < 0 \\ 0 & \text{otherwise} \end{cases}$

easy, but if NSC<0, the problem is difficult and smaller values indicate increased difficulty. The algorithm for calculating NSC is given in Algorithm 3.

A significant aspect of computing the NSC measure involves deciding on the bin partitioning strategy (step 1 of Algorithm 3). Possible strategies include (Vanneschi, 2004): bins of equal size, bins containing equal numbers of points and size-driven bisection (which takes both the size of the bins and the number of points into account).

Given a fitness cloud based on PSO updates, Algorithm 3 can be used to compute the NSC measure for PSO. The notations NSC$_{cog}$, NSC$_{soc}$ and NSC$_{trd}$ are used to denote the NSC measures derived using the cognitive, social and traditional PSO update strategies, respectively.

2.5 Summary and discussion of proposed measures

Table 1 summarises the proposed measures in terms of the required parameters, range of values produced and the computational complexity. All measures depend on the size of the sample, n. The FCI and NSC measures have an additional two parameters for the PSO updates, while the NSC measures also require two parameters related to the binning of points. All measures have a bound on output values, except for the NSC measures, which have unbounded ranges. All measures have linear time complexity with respect to the size of the sample, but the IL$_{ns}$ has polynomial memory requirements with respect to the sample size.

The dependence of all the measures on the sample size is a significance factor that warrants further investigation. The effect of the size of the sample, sampling strategy and the level of uncertainty, due to the stochastic nature of the measures, are factors that can influence the robustness of the measures. The aim of this study is to investigate proposed measures in terms of correlation with PSO performance. Based on results from this study, further work should include a full sensitivity analysis of the measures that show promise as predictors of PSO performance.

Table 1 Proposed measures of searchability

Proposed Measure		**Parameters**	**Result: range and interpretation**	**Computational Complexity**
FDC_s	Fitness distance correlation searchability measure	size of sample, n	$[-1, 1]$: For a minimisation problem, 1 indicates the highest measure of searchability (perfect correlation between objective function values and distance to the best solution)	Time: $\mathcal{O}(n)$ Space: $\mathcal{O}(n)$
IL_{ns}	Information landscape negative searchability measure	size of sample, n	$[0, 1]$: A value of 0 indicates maximum searchability (no difference from the reference landscape vector $\mathbf{v}_r$)	Time: $\mathcal{O}(n)$ Space: $\mathcal{O}(n^2)$
FCI_{cog} FCI_{soc} FCI_{trd}	Fitness cloud index based on cognitive, social or traditional PSO updates	(1) size of sample, n, (2) inertia weight, w, (3) acceleration constants, c_1 and c_2)	$[0, 1]$: indicating the proportion of solutions for which the objective function value improved after two PSO updates	Time: $\mathcal{O}(n)$ Space: $\mathcal{O}(n)$
NSC_{cog} NSC_{soc} NSC_{trd}	Negative slope coefficient with neighbourhood defined using cognitive, social or traditional PSO updates	(1) size of sample, n, (2) inertia weight, w, (3) acceleration constants, c_1 and c_2, (4) minimum number of points in a bin, (5) minimum size of a bin	$(-\infty, 0]$: A value of 0 indicates maximum searchability (no negative slopes between centroids of bins in the fitness cloud); smaller values indicate decreased searchability	Time: $\mathcal{O}(n)$ Space: $\mathcal{O}(n)$

3 Experimentation on one-dimensional problems

This section performs experiments on the proposed measures of searchability. Simple one-dimensional benchmarks are used to see if the proposed measures give expected results, based on a visual inspection of functions. Section 4 experiments with higher dimensional problems.

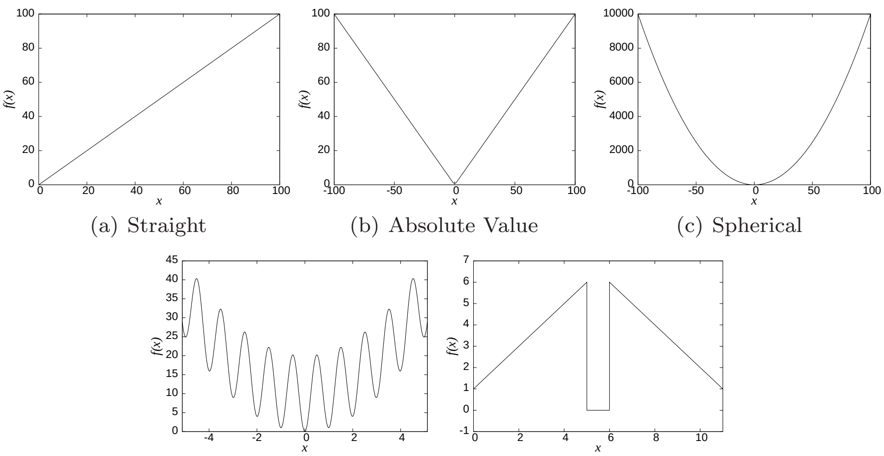
3.1 Benchmark functions

Five simple benchmark functions were selected with expected decreasing searchability. These functions are defined in Table 2 and are illustrated in Figure 1. The expected results of each of the proposed measures on the benchmark functions are discussed in this section.

The FDC_s metric quantifies the correlation between objective function values and distance to the best solution of the sample. For both the Straight and Absolute Value benchmarks, this value should be at the maximum (+1) because the functions are linear. For the Spherical function, the value should be close to 1, because there is a positive correlation, but not a perfect linear correlation be-

Table 2 One-dimensional benchmark function definitions.

Straight	$f(x) = x, \quad x \in [0, 100]$
Absolute Value	$f(x) = \lvert x \rvert, \quad x \in [-100, 100]$
Spherical	$f(x) = x^2, \quad x \in [-100, 100]$
Rastrigin	$f(x) = x^2 - 10\cos(2\pi x) + 10, \quad x \in [-5.12, 5.12]$
Hole-in-Mountain	$f(x) = x + 1, \quad \text{for } x \in [0, 5)$ $= 0, \quad \text{for } x \in [5, 6]$ $= -x + 12, \quad \text{for } x \in (6, 11]$



(a) Straight (b) Absolute Value (c) Spherical

(d) Rastrigin (e) Hole-in-Mountain

Fig. 1 Plots of one-dimensional benchmark functions as defined in Table 2 with different searchability characteristics.

tween objective function value and distance. Rastrigin should give a lower FDC_s value than Spherical, but still an overall positive correlation due to the underlying spherical shape of Rastrigin. It is expected that Hole-in-Mountain give a negative value for FDC_s due to the deceptive structure of the function.

The IL_{ns} measure quantifies the difference in the information landscape between the benchmark and the shifted Spherical function. For the Straight, Absolute Value and Spherical benchmarks, the IL_{ns} measure should be close to 0, as the information is identical to the shifted Spherical function. Since the optimum of the Spherical function is shifted to the position of the best solution from a sample (and not necessarily the optimum of the benchmark function), there may be slight differences in the information landscapes, but these should be small. The IL_{ns} measure should increase for Rastrigin and be closer to 1 for Hole-In-Mountain.

The FCI measures simply quantify the proportion of elements in the fitness cloud that have improved objective function values. All FCI measures should yield values of 1 for the Straight function, since if any particle is pulled in the direction of a better particle, regardless of whether it is a global or personal best guide, the objective function value can only improve. The Absolute Value and Spherical functions should give good FCI values (close to 1). Some points may be below the diagonal in the fitness cloud, since the objective function value of a particle can deteriorate if it moves in the direction of a better particle, but overshoots the global minimum and moves to a position higher than the original position. In the case of the Rastrigin function, due to the ruggedness, there are many opportunities for the objective function value of a particle to deteriorate if pulled in the direction of a better particle, regardless of whether it is a global or personal best guide, so lower FCI values are expected. The Hole-in-Mountain function should result in low searchability for the social-only and traditional PSO models (lower FCI values) and relatively high searchability for the cognitive-only PSO model. For the social-only and traditional models, assuming one of the initial random points was positioned in the 'hole' (the global minimum plateau), particles will be pulled towards the centre, resulting in an increase (deterioration) of objective function value for many particles. Although this is the desired behaviour for a search algorithm (moving towards the global optimum), the measure predicts searchability, not optimality. In the case of the cognitive-only model, the simple linear slopes that define most of the Hole-in-Mountain function should give a similar, but slightly worse, searchability profile to the Absolute Value benchmark.

