Quantifying Ruggedness of Continuous Landscapes using Entropy

Katherine M. Malan, *Member, IEEE* Andries P. Engelbrecht, *Senior Member, IEEE*

***Abstract*— A major unsolved problem in the field of optimisation and computational intelligence is how to determine which algorithms are best suited to solving which problems. This research aims to analytically characterise individual problems as a first step towards attempting to link problem types with the algorithms best suited to solving them. In particular, an information theoretic technique for analysing the ruggedness of a fitness landscape with respect to neutrality was adapted to work in continuous landscapes and to output a single measure of ruggedness. Experiments run on test functions with increasing ruggedness show that the proposed measure of ruggedness produced relative values consistent with a visual inspection of the problem landscapes. Combined with other measures of complexity, the proposed ruggedness measure could be used to more broadly characterise the complexity of fitness landscapes in continuous domains.**

I. Introduction

Extensive work and research has been conducted in a relatively short period of time in the field of population-based optimisation techniques. Although these techniques have been used successfully to solve many different optimization problems, there is no clear understanding of which approaches or which variations on which algorithms are in general more suitable for which kinds of problems. It is also true that no one of the algorithms is at all times superior to the other [1].

Some have tried to predict which problems will be 'hard' for population-based optimisation algorithms to solve [2], [3], [4], [5], [6]. Other studies have, however, shown that these measures are not useful for predicting problem hardness [7], [8], [9] and that no satisfactory problem difficulty measure for search heuristics has been found [10], [11]. Instead of trying to determine whether a particular optimisation problem would be hard to solve using a population-based algorithm or not, 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.

There are a number of existing techniques for analysing fitness landscapes. Some are descriptive (for example, whether a function is GA-deceptive or not [4]), while others are analytical, resulting in an attribute (usually a number) as output to a fitness function [10]. Examples of analytical techniques include measures of correlation between fitness values [12], [13], and measures of hardness based on the amount of information present in a landscape [2], [3]. He *et al.* [11] classify difficulty measures as exact or approximate, depending on whether a theoretical (deterministic algorithm) or empirical approach (randomized algorithm) is used. They also distinguish between predictive (where an algorithm's worst-case running time is bounded by a polynomial in n) and non-predictive realizations. It is important to note that a perfect classification scheme of fitness functions that is also efficient, cannot exist [10], [11].

Some of the problems associated with techniques for classifying problems include the following:

1) many techniques assume a particular algorithm (e.g. measures of GA-hardness), so they would not apply to problems in general;
2) many techniques are defined based on the assumption that the problem is encoded using a discrete representation and are not defined for real-encoded problems;
3) most techniques are descriptive classifications [10], so they do not result in an attribute or number as output (for example, labelling a problem as hard or not hard does not help in relative comparisons between problems).

There is no doubt that the structure of a fitness landscape affects an algorithm's ability to search that space. It is also clear that investigating a single characteristic of a fitness landscape (such as modality) is not sufficient for determining whether the problem will be difficult for a given search algorithm or not. For example, Horn and Goldberg [14] show that there are problems with maximum modality (such as their one-max function with "bumps") that are easy for a GA to optimise and there are problems with minimal modality (such as long path problems) that are hard for a GA to optimise. Although Rudolph [15] later showed that Horn and Goldberg's unimodal binary long-path problems can be solved in polynomial expected time using mutation and elitist selection, Droste *et al.* [16] later proved that the class of unimodal functions is in general difficult in black-box optimisation.

In an attempt to analyse optimisation problems, what seems to be needed is a combined approach, where a whole host of characteristics is analysed together to more broadly characterise a problem. The aim of this research is to develop approximate techniques for quantifying characteristics of real-encoded problems. This paper focuses in particular on a technique for quantifying the ruggedness of fitness landscapes using entropy. It is hoped that by developing a suite of such techniques, a way of characterising problems can be developed that will help in identifying links between problem types and the algorithms that would be best suited to solving

Katherine M. Malan and Andries P. Engelbrecht are with the Department of Computer Science, University of Pretoria, South Africa (email: kmalan@cs.up.ac.za).

978-1-4244-2959-2/09/$25.00 ©2009 IEEE

them.