For the same reasons as described above, all NSC measures should result in 0 for the Straight function (perfect searchability), values close to 0 (small negative) for Absolute Value and Spherical, and smaller values (larger negative) for Rastrigin. The Hole-in-Mountain should have a smaller (larger negative) NSC_{soc} and NSC_{trd} value than the other functions, but a reasonably good NSC_{soc} value.

3.2 Experimental setup

For each benchmark problem, 30 independent runs of the algorithms for computing each measure were performed. The calculations of all measures were based on sample sizes of 500 points (randomly sampled from a uniform distribution). For the PSO updates, the inertia weight (w), cognitive acceleration (c_1) and social acceleration (c_2) were set to the popular values of 0.7298, 1.496, and 1.496, respectively (Eberhart and Shi, 2000), a parameter choice that guarantees convergence to an equilibrium state (Trelea, 2003; Van Den Bergh and Engelbrecht, 2006; Cleghorn and Engelbrecht, 2014). For the NSC measures, size-driven bisection was used for partitioning the fitness clouds with the minimum number of points in a bin set to 30 and the minimum size of a bin set to 5% of the range of the problem.

3.3 Results and discussion

Table 3 lists the mean measures over 30 runs for each benchmark function with standard deviations shown below the means in parentheses. Note that the underlying searchability measure values were tested and found to be approximately normally distributed. A study of the values reveals the following:

- The values for the FDC_s and IL_{ns} measures are in line with the expected values as discussed in Section 3.1. Relatively low standard deviations for these first two measures also indicate that the measures are fairly reliable.
- The values for the FCI_{cog} are in line with the expected values. Slightly lower FCI_{soc} and FCI_{trd} values for the Absolute Value and Spherical functions indicate that more particles overshot the minimum to higher objective function

Table 3 Values of the FDC_s, IL_{ns}, FCI and NSC measures for the one-dimensional problems shown in Table 2. Values are averages over 30 runs, each with 500 random initial points. Standard deviations are given below each value in parentheses.

Function	FDC_s	IL_{ns}	FCI_{cog}	FCI_{soc}	FCI_{trd}	NSC_{cog}	NSC_{soc}	NSC_{trd}
Straight	1	0	1	1	1	-3.058	-8.076	-6.837
	(±0)	(±0)	(±0)	(±0)	(±0)	(±1.838)	(±2.423)	(±1.799)
Absolute	1	0.002	0.964	0.876	0.788	-7.652	-12.900	-14.858
Value	(±0)	(±0.002)	(±0.009)	(±0.012)	(±0.019)	(±3.428)	(±4.913)	(±5.811)
Spherical	0.968	0.002	0.964	0.871	0.791	-12.640	-21.788	-22.149
	(±0.002)	(±0.001)	(±0.011)	(±0.016)	(±0.021)	(±4.838)	(±5.869)	(±6.079)
Rastrigin	0.709	0.254	0.781	0.800	0.780	-11.774	-21.860	-13.858
	(±0.018)	(±0.011)	(±0.017)	(±0.019)	(±0.017)	(±8.684)	(±9.810)	(±7.906)
Hole-in-	-0.401	0.795	0.928	0.430	0.387	-3.828	-8.587	-5.508
Mountain	(±0.052)	(±0.031)	(±0.012)	(±0.048)	(±0.024)	(±5.122)	(±11.039)	(±9.776)

values than for the cognitive-only model, most probably due to the higher velocities. As predicted, the Hole-in-Mountain function has high searchability for the cognitive-only PSO model (0.928) and low searchability for the social-only (0.430) and traditional (0.387) PSO models. To see this difference visually, the fitness clouds of sample runs on the Hole-in-Mountain function are plotted in Figure 2. It is clear from the plots that the cognitive PSO updates result in high levels of searchability, while the social PSO updates result in low levels of improvement in objective function value.

- All of the NSC values are negative, which is not in line with the expected values. For example, the Straight function should have perfect PSO searchability and yet the mean NSC values are negative, which is supposed to indicate areas of negative searchability. It is also unexpected that the Spherical function have relatively low NSC measures in relation to the other problems.

The experiments on simple one-dimensional problems show that the NSC values did not give results as predicted. For an investigation into possible reasons why the NSC values based on PSO updates are unpredictable, see (Malan, 2014). Since the NSC measure does not give meaningful values on one-dimensional problems, the measure is not used further in this study.

4 Linking to PSO performance on higher dimensional problems

Results in the previous section show that the FDC_s, IL_{ns} and FCI measures are meaningful indicators of different aspects of searchability for the simple one-dimensional functions studied. This section evaluates the measures on higher dimensional benchmark problems. The performance of a traditional global best (gbest) PSO algorithm (Kennedy and Eberhart, 1995) on the same benchmarks is evaluated and the link between the searchability measures and actual algorithm performance is investigated.

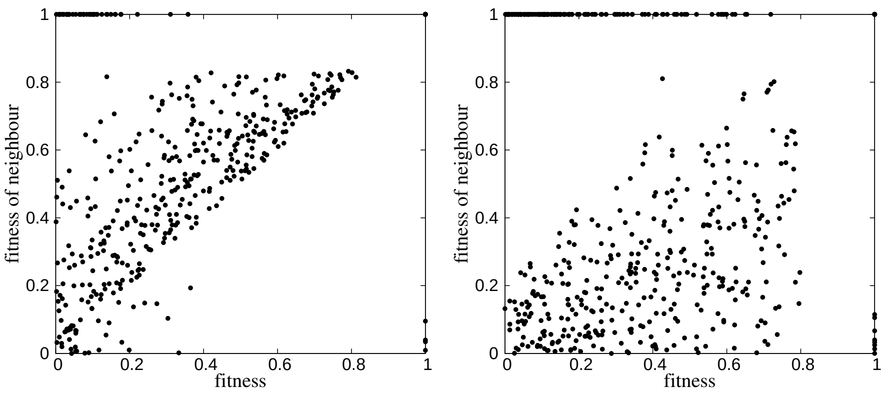


Fig. 2 Fitness clouds from sample runs on Hole-in-mountain function using (a) Cognitive PSO and (b) Social PSO update strategies.

4.1 Benchmark problems and expected results

The benchmark functions used to test the proposed measures in different dimensions are defined in Table 4 and one-dimensional versions of some of these functions are plotted in Figure 3. The functions were used for dimensions 1 (not applicable for Rana and Rosenbrock), 2, 5, 15 and 30. These functions cover a range of characteristics. All are multimodal, except for Spherical, Quadric, and Rosenbrock for dimensions 1 to 3. (Note that although the Rosenbrock function is widely stated as unimodal, it has been shown to be multimodal for dimensions of 4 and higher (Shang and Qiu, 2006).) Although Quadric (also known as Schwefel 1.2) is unimodal and is equivalent to Spherical in 1 dimension, it has been shown to have a weak fitness distance correlation (to the known optimum) in higher dimensions (Müller and Sbalzarini, 2011). Functions Griewank and Step are rugged, but the underlying shapes match the Spherical function, so the fitness distance values should be similar to Spherical in higher dimensions. Likewise, the underlying shape of Salomon is the same as the Absolute Value function, so should also give relatively high fitness distance values. Rana and Schwefel 2.26 are multi-funnelled and so should give lower fitness distance values in higher dimensions. The IL_{ns} measure quantifies the difference between the information landscape of the problem and the information landscape of Spherical. Any problem that shares the same basic underlying shape as Spherical should therefore have a low IL_{ns} measure. All functions that have high (or low) FDC_s values, should therefore have low (or high) IL_{ns} values. For the FCI values, it is difficult to predict the effect of just two PSO updates on objective function values, particularly in higher dimensions.

4.2 Experimental setup

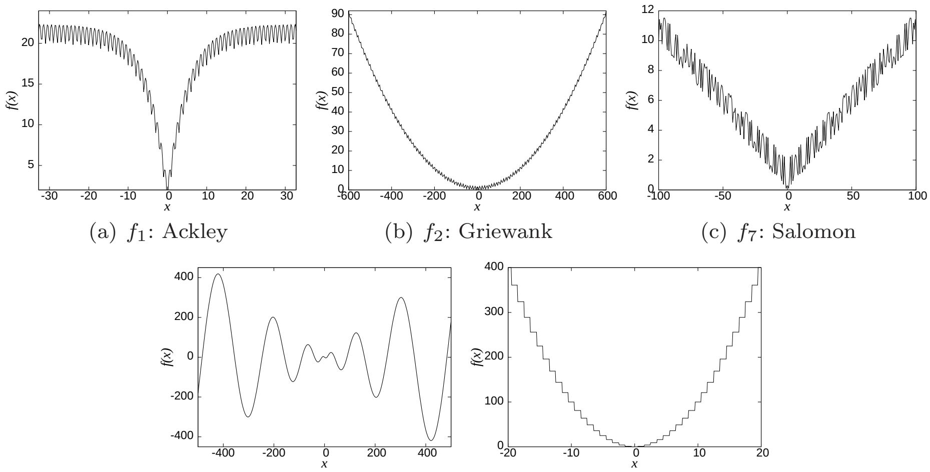
For the FDC_s calculations, uniform random samples of $500 \times D$ (dimension) were used. This involves $500 \times D$ objective function evaluations and $500 \times D$ Euclidean

Table 4 Benchmark Functions, where D is the dimension of the problem and f^* denotes the global optimum of f.

Function	Definition, domain and global optimum (f^*)
Ackley	$f_1(\mathbf{x}) = -20 \exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{\sum_{i=1}^{D}\cos(2\pi x_i)}{D}\right) + 20 + e$ $x_i \in [-32, 32], \quad f_1^* = f_1(0,\ldots,0) = 0$
Griewank	$f_2(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ $x_i \in [-600, 600], \quad f_2^* = f_2(0,\ldots,0) = 0$
Quadric	$f_3(\mathbf{x}) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$ $x_i \in [-100, 100], \quad f_3^* = f_3(0,\ldots,0) = 0$
Rana	$f_4(\mathbf{x}) = \sum_{i=1}^{D} x_i \sin(\alpha)\cos(\beta) + \left(x_{(i+1)\bmod D} + 1\right)\cos(\alpha)\sin(\beta),$ $D > 1, \quad \alpha = \sqrt{\lvert x_{i+1} + 1 - x_i \rvert}, \quad \beta = \sqrt{\lvert x_i + x_{i+1} + 1 \rvert}$ $x_i \in [-512, 512], \quad f_4^* = f_4(-512,\ldots,-512)$
Rastrigin	$f_5(\mathbf{x}) = \sum_{i=1}^{D}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$ $x_i \in [-5.12, 5.12], \quad f_5^* = f_5(0,\ldots,0) = 0$
Rosenbrock	$f_6(\mathbf{x}) = \sum_{i=1}^{D-1}\left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right), \quad D > 1$ $x_i \in [-2.048, 2.048], \quad f_6^* = f_6(1,\ldots,1) = 0$
Salomon	$f_7(\mathbf{x}) = -\cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2} + 1$ $x_i \in [-100, 100], \quad f_7^* = f_7(0,\ldots,0) = 0$
Schwefel 2.26	$f_8(\mathbf{x}) = -\sum_{i=1}^{D}\left(x_i \sin(\sqrt{\lvert x_i \rvert})\right)$ $x_i \in [-500, 500], \quad f_8^* = f_8(420.9687,\ldots,420.9687)$
Spherical	$f_9(\mathbf{x}) = \sum_{i=1}^{D} x_i^2$ $x_i \in [-100, 100], \quad f_9^* = f_9(0,\ldots,0) = 0$
Step	$f_{10}(\mathbf{x}) = \sum_{i=1}^{D}(\lfloor x_i + 0.5 \rfloor)^2$ $x_i \in [-20, 20], \quad f_{10}^* = f_{10}(0,\ldots,0) = 0$

distance calculations. The IL_{ns} calculations were based on samples of 5000 points in all dimensions. The size of the sample was chosen to be constant due to the polynomially increasing memory requirements and was set as the same sample size as FDC_s in 10 dimensions. For the FCI measures, samples of size 500 were used. For the PSO updates, the inertia weight (w), cognitive acceleration (c_1) and social acceleration (c_2) were set to the values of 0.7298, 1.496, and 1.496, respectively.

Each of the problem instances (function and dimension combinations) was solved using a traditional gbest PSO algorithm with 50 particles and the same parameter settings as the PSO updates of the FCI measures. The algorithm was run 30 times on each problem instance and performance was quantified using the SRate, SSpeed and QMetric performance measures, described in the following section.



(a) f_1: Ackley (b) f_2: Griewank (c) f_7: Salomon

(d) f_8: Schwefel 2.26 (e) f_{10}: Step

Fig. 3 Plot of one-dimensional versions of benchmark functions defined in Table 4. Spherical and Rastrigin are plotted in Figure 1 and Quadric is not plotted as it is equivalent to Spherical in one dimension. Rana and Rosenbrock are not defined for one dimension.

.

4.3 Performance measures

This section defines what is meant by a successful run in the context of this study and describes the performance measures used, originally proposed in (Malan and Engelbrecht, 2014a).

4.3.1 Successful run and fixed accuracy level

A *run* of an algorithm on a problem is a single execution of the algorithm with a given maximum number of objective function evaluations (MaxFES) as the stopping condition. For all runs in this study, the MaxFES was set to $10000 \times D$ (dimension), equivalent to $200 \times D$ iterations using a swarm of 50 particles. A *successful run* of an algorithm on a problem is a run that finds a solution with an objective function value that is within a set fixed accuracy level from the objective function value of the global optimum. Suganthan et al (2005) define fixed accuracy levels for each benchmark function, such as 10^{-6} for F_1 (Shifted Sphere Function) and 10^{-2} for F_6 (Shifted Rosenbrock's Function). The approach used in this study was to determine a fixed accuracy level for each function/dimension combination based on the estimated range of objective function values. For each problem, the fixed accuracy level was set to the estimated range of objective function values rounded down to the nearest 10^n (where n is an integer) multiplied by 10^{-8}, up to a maximum fixed accuracy level of 10^{-3}. For example, the Ackley benchmark function in one dimension with domain $[-32, 32]$ has a rounded down range of objective function values of 10^1, resulting in a fixed accuracy of 10^{-7}.

4.3.2 Success rate

The success rate (SRate) is defined as the number of successful runs divided by the total number of runs (Suganthan et al, 2005). SRate is a value in the range $[0, 1]$ where 1 indicates the highest possible rate of success.

4.3.3 Success speed

The number of objective function evaluations taken to reach the global optimum (within the fixed accuracy level) for a given run r is known as FES_r. The success speed of a run r (SSpeed_r) is defined as:

$$\text{SSpeed}_r = \begin{cases} 0 & \text{if run not successful} \\ \frac{\text{MaxFES}-(\text{FES}_r-1)}{\text{MaxFES}} & \text{otherwise.} \end{cases} \quad (11)$$

The metric SSpeed_r is a value in the range $[0, 1]$. The highest value for SSpeed_r can only be obtained if the global optimum is reached in the first objective function evaluation (if FES_r is 1) and this would indicate the highest possible performance in terms of speed. The success speed (SSpeed) over ns successful runs, is defined as:

$$\text{SSpeed} = \begin{cases} \frac{\sum_{r=1}^{ns} \text{SSpeed}_r}{ns} & \text{if } ns > 0 \\ 0 & \text{if } ns = 0 \; . \end{cases} \quad (12)$$

4.3.4 Quality metric

Given a run of an optimisation algorithm on benchmark function f with resulting best objective function value found f^{min}, the difference in objective function value between the best found solution and the optimal solution, f^*, is quantified as $f^{min} - f^*$. This difference is an absolute measure of error, where 0 is the minimum error and corresponds with the highest possible solution quality. To convert the error into a positive measure of quality in the range $[0, 1]$, the found solution, f^{min}, is subtracted from the estimated maximum objective function value, $\hat{f}$, and scaled by the estimated range of objective function values:

$$q = \frac{\hat{f} - f^{min}}{\hat{f} - f^*} \; . \quad (13)$$

To better distinguish between q values closer to 1, the value of q is scaled exponentially to produce the QMetric measure as follows:

$$\text{QMetric} = 2^{q^{10^4}} - 1 \; . \quad (14)$$

For example, given a problem with an objective function value range of $[0, 1]$ (with associated fixed accuracy level of 10^{-8}) and global optimum of 0, a best found solution of 10^{-8} would be regarded as a successful run. The resulting QMetric value would be 1.000 (rounded to 3 decimal places), indicating the highest rounded solution quality. On the other hand, a solution of 10^{-5} would result in a q value of 0.99999 and an associated QMetric value of 0.872, indicating a lower solution quality. Any solution with a q value of 0.001 and larger will result in a QMetric value of 0 (rounded to 3 decimal places).