In Section II a proposed approach to analysing the ruggedness of discrete landscapes with respect to neutrality (as proposed by Vassilev *et al.* [17], [18], [19]) is described with slight modifications to work with real-encoded problems. Section III then describes how this technique was tested on a set of benchmark functions with increasing ruggedness. Finally, a proposed single measure for ruggedness is presented with results in Section IV.

II. Analysing ruggedness using entropy

Ruggedness refers to the number and distribution of local optima. Entropy is a measure of the uncertainty involved in sampling from a source [20]. Given a sample of fitness values resulting from a random walk on a problem landscape, and using a suitable encoding, an estimate of entropy of this sample could form the basis for an estimate of the ruggedness of the problem landscape. Vassilev *et al.* [17], [18], [19] propose such an information theoretic technique for studying the structure of discrete landscapes in terms of their smoothness, ruggedness and neutrality. This paper is limited to discussing their approach to estimating ruggedness (and not smoothness) with respect to neutrality.

The basic idea is to obtain a landscape path by performing a random walk on the landscape. This path is represented as an ensemble of three-point objects, where each object is a point on the path together with its neighbours. Each three-point object is classified as one of the following:

- neutral (a point together with its neighbours has equal fitness values);
- smooth (the fitness differences between the three points changes in one direction: a slope);
- rugged (the fitness differences between the three points changes in two directions: a peak, valley or step).

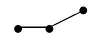
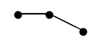
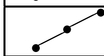
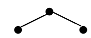
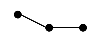
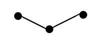
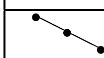
Table I gives all possible shapes of three-point objects with their classification. The encoding is discussed in Section II-A. Deciding whether a given three-point object is neutral, smooth or rugged would depend on the margin of error used to determine whether two values are 'equal' or not. By increasing this margin of error, a larger number of objects (such as very small steps, or very shallow valleys) could be regarded as neutral.

Given a sample of three-point objects obtained from a random walk on the problem landscape and a classification as described in Table I, an information function is used to estimate the entropy of the subset of objects that are rugged. By increasing the margin of error used to determine whether two fitness values are 'the same' or not, the information function is used to reveal how the measure of entropy changes as the neutrality of the landscape is increased. The following section provides a formalisation of this approach.

A. Formalisation of approach

This section describes Vassilev *et al.*'s [19] approach to estimating ruggedness with respect to neutrality. Assume that a random walk on a landscape generates the time series of

TABLE I
Classification and encoding of three-point objects (a point on a path together with its neighbours) as neutral, rugged or smooth based on relative fitness values.

Object shape	Classification	Encoding
	neutral	0 0
	rugged	0 1
	rugged	0 $\bar{1}$
	rugged	1 0
	smooth	1 1
	rugged	1 $\bar{1}$
	rugged	$\bar{1}$ 0
	rugged	$\bar{1}$ 1
	smooth	$\bar{1}$ $\bar{1}$

fitness values $\{f_t\}_{t=0}^{n}$. This time series is represented as a string, $S(\varepsilon) = s_1 s_2 s_3 ... s_n$, of symbols $s_i \in \{\bar{1}, 0, 1\}$, obtained by the function:

$$\begin{aligned} s_i = \Psi_{f_t}(i, \varepsilon) &= \bar{1}, \text{ if } f_i - f_{i-1} < -\varepsilon \\ &= 0, \text{ if } |f_i - f_{i-1}| \leq \varepsilon \\ &= 1, \text{ if } f_i - f_{i-1} > \varepsilon \end{aligned} \quad (1)$$

The parameter ε is a real number that determines the accuracy of the calculation of string $S(\varepsilon)$. A low value for ε would result in a high sensitivity to the differences between neighbouring fitness values.

Based on this definition of the string $S(\varepsilon)$, an entropic measure $H(\varepsilon)$ is defined as follows:

$$H(\varepsilon) = -\sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]} \quad (2)$$

where p and q are elements from the set $\{\bar{1}, 0, 1\}$, $P_{[pq]}$ is defined as

$$P_{[pq]} = \frac{n_{[pq]}}{n} \quad (3)$$

and $n_{[pq]}$ is the number of sub-blocks pq in the string $S(\varepsilon)$.