4.4 Searchability measures results

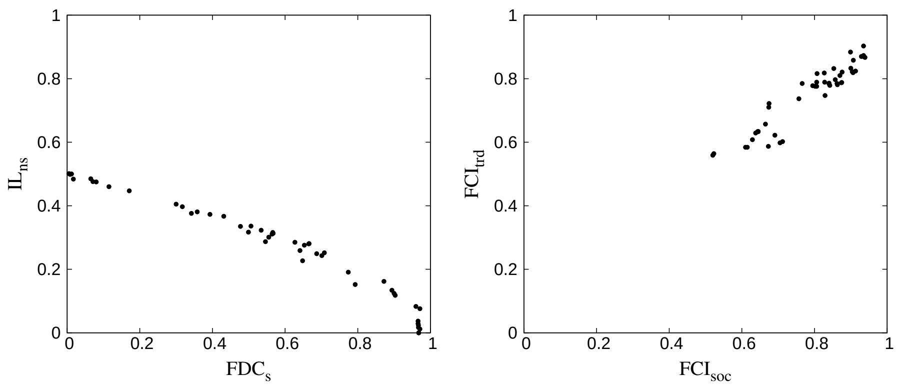
The results are summarised in Table 5. For each problem / dimension combination, the five searchability measures, FDC_s, IL_{ns}, FCI_{cog}, FCI_{soc}, and FCI_{trd} are reported as mean values based on 30 runs of the algorithm with standard deviations (given the approximately normal distribution of the underlying data).

The values of FDC_s range from values close to 1 for Spherical (f_9) in low dimensions, to as low as 0.01 for Rana (f_4) in higher dimensions. For most functions the value of FDC_s decreases as the dimension increases. For example, Spherical (f_9) reduces from 0.97 in $1D$ to 0.57 in $30D$. The fitness distance correlation coefficient of the Spherical function based on the true optimum should stay close to 1 for any dimension (Müller and Sbalzarini, 2011). Estimating the optimum can, however, lead to lower FDC_s values, if the estimated optimum is not close to the true optimum. As a simplified example, consider the Spherical function in $1D$ with an inadequate sample of three points: $x_0 = -2$, $x_1 = 1$, $x_2 = 2$, with associated objective function values $f_9(x_0) = 4$, $f_9(x_1) = 1$, $f_9(x_2) = 4$. The estimated minimum of this sample is therefore at x_1 and the associated distance values to x_1 are $d_0 = 3$, $d_1 = 0$, and $d_2 = 1$. The resulting FDC_s value is approximately 0.76, which is not a reflection of the perfect bowl shape of the Spherical function. In a similar way, a sample of 500×30 points in 30 dimensions is an inadequate sample and results in FDC_s values that are lower than the true FDC values. The question then is whether the FDC_s measure provides meaningful information if the value is far from the true FDC value. Investigating the values in Table 5, it would seem that the relative FDC_s values within each dimension are consistent. For example, the $1D$ functions with high FDC_s values (> 0.9) are Griewank (f_2), Quadric (f_3), Salomon (f_7), Spherical (f_9) and Step (f_{10}). In $30D$, the group of functions with the highest FDC_s values (> 0.5) are Griewank (f_2), Salomon (f_7), Spherical (f_9) and Step (f_{10}). The only function that is in the first group and not in the second group is Quadric, which is equivalent to Spherical in $1D$, but has been shown to have a weak FDC in higher dimensions (Müller and Sbalzarini, 2011). The functions with the lowest FDC_s values in $30D$ are Schwefel 2.26 (f_8) (0.06) and Rana (f_4) (0.01), which are both multi-funnelled landscapes, so are expected to have the lowest values. It would seem, therefore, that there is value in the *relative* FDC_s values in the different dimension groups.

In lieu of discussing the IL_{ns} values in Table 5, Figure 4(a) shows a scatterplot of the FDC_s values against the IL_{ns} values. It is clear that there is a very strong negative correlation between the FDC_s and IL_{ns} values (Spearman's correlation coefficient of -0.990). This shows that although the two measures use very different approaches, they are capturing essentially the same information on the fitness landscape: high fitness distance correlation seems to imply an information landscape that is similar to the Spherical landscape. Alternatively, an information landscape that is very different from the Spherical landscape seems to imply low fitness distance correlation.

Figure 4(b) shows that there is also a strong correlation between the FCI_{soc} and FCI_{trd} measures (Spearman's coefficient of 0.925). This shows that there is a strong link between the proportion of improved solutions resulting from two steps of the social PSO update and the proportion of improved solutions resulting from two steps of the traditional PSO update. In contrast, there is a weak correlation between the FCI_{cog} and FCI_{trd} measures (Spearman's coefficient of 0.372) and between the FCI_{cog} and FCI_{soc} measures (Spearman's coefficient of 0.454). A possible reason for the high correlation between FCI_{soc} and FCI_{trd} is the relatively large influence of the final social term in Equation 8. Although the constants c_1 and c_2 are equal, the distance of particles to the gobal best particle is usually larger than the distance to the personal best particle, resulting in a dominant third term in most cases.

Other than the strong correlations between measures illustrated in Figure 4, there is also a fairly strong correlation between the FDC_s and FCI_{soc} measures (Spearman's coefficient of 0.799). This means that there is a link between these measures: when the objective function values and distance are highly correlated, PSO social updates will probably result in an improvement in objective function value. The correlation between FDC_s and FCI_{cog} is only moderate (Spearman's coefficient of 0.546), reflecting that there is some limited overlap in the information captured by these measures, but that each measure captures something different.



(a) FDC_s and IL_{ns} measures (-0.990). (b) FCI_{soc} and FCI_{trd} measures (0.925).

Fig. 4 Scatter diagrams showing the correlation between searchability measures that show strong correlation. Spearman's correlation coefficient values are given in parentheses in the sub-captions.

4.5 PSO performance and searchability measures

The last three columns of Table 5 give the performance metrics, based on 30 runs of the PSO algorithm. For a feature metric to be useful it should show some correlation (or negative correlation) to performance. Figures 5(a) to 5(d) show the scatter plots of four of the searchability measures against the QMetric performance measure. The scatter plot of the FCI_{trd} measure is not shown, due to the high similarity to the FCI_{soc} diagram in Figure 5(d). All measures (except for the FCI_{trd} with a Spearman's coefficient of 0.322) show a moderate correlation or negative correlation to QMetric. The measure that provides the strongest correlation is the FCI_{cog} measure (Spearman's correlation coefficient of 0.662). These results show

Table 5 Searchability landscape measures alongside PSO performance metrics for benchmark functions f defined in Table 4 in different dimensions (D). All searchability measures are means over 30 independent runs and standard deviations are given next to each mean. Performance metrics are QMetric, SRate and SSpeed.