The formula for $H(\varepsilon)$ (equation 2) is an instance of the more general b-ary entropy [20]. The result is a value in the range $[0, 1]$ and is an estimate of the variety of 'shapes' in the walk [17]. Note that equation 2 only considers the subset of elements that are rugged (sub-blocks pq in the string $S(\varepsilon)$ where $p \neq q$). The base of the logarithmic function is 6, because there are 6 possible rugged shapes (see Table I). For each rugged element, equation 3 calculates the probability of that element occurring. Therefore, the higher the value of $H(\varepsilon)$, the more the variety of rugged shapes in the walk and so the more rugged the landscape.

The parameter ε determines how sensitive equation Ψ_{f_t} (equation 1) is to differences in fitness values. By increasing the value of ε the neutrality of the landscape is increased. The smallest value of ε for which the landscape becomes flat (i.e. for which $S(\varepsilon)$ is a string of 0's) is called the information stability and is denoted by ε^*. The value of ε^* would therefore equate to the largest difference in fitness values between any two successive points on the path.

Unlike other statistical approaches to measuring ruggedness (such as correlation functions [12], [13]), $H(\varepsilon)$ is not a measure of the ruggedness of the landscape, but rather an indication of the relationship between the ruggedness and neutrality.

B. Random walk algorithm

The calculation of entropic measure $H(\varepsilon)$ is based on a random walk through the landscape, resulting in a sequence of fitness values. In the case of binary landscapes, a random walk could be implemented as follows [19]: start from a randomly chosen landscape point, generate all neighbours of the current point by mutation (bit flip), choose randomly one neighbour and record its fitness, generate all neighbours of the new point, which becomes the current point, and so on.

In the case of continuous landscapes, unlike in the case of discrete landscapes, there is no set of all possible neighbours of any point. An algorithm for a walk through continuous landscapes was needed. The purpose was to define a walk through the landscape that in some way resembles a search path through the problem space by an individual in a population-based algorithm. The approach used was to implement a *random increasing walk* algorithm, where each successive step has a larger **x**-value (where **x** is the vector defining the problem space), but the increase is of a random size (within a maximum step size). The *random increasing walk* algorithm is given in Figure 1.

C. Modification to the definition of $S(\varepsilon)$

The *random increasing walk* algorithm defined in Section II-B, generates a time series of fitness values $\{f_t\}_{t=0}^n$. This time series can be represented as a string $S(\varepsilon)$ of symbols taken from $\{\bar{1}, 0, 1\}$. In Vassilev *et al.*'s definition [17], the string $S(\varepsilon)$ is considered with periodic boundary conditions (based on the assumption that the landscape is statistically isotropic). This means that a walk of n steps results in a string $S(\varepsilon)$ of n symbols, because the last fitness value in the walk is compared back with the first fitness value of the walk to determine the final symbol of $S(\varepsilon)$.

For continuous landscapes, the definition of $S(\varepsilon)$ was modified to not 'wrap around' in this way. A walk of n steps would therefore result in a string $S(\varepsilon)$ of $n-1$ symbols.

III. A VISUAL ANALYSIS OF THE TECHNIQUE

To determine whether the above approach could be used as a basis for obtaining a single measure of ruggedness, a visual analysis of the behaviour of $H(\varepsilon)$ (equation 2) was performed on a number of benchmark functions.

A. Benchmark functions

Seven benchmark functions were selected with different levels of ruggedness. These functions are specified in Table II and one-dimensional graphs of these functions are shown in Figure 2 (note that the Step function is plotted for the domain [-20,20] to visualise the steps). Based on a visual inspection, the following observations are made:

- The functions are displayed with increasing levels of ruggedness. A measure of ruggedness should therefore rank these functions in the same order in which they are displayed in Figure 2.
- The Table function is highly neutral, whereas the Step function has both high neutrality (due to the many short flat sections) and high ruggedness (due to the many steps in the fitness landscape).
- Despite occupying very different domains, the Straight function and the Half Spherical function have similar shapes and should therefore have similar low levels of ruggedness.
- The Rastrigin function and Step function display the same macro shape and are both very rugged, although in different ways.