f	D	Searchability measures					Performance metrics		
		FDC_s	IL_{ns}	FCI_{cog}	FCI_{soc}	FCI_{trd}			
f_1	1	0.79 ±0.01	0.15 ±0.00	0.80 ±0.02	0.88 ±0.01	0.82 ±0.02	1.00	1.00	0.47
f_1	2	0.77 ±0.01	0.19 ±0.00	0.77 ±0.01	0.91 ±0.02	0.86 ±0.02	1.00	1.00	0.63
f_1	5	0.70 ±0.03	0.24 ±0.01	0.74 ±0.02	0.93 ±0.01	0.90 ±0.02	1.00	1.00	0.78
f_1	15	0.51 ±0.03	0.34 ±0.01	0.73 ±0.02	0.90 ±0.03	0.88 ±0.02	0.93	0.93	0.86
f_1	30	0.43 ±0.02	0.37 ±0.01	0.72 ±0.02	0.85 ±0.04	0.83 ±0.03	0.30	0.30	0.86
f_2	1	0.97 ±0.00	0.02 ±0.00	0.93 ±0.01	0.86 ±0.01	0.79 ±0.02	1.00	1.00	0.72
f_2	2	0.97 ±0.01	0.04 ±0.02	0.90 ±0.01	0.91 ±0.01	0.82 ±0.02	0.92	0.77	0.67
f_2	5	0.90 ±0.03	0.12 ±0.02	0.84 ±0.01	0.94 ±0.02	0.87 ±0.02	0.56	0.07	0.21
f_2	15	0.65 ±0.02	0.28 ±0.02	0.77 ±0.02	0.84 ±0.06	0.78 ±0.05	0.69	0.10	0.91
f_2	30	0.57 ±0.03	0.31 ±0.01	0.70 ±0.02	0.71 ±0.11	0.60 ±0.07	0.90	0.37	0.92
f_3	1	0.97 ±0.00	0.00 ±0.00	0.96 ±0.01	0.88 ±0.02	0.79 ±0.02	1.00	1.00	0.94
f_3	2	0.65 ±0.01	0.23 ±0.00	0.91 ±0.01	0.86 ±0.01	0.80 ±0.01	1.00	1.00	0.89
f_3	5	0.34 ±0.03	0.38 ±0.01	0.88 ±0.01	0.83 ±0.02	0.79 ±0.02	1.00	1.00	0.91
f_3	15	0.12 ±0.01	0.46 ±0.01	0.86 ±0.02	0.81 ±0.03	0.78 ±0.03	1.00	1.00	0.87
f_3	30	0.07 ±0.01	0.48 ±0.00	0.86 ±0.02	0.80 ±0.02	0.78 ±0.03	1.00	1.00	0.71
f_4	2	0.02 ±0.06	0.48 ±0.02	0.73 ±0.02	0.61 ±0.28	0.58 ±0.18	0.19	0.00	0.00
f_4	5	0.01 ±0.04	0.50 ±0.01	0.70 ±0.02	0.66 ±0.16	0.66 ±0.10	0.00	0.00	0.00
f_4	15	0.01 ±0.02	0.50 ±0.01	0.70 ±0.02	0.64 ±0.10	0.63 ±0.09	0.00	0.00	0.00
f_4	30	0.01 ±0.01	0.50 ±0.01	0.70 ±0.02	0.65 ±0.12	0.63 ±0.08	0.00	0.00	0.00
f_5	1	0.71 ±0.01	0.25 ±0.00	0.78 ±0.01	0.80 ±0.02	0.78 ±0.02	1.00	1.00	0.81
f_5	2	0.64 ±0.06	0.26 ±0.02	0.74 ±0.02	0.83 ±0.02	0.82 ±0.02	1.00	1.00	0.79
f_5	5	0.50 ±0.08	0.32 ±0.02	0.73 ±0.02	0.81 ±0.06	0.82 ±0.03	0.53	0.53	0.73
f_5	15	0.39 ±0.04	0.37 ±0.01	0.71 ±0.02	0.77 ±0.07	0.78 ±0.06	0.00	0.00	0.00
f_5	30	0.36 ±0.02	0.38 ±0.01	0.71 ±0.02	0.68 ±0.11	0.72 ±0.07	0.00	0.00	0.00
f_6	2	0.55 ±0.02	0.29 ±0.00	0.89 ±0.01	0.67 ±0.03	0.71 ±0.02	1.00	1.00	0.84
f_6	5	0.69 ±0.06	0.25 ±0.02	0.85 ±0.02	0.81 ±0.05	0.79 ±0.03	0.93	0.07	0.32
f_6	15	0.55 ±0.08	0.30 ±0.03	0.76 ±0.02	0.76 ±0.06	0.74 ±0.04	0.89	0.00	0.00
f_6	30	0.48 ±0.07	0.34 ±0.02	0.67 ±0.02	0.62 ±0.10	0.58 ±0.08	0.48	0.00	0.00
f_7	1	0.97 ±0.00	0.08 ±0.00	0.82 ±0.02	0.87 ±0.01	0.81 ±0.02	1.00	1.00	0.57
f_7	2	0.96 ±0.01	0.08 ±0.00	0.80 ±0.02	0.90 ±0.01	0.83 ±0.02	1.00	1.00	0.59
f_7	5	0.87 ±0.05	0.16 ±0.02	0.77 ±0.02	0.93 ±0.02	0.87 ±0.02	0.00	0.00	0.00
f_7	15	0.63 ±0.04	0.28 ±0.02	0.71 ±0.02	0.84 ±0.06	0.79 ±0.06	0.00	0.00	0.00
f_7	30	0.53 ±0.02	0.32 ±0.01	0.66 ±0.02	0.67 ±0.08	0.59 ±0.07	0.00	0.00	0.00
f_8	1	0.32 ±0.03	0.40 ±0.01	0.82 ±0.01	0.52 ±0.02	0.56 ±0.03	1.00	1.00	0.82
f_8	2	0.30 ±0.06	0.41 ±0.01	0.78 ±0.02	0.52 ±0.04	0.56 ±0.02	0.97	0.97	0.82
f_8	5	0.17 ±0.11	0.45 ±0.02	0.77 ±0.02	0.63 ±0.08	0.61 ±0.04	0.40	0.40	0.83
f_8	15	0.08 ±0.08	0.48 ±0.02	0.77 ±0.02	0.64 ±0.06	0.63 ±0.04	0.00	0.00	0.00
f_8	30	0.06 ±0.04	0.49 ±0.02	0.77 ±0.02	0.64 ±0.07	0.63 ±0.05	0.00	0.00	0.00
f_9	1	0.97 ±0.00	0.00 ±0.00	0.96 ±0.01	0.87 ±0.01	0.79 ±0.02	1.00	1.00	0.85
f_9	2	0.97 ±0.00	0.01 ±0.01	0.91 ±0.01	0.91 ±0.01	0.82 ±0.02	1.00	1.00	0.89
f_9	5	0.90 ±0.03	0.12 ±0.02	0.85 ±0.02	0.94 ±0.03	0.87 ±0.02	1.00	1.00	0.92
f_9	15	0.67 ±0.02	0.28 ±0.01	0.77 ±0.02	0.84 ±0.05	0.78 ±0.05	1.00	1.00	0.94
f_9	30	0.57 ±0.02	0.32 ±0.01	0.71 ±0.02	0.71 ±0.07	0.60 ±0.08	1.00	1.00	0.94
f_{10}	1	0.97 ±0.00	0.03 ±0.00	0.85 ±0.01	0.83 ±0.02	0.75 ±0.02	1.00	1.00	0.99
f_{10}	2	0.97 ±0.01	0.02 ±0.01	0.88 ±0.01	0.90 ±0.02	0.82 ±0.02	1.00	1.00	0.99
f_{10}	5	0.89 ±0.04	0.13 ±0.02	0.84 ±0.02	0.93 ±0.02	0.87 ±0.02	1.00	1.00	0.98
f_{10}	15	0.66 ±0.05	0.28 ±0.01	0.77 ±0.02	0.86 ±0.06	0.78 ±0.05	1.00	1.00	0.97
f_{10}	30	0.56 ±0.01	0.31 ±0.01	0.70 ±0.02	0.69 ±0.09	0.62 ±0.08	0.92	0.90	0.86

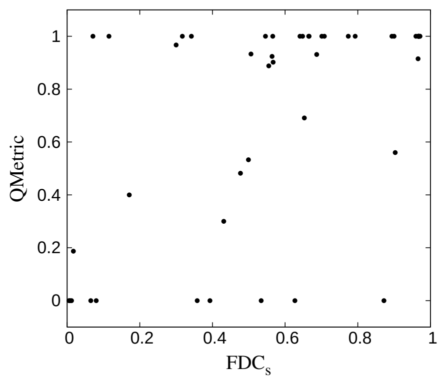
that no single searchability measure is a good predictor of hardness of a problem for PSO.