TABLE II
BENCHMARK FUNCTIONS (D IS THE DIMENSION OF THE PROBLEM)

Table	$f(x) = 0, \quad x \in [-5, 0.5)$ $= 1, \quad x \in [-0.5, 0.5]$ $= 0, \quad x \in (0.5, 5]$
Straight	$f(\mathbf{x}) = \sum_{i=1}^{D} x_i,$ $x_i \in [-5, 5]$
Half Spherical	$f(\mathbf{x}) = \sum_{i=1}^{D} x_i^2,$ $x_i \in [0, 100]$
Spherical	$f(\mathbf{x}) = \sum_{i=1}^{D} x_i^2,$ $x_i \in [-100, 100]$
Rastrigin	$f(\mathbf{x}) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10],$ $x_i \in [-5.12, 5.12]$
Step	$f(\mathbf{x}) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2,$ $x_i \in [-100, 100]$
Ackley	$f(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)$ $-exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e,$ $x_i \in [-32, 32]$

B. Testing approach

Using the seven benchmark functions the following was done:

- An initial *random increasing walk* of 10,000 steps was performed on each benchmark function. This walk was used to calculate the information stability measure (ε^*) for each benchmark. Recall that ε^* is the smallest value of ε for which the landscape (as represented by the walk) becomes flat. For each function this value of ε^*

Input data:

- the number of dimensions of the problem
- the domain of the problem
- the number of steps required in the walk (if not specified, the number of steps is set to 100)
- the step size (if not specified, the step size is set = (range of the problem domain)/100).

Create an array of vectors for storing the walk (called *walk*)
  Initialise *count* = 0 (for counting the number of steps in the walk)
  Set *walk*[0] to a random position within the bounds of the problem
  while *count* < *number of steps required* **do**
    for every dimension *i* of the problem
      Generate a random number (*step*) in the range [0,*stepSize*]
      Set $walk[count]_i$ = $walk[count]_{i-1}$ + *step*
      if $walk[count]_i$ > the maximum bound of the space **then**
        Set $walk[count]_i$ = $walk[count]_i-$ (range of the problem domain)
    Set *count* = *count* + 1
  end

Fig. 1. Random Increasing Walk Algorithm

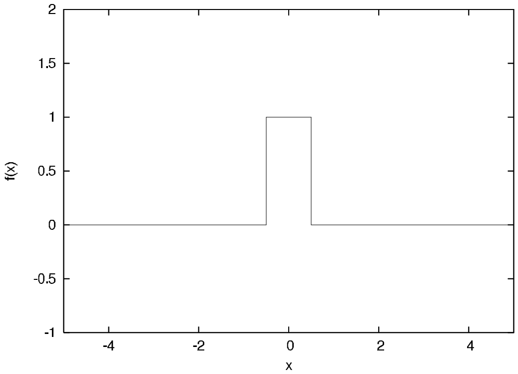
was used as the upper bound on the range of sensible values for ε.

- Thirty independent *random increasing walks* of 10,000 steps were performed on each benchmark function.
- For each walk, $H(\varepsilon)$ was calculated for increasing values of ε. The behaviour of $H(\varepsilon)$ for small values of ε can be significant for characterising the ruggedness with respect to neutrality of the landscape [19]. For this reason, a larger proportion of smaller values for ε were chosen. To compare benchmarks with widely different fitness value ranges, different values for ε were used for each benchmark function, based on the value of ε^* of the particular benchmark function. The nine values of ε were defined as: 0, $\frac{\varepsilon^*}{128}, \frac{\varepsilon^*}{64}, \frac{\varepsilon^*}{32}, \frac{\varepsilon^*}{16}, \frac{\varepsilon^*}{8}, \frac{\varepsilon^*}{4}, \frac{\varepsilon^*}{2}, \varepsilon^*$.
- For each benchmark function, the mean of $H(\varepsilon)$ ($\overline{H}(\varepsilon)$) over the 30 walks was calculated for each of the nine values for ε.

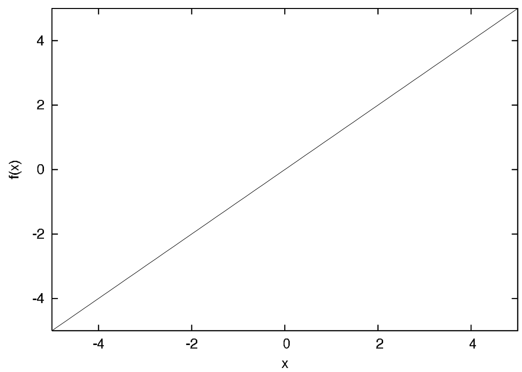
C. Results and discussion

A graph showing $\overline{H}(\varepsilon)$ for different values of ε is shown in Figure 3. This graph illustrates the trend of how ruggedness changes with respect to neutrality. The use of relative ε-values that depend on the value of ε^* for each function allows for easy comparisons between benchmarks occupying very different fitness ranges. The following observations can be made on the behaviour of $H(\varepsilon)$:

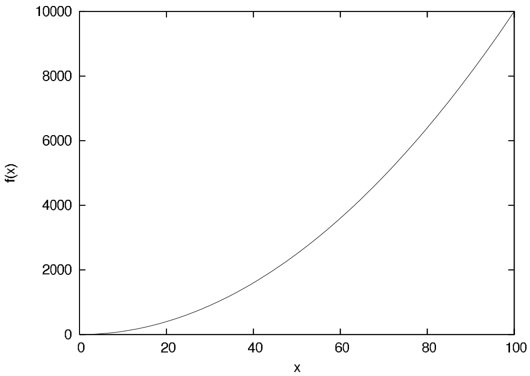
- The value ε^* is defined as the smallest value of ε for which the landscape becomes flat. This is when $S(\varepsilon)$ becomes a string of 0's. As can be seen on the graph in Figure 3, the value of $\overline{H}(\varepsilon^*)$ for all functions is 0, which is as expected.
- In the case of all benchmark functions, except for Table and Step, $\overline{H}(\varepsilon)$ is an increasing function for small values of ε. The least rugged and most neutral of the functions is the Table function, which has a low value for $\overline{H}(0)$ and maintains this value until ε is greater than $\frac{\varepsilon^*}{2}$. The only other function with high neutrality is the Step function, which has its highest value for $\overline{H}(\varepsilon)$ where $\varepsilon = 0$. Figure 4 illustrates, using a very simple example, how the value of $H(\varepsilon)$ can increase with an increase in ε. Assume a straight walk on function $f(x) = x^2$ produces the time series of fitness values $\{f_0 = 0, f_1 = 1, f_2 = 4, f_3 = 9\}$. The value of $S(\varepsilon = 0)$ is 1 1 1 (a perfectly smooth landscape), since the fitness values are all increasing. This results in a value for $H(0)$ of 0. When ε increases to 2, $S(\varepsilon = 2)$ changes to 0 1 1, since the difference between f_1 and f_0 is less than ε, so it is regarded as no change. The increase in neutrality (by increasing ε) has resulted in the introduction of a rugged element (0 1) in the previously perfectly smooth landscape represented by $S(\varepsilon = 0)$ and hence the value of $H(\varepsilon)$ increases. It is therefore only in the case of functions with low neutrality that $H(\varepsilon)$ is an increasing function for small values of ε. If a function has low neutrality, increasing the neutrality (i.e. increasing the value of ε) introduces more rugged elements (steps) which were present in low proportions (or even absent) for lower values of ε.
- In the case of all functions with low neutrality the value of $\overline{H}(\varepsilon)$ peaks at a particular value for ε and then decreases after that. To understand this behaviour, consider the Straight function: For small values of ε the path represented by $S(\varepsilon)$ is smooth (except for the cases where the *random increasing walk* reaches the maximum bound of the space and wraps around to the beginning). With larger values of ε the magnifying glass with which the discrete path through the landscape is investigated zooms out. Some successive points which previously were regarded as having different fitness values are now regarded as having the same fitness values. This results in the introduction of rugged elements (in the form of step shapes) into the analysis