The scatterplots in Figure 5 have a large proportion of values at the top and at the bottom with a few points scattered in between. This is indicative of distinct groups of problems based on success or failure of the algorithm in solving the problem. QMetric on its own only captures part of the picture of performance, not considering, for example, how quickly a solution is found. An alternative approach to visualising the results is to allocate each problem instance solved by the algorithm into a performance class using a combination of QMetric, SRate and SSpeed values as follows:

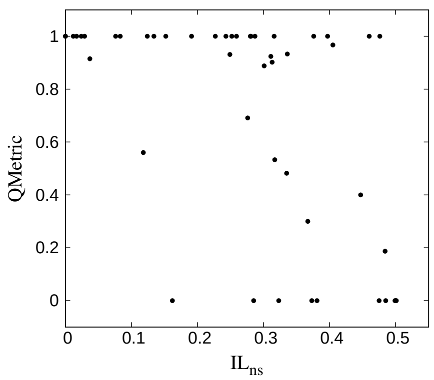
- *Always solved and fast*: problems with an SRate of 1, indicating that the solution was found for all 30 runs of the algorithm, and an SSpeed > 0.5, indicating that the algorithm was able to find the solution in less than half of the allowable time (maximum number of objective function evaluations) on average.
- *Always solved*: problems with an SRate of 1 and an SSpeed ≤ 0.5, indicating that the solution was found for all 30 runs of the PSO algorithm, but that more than half of the allowable objective function evaluations were used to find the solution on average.
- *Sometimes solved*: problems with an SRate less than 1, but greater than 0, indicating that the solution was found for some of the runs.
- *Almost solved*: problems with an SRate of 0, but a QMetric value greater than 0, indicating that although none of the runs found the solution to within the required fixed accuracy level, a solution was sometimes found that was very close to the optimum.
- *Not solved*: problems with all performance metric values equal to 0.

Figure 6 plots these classes against the searchability measures with each instance grouped according to dimension. In Figure 6, if a given searchability measure is a good predictor of PSO performance, then the order of symbols in a dimension column should match the order of symbols in the legend. Note that for the IL_{ns} measure in Figure 6(b), the symbols in the legend are displayed in reverse order, because the measure is a negative searchability measure.

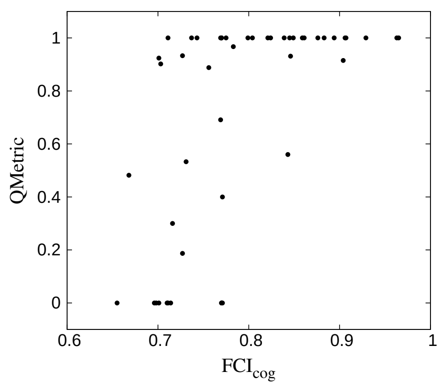
Figure 6 shows that in the one-dimensional case, all problems were solved in all cases and all except one were solved fast. The problem that took longer than the others to solve on average is the Ackley function. This could be because of the high level of ruggedness of that function. For the two-dimensional case there is one function that was almost solved (Rana) and two functions that were sometimes solved (Griewank and Schwefel 2.26). The measures that provide the best predictive value for the $2D$ case are the FDC_s and IL_{ns} measures. The order of all symbols in the plot, except for one cross (Griewank), match the legend. For the $5D$ problems, there are two problems that are not solved (Rana and Salomon), indicated by the two circles. For the Rana function, all the searchability measures are mostly in line with this failure (the highest circle in Figure 6(b) and the lowest circle in 6(a) and 6(c)). However, for the Salomon function, the searchability measures are not indicative of the failure. The steepness of the gradients for the Salomon function could be an alternative indicator of the failure in this case (Malan and Engelbrecht, 2013b). For the higher dimensional problems (15 and 30D), all searchability measures provide some value as predictors of algorithm failure, although none predict all cases correctly. These examples illustrate that



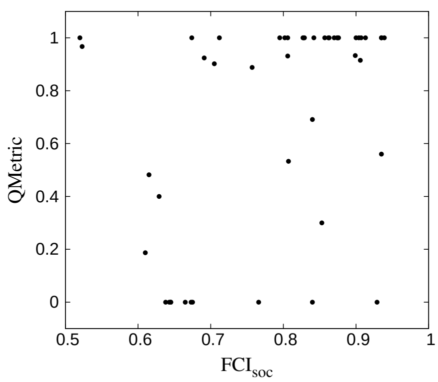
(a) FDC_s measure (0.545).



(b) IL_{ns} measure (-0.561).



(c) FCI_{cog} measure (0.662).



(d) FCI_{soc} measure (0.466).

Fig. 5 Scatter diagrams showing the correlation between the performance of a traditional PSO algorithm (as quantified by QMetric) and different searchability measures. Spearman's correlation coefficient values are given in parentheses in the sub-captions.

the four searchability measures provide some insight into the difficulty of problems for PSO algorithms, but that they do not provide the full picture of what makes a problem hard for a PSO.

4.6 Discussion

The proposed FDC_s and IL_{ns} measures can be used to quantify the searchability with respect to local search of an unknown optimisation problem. Results on a set of benchmark problems show that there is a very strong negative correlation between these measures, which indicates that the two measures capture similar information. Both measures are based on initial random samples, require a single parameter (the size of the sample) and have linear time complexity with respect to the size of the sample. The memory requirement of the IL_{ns} measure, however, is polynomial with respect to the sample size, so it may not be a suitable measure for

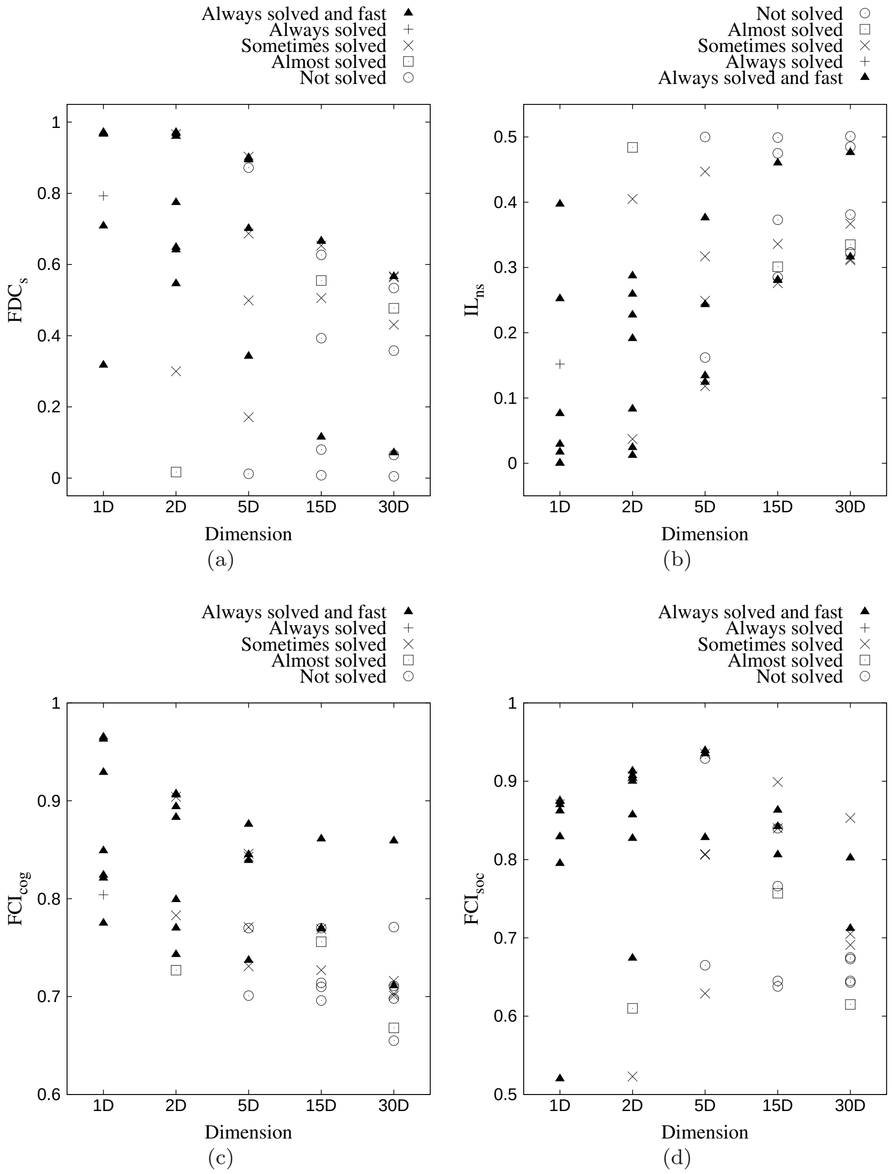


Fig. 6 Performance of a traditional PSO algorithm on benchmark problems plotted against searchability measures (a) FDC$_s$, (b) IL$_{ns}$, (c) FCI$_{cog}$, and (d) FCI$_{soc}$. Each problem instance from Table 5 is plotted using a symbol based on the actual performance of the PSO algorithm on the problem. Symbols are grouped according to dimension. If the order of the symbols in a dimension column matches the order of symbols in the legend, this indicates that the searchability measure is a good predictor of PSO performance for that dimension.

higher dimensional problems that require large sample sizes. For the benchmark problems considered, the FDC_s and IL_{ns} values were moderately correlated and negatively correlated with performance of a traditional gbest PSO algorithm (as measured by QMetric).

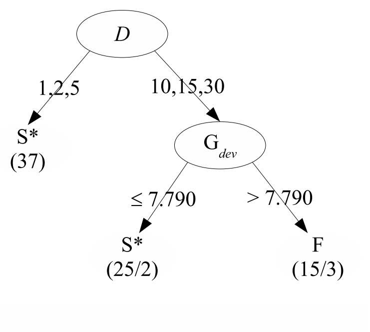
The FCI_{cog}, FCI_{soc} and FCI_{trd} measures quantify the searchability of a problem with respect to cognitive, social and combined updates in the initial stages of PSO search. For the benchmark problems considered, both cognitive and social FCI measures were moderately correlated with performance of a traditional gbest PSO algorithm (as measured by QMetric), with FCI_{cog} showing a stronger correlation than FCI_{soc}. The traditional FCI measure, FCI_{trd}, was weakly correlated with performance.

It is a premise of this study that no single problem feature on its own can serve as a predictor of problem difficulty. Instead, a range of different features need to be considered together to attempt to predict algorithm performance on an unseen problem. In this scenario, the searchability measures proposed above show potential value as part-predictors of PSO performance if used with other measures for features such as ruggedness, presence of funnels and gradients.

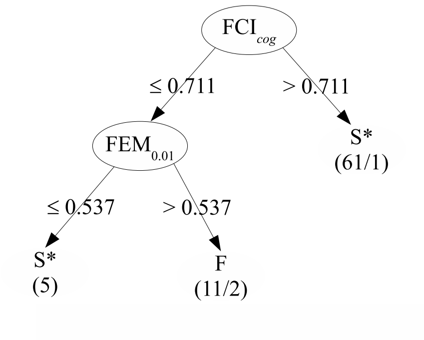
5 Predicting PSO failure with multiple fitness landscape measures

The objective of this section is to see whether classifiers can be constructed to predict failure for variations of PSO based on fitness landscape characteristics. Decision trees are used to build the classifiers, allowing for easy identification of the most relevant features in predicting failure. Furthermore, with the aim of better understanding the algorithms, it is shown how rules can be extracted from the decision trees to describe why PSO failed with reference to landscape characteristics.

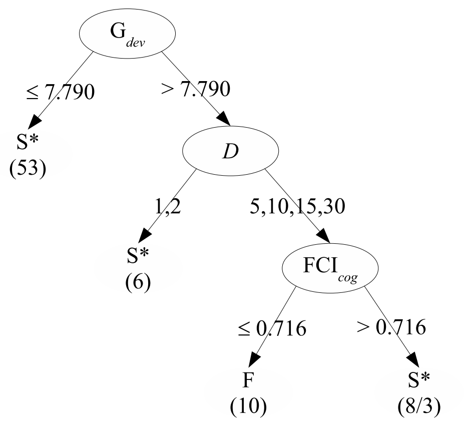
The full details of the study are described in (Malan and Engelbrecht, 2014b). 24 benchmark functions covering a range of different characteristics were used at dimensions $D = 1, 2, 5, 10, 15, 30$. Ten fitness landscape measures were calculated, namely four of the searchability measures proposed in this paper (FDC_s, IL_{ns}, FCI_{cog}, FCI_{soc}) and six other measures of micro ruggedness, macro ruggedness, funnels, average gradient, gradient deviation and fitness cloud index mean standard deviation (called $FEM_{0.01}$, $FEM_{0.1}$, DM, G_{avg}, G_{dev} and $FCI_{\overline{\sigma}}$, respectively). To determine the actual difficulty of the problems, each problem instance was solved using seven PSO variations. Each problem instance was classified as F (if the problem was not solved) or S* (indicating one of the other performance classes as described in Section 4.5). The full dataset was then divided into a training set (2/3 of the patterns) and a testing set (1/3 of the patterns). Using the training set, the C4.5 decision tree induction algorithm (Quinlan, 1993) was used to derive classification models for the different PSO variants. Three of the resulting failure prediction models are illustrated in Figure 7: traditional gbest PSO, local best PSO (Eberhart and Kennedy, 1995) and modified barebones PSO (Kennedy, 2003). In each case, the training set was used to generate the tree after which the model was tested for accuracy using the testing set. The training and testing accuracies are reported in the captions of the figures. The details of the training data classification are shown on the trees, with the total number of instances that reached each leaf node indicated in parentheses below the node. The number of instances that



(a) Traditional gbest PSO. Training accuracy: 93.5%, testing accuracy: 92.3%.



(b) Modified barebones PSO. Training accuracy: 96.1%, testing accuracy: 94.9%.



(c) Local best PSO. Training accuracy: 96.1%, testing accuracy: 92.3%.

Fig. 7 Failure prediction models for three PSO variants.

Table 6 Confusion matrices with respect to the training data for the three PSO failure prediction models illustrated in Figure 7.

Traditional gbest

Predicted class		
S*	F	
32	1	S*
2	4	F

Modified barebones

Predicted class		
S*	F	
32	0	S*
3	4	F

Local best

Predicted class		
S*	F	
34	1	S*
1	3	F

were incorrectly classified by the node, if any, are indicated after a slash in the parentheses. The confusion matrices with regard to the testing data are given in Table 6.

The training and testing accuracies achieved by the models show that it was possible to predict PSO failure based on fitness landscape features with a fairly high degree of accuracy for the benchmark problems considered. The resulting prediction models show that different fitness landscape metrics feature in the tree models of the different algorithms. This supports the idea that a single feature cannot be used to predict problem difficulty. For traditional gbest PSO, the dimension with the gradient deviation metric were the most signficant features for classifying failure or success. For modified barebones PSO, the FCI_{cog} metric with micro-ruggedness were the most significant, while for local best PSO, the gradient deviation metric with dimension and FCI_{cog} were the most significant features.

To illustrate how decision tree models can lead to further understanding, consider the modified barebones classifier in Figure 7(c). The following rule can be deduced from the tree: modified barebones PSO is predicted to fail if $\text{FCI}_{cog} \leq 0.711$ and $FEM_{0.01} > 0.537$. In fuzzy terms, this can be re-expressed as: modified barebones PSO is predicted to fail if many cognitive updates result in a deterioration in objective value and micro ruggedness is fairly high. Both the FCI_{cog} and micro ruggedness can be seen as measures more focussed on the measurement of local neighbourhood. Therefore, modified barebones PSO is not suited to problems where local neighbourbood information is misleading.

6 Conclusion

This article investigated a number of measures of searchability for continuous optimisation problems. Two general measures of searchability were formulated as adaptations of previously proposed measures of difficulty (fitness distance correlation and information landscape hardness measure). In addition, six measures were derived from fitness clouds based on PSO updates. All measures were evaluated on simple one dimensional functions to see if results were consistent with a visual inspection of the functions. Results of the negative slope coefficient based on PSO updates were not consistent with expected results and this measure was abandoned. The remaining searchability measures were tested on higher dimensional problems and all measures showed some correlation to the performance of a traditional gbest PSO algorithm.

Solving the algorithm selection problem requires the existence of multiple features for suitably characterising problems. The techniques proposed in this article contribute a number of practical measures that can be applied to continuous optimisation problems. These can be combined with other techniques for extracting problem features such as ruggedness, presence of funnels and gradients to form a multi-featured problem characteriser. Such a characteriser can be used to gain insight into new optimisation problems to be solved and in future guide the choice of the most appropriate algorithm to solve a given problem.

References

Altenberg, L. (1994). The evolution of evolvability in genetic programming. In: Kinnear K (Ed) *Advances in Genetic Programming*, MIT Press, Cambridge, MA, USA, 47–74.

Altenberg, L. (1997). Fitness distance correlation analysis: An instructive counterexample. In: Baeck, T. (Ed) *Proceedings of the Seventh International Conference on Genetic Algorithms* (pp. 57–64). Morgan Kaufmann, San Francisco, CA, USA.

BBOB Black-Box Optimization Benchmarking: Comparing Continuous Optimisers. URL http://coco.gforge.inria.fr/.