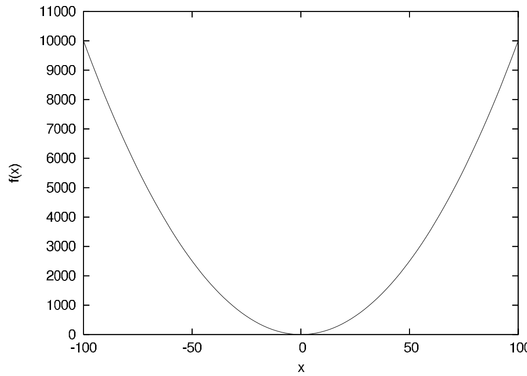
(a) Table function



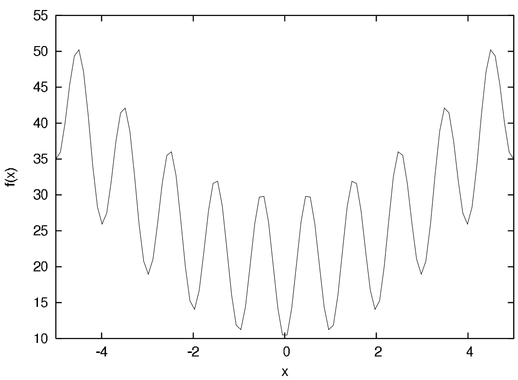
(b) Straight function



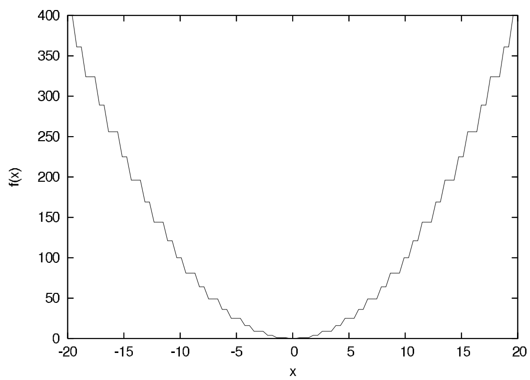
(c) Half Spherical function



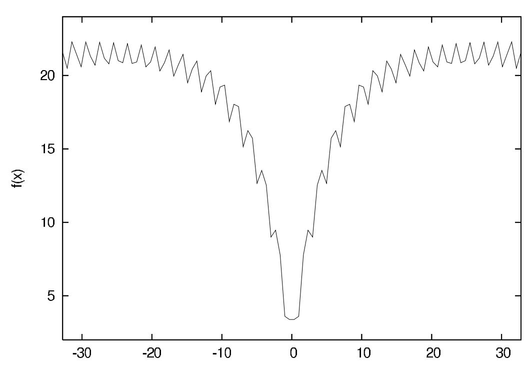
(d) Spherical function



(e) Rastrigin function



(f) Step function



(g) Ackley function

Fig. 2. One-dimensional functions of increasing ruggedness.

with a corresponding increase in the value of $H(\varepsilon)$. For greater values of ε, more and more neutral shapes are introduced until the path becomes perfectly neutral.

IV. Proposed Single Ruggedness Measure

A future aim of this research is to analyse a host of characteristics (including ruggedness) together to broadly characterise a problem. If the characterisation of the landscape as illustrated in Figure 3 could be reduced to a single scalar value for characterising ruggedness, this would facilitate the future aim of combining different characteristics of problems to be analysed further. On inspection of the behaviour of $H(\varepsilon)$ for different values of ε, as shown in Figure 3, it would seem that a significant value for each function is the maximum value of $H(\varepsilon)$. The point at which the maximum of $H(\varepsilon)$ occurs would correspond to the level of magnification that produces the most 'difference' in the landscape in the form of the maximum number of rugged elements. To characterise the ruggedness of a function, the following single value measure is proposed for a fitness function f:

$$R_f = \max_{\forall \varepsilon \in [0, \varepsilon^*]} \{H(\varepsilon)\}$$

To approximate the theoretical value of R_f, the maximum of $\overline{H}(\varepsilon)$ was calculated for all ε values in the set $\{0, \frac{\varepsilon^*}{128}, \frac{\varepsilon^*}{64}, \frac{\varepsilon^*}{32}, \frac{\varepsilon^*}{16}, \frac{\varepsilon^*}{8}, \frac{\varepsilon^*}{4}, \frac{\varepsilon^*}{2}, \varepsilon^*\}$. The values of this approximation of R_f for the seven one-dimensional benchmark functions are shown in Table III. The ranking of these functions from lowest to highest based on the approximate value of R_f shows that this ordering corresponds to the functions as displayed in Figure 2, which were ranked based on visual ruggedness. The ruggedness measure for the Straight and Half Spherical functions are very close in value despite occupying very different domains, which is as expected. The approximate values of R_f for Rastrigin(1D) and Step(1D) are very close in value, which was expected due to their similar macro shapes.

To determine whether the R_f measure scales well for increased dimensions, the values were calculated for five of the benchmarks to 30 dimensions. Note that the Random Increasing Walk as described in Figure 1 is defined for multiple dimensions. The results for the Half Spherical, Spherical, Rastrigin, Step