Bischl, B., Mersmann, O., Trautmann, H., & Preuss, M. (2012). Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In: Soule, T. (Ed) *Proceedings of the Fourteenth International Genetic and Evolutionary Computation Conference* (pp. 313-320), ACM, New York, NY, USA.

Borenstein, Y., & Poli, R. (2005a). Information landscapes. In: *Proceedings of the 2005 Genetic and Evolutionary Computation Conference* (pp. 1515–1522), ACM, New York, NY, USA.

Borenstein, Y., & Poli, R. (2005b). Information landscapes and problem hardness. In: *Proceedings of the 2005 Genetic and Evolutionary Computation Conference* (pp. 1425–1431), ACM, New York, NY, USA.

Cleghorn, C.W., & Engelbrecht, A.P. (2014). A generalized theoretical deterministic particle swarm model. *Swarm Intelligence*, 8(1), 35–59.

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micromachine and Human Science* (pp. 39–43), IEEE, Piscataway, NJ, USA.

Eberhart, R., & Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In: *Proceedings of the IEEE Congress on Evolutionary Computation* (Vol 1, pp. 84–88), IEEE, Piscataway, NJ, USA.

Engelbrecht, A.P. (2012). Particle swarm optimization: Velocity initialization. In: *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 1–8), IEEE, Piscataway, NJ, USA.

Helwig, S., & Wanka, R. (2008). Theoretical analysis of initial particle swarm behavior. In: Rudolph, G., Jansen, T., Lucas, S.M., Poloni, C., Beume, N. (Eds) *Proceedings of the Tenth International Conference on Parallel Problem Solving from Nature*, Lecture Notes in Computer Science (Vol 5199, pp. 889–898), Springer, Berlin, Heidelberg, Germany.

Jansen, T. (2001). On classifications of fitness functions. In: Kallel L, Naudts B, Rogers A (Eds), *Theoretical Aspects of Evolutionary Computing* (pp. 371–385), Springer, Berlin, Heidelberg, Germany.

Jones, T., & Forrest, S. (1995). Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: *Proceedings of the Sixth International Conference on Genetic Algorithms* (pp. 184–192), Morgan Kaufmann, San Francisco, CA, USA.

Kennedy, J. (1997). The particle swarm: Social adaptation of knowledge. In: *Proceedings of the IEEE International Conference on Evolutionary Computation* (pp. 303–308), IEEE, Piscataway, NJ, USA.

Kennedy, J. (2003). Bare bones particle swarms. In: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium* (pp. 80–87), IEEE, Piscataway, NJ, USA.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In: *Proceedings of the IEEE International Joint Conference on Neural Networks* (pp. 1942–1948), IEEE, Piscataway, NJ, USA.

Langdon, W.B., & Poli, R. (2007). Evolving problems to learn about particle swarm optimizers and other search algorithms. *IEEE Transactions on Evolutionary Computation* 11(5), 561–578.

Lu, G., Li, J., & Yao, X. (2011). Fitness-probability cloud and a measure of problem hardness for evolutionary algorithms. In: Merz P, Hao J-K (Eds), *Evolutionary Computation in Combinatorial Optimization*, Lecture Notes in Computer

Science (Vol 6622, pp 108–117), Springer, Berlin, Heidelberg, Germany.
Malan, K.M. (2014). Characterising Continuous Optimisation Problems for Particle Swarm Optimisation Performance Prediction. PhD thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa.
Malan, K.M., & Engelbrecht, A.P. (2013a). Ruggedness, funnels and gradients in fitness landscapes and the effect on PSO performance. In: *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 963–970), IEEE, Piscataway, NJ, USA.
Malan, K.M., & Engelbrecht, A.P. (2013b). Steep gradients as a predictor of PSO failure. In: *Proceedings of the Fifteenth International Conference on Genetic and Evolutionary Computation Conference*, Companion (pp. 9–10), ACM, New York, NY, USA.
Malan, K.M., & Engelbrecht, A.P. (2013c). A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences* 241, 148–163.
Malan, K.M., & Engelbrecht, A.P. (2014a). Fitness landscape analysis for metaheuristic performance prediction. In: Richter H, Engelbrecht AP (Eds) *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation (Vol 6, pp. 103–132), Springer, Berlin, Heidelberg, Germany.
Malan, K.M., & Engelbrecht, A.P. (2014b). Particle swarm optimisation failure prediction based on fitness landscape characteristics. In: *Proceedings of IEEE Symposium on Swarm Intelligence* (To appear), IEEE, Piscataway, NJ, USA.
Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., & Rudolph, G. (2011). Exploratory landscape analysis. In: *Proceedings of the Thirteenth Annual Conference on Genetic and Evolutionary Computation* (pp. 829–836), ACM, New York, NY, USA.
Müller, C.L., & Sbalzarini, I.F. (2011). Global characterization of the CEC 2005 fitness landscapes using fitness-distance analysis. In: Di Chio et al (Eds), *Applications of Evolutionary Computation*, Lecture Notes in Computer Science (Vol 6624, pp. 294–303), Springer, Berlin, Heidelberg, Germany.
Muñoz, M.A., Kirley, M., & Halgamuge, S.K. (2012). A meta-learning prediction model of algorithm performance for continuous optimization problems. In: Coello Coello, C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (Eds) *Proceedings of the Twelfth International Conference on Parallel Problem Solving from Nature – Part I* (pp. 226–235), Springer, Berlin, Heidelberg, Germany.
Naudts, B., & Kallel, L. (1998). Some Facts About So Called GA-Hardness Measures. Tech. Rep. 379, CMAP, Ecole Polytechnique, France.
Naudts, B., & Kallel, L. (2000). A comparison of predictive measures of problem difficulty in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 4(1), 1–15.
Quick, R.J., Rayward-Smith, V.J., & Smith, G.D. (1998). Fitness distance correlation and ridge functions. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (Eds) *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature* (pp. 77–86), Springer, Berlin, Heidelberg, Germany.
Quinlan, J.R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco, CA, USA.
Reeves, C.R. (1999). Predictive measures for problem difficulty. In: *Proceedings of the 1999 Congress on Evolutionary Computation* (pp. 736–743), IEEE, Piscat-

away, NJ, USA.
Rice, J.R. (1976). The algorithm selection problem. *Advances in Computers* 15, 65–118.
Shang, Y.W., & Qiu, Y.H. (2006). A note on the extended Rosenbrock function. *Evolutionary Computation* 14(1), 119–126.
Smith, T., Husbands, P., Layzell, P., & O'Shea, M. (2002). Fitness landscapes and evolvability. *Evolutionary Computation* 10(1), 1–34.
Smith-Miles, K. (2008). Towards insightful algorithm selection for optimisation using meta-learning concepts. In: *Proceedings of the IEEE Joint Conference on Neural Networks* (pp. 4118–4124), IEEE, Piscataway, NJ, USA.
Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., & Tiwari, S. (2005). Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Tech. rep., Nanyang Technological University, Singapore.
Trelea, I.C. (2003). The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters* 85(6), 317–325.
Turney, P.D. (1999). Increasing evolvability considered as a large-scale trend in evolution. In: *Proceedings of 1999 Genetic and Evolutionary Computation Conference* Workshop Program (pp. 43–46), Morgan Kaufmann, San Mateo, CA, USA.
Van Den Bergh, F., & Engelbrecht, A.P. (2006). A study of particle swarm optimization particle trajectories. *Information Sciences* 176(8), 937–971.
Vanneschi, L. (2004). Theory and Practice for Efficient Genetic Programming. Phd thesis, Faculty of Sciences, University of Lausanne, Switzerland.
Vanneschi, L., Clergue, M., Collard, P., Tomassini, M., & Verel, S. (2004). Fitness clouds and problem hardness in genetic programming. In: Deb, K., Poli, R., Banzhaf, W., Beyer, H.-G., Burke, E., Darwen, P., Dasgupta, D., Floreano, D., Foster, J., Harman, M., Holland, O., Lanzi, P.L., Spector, L., Tettamanzi, A.G.B., Thierens, D., Tyrrell, A. (Eds) *Proceedings of Genetic and Evolutionary Computation Conference*, Lecture Notes in Computer Science (Vol 3103, pp. 690–701), Springer, Berlin, Heidelberg, Germany.
Verel, S., Collard, P., & Clergue, M. (2003). Where are bottlenecks in NK fitness landscapes? In: *Proceedings of the 2003 Congress on Evolutionary Computation* (Vol 1, pp. 273–280), ACM, New York, NY, USA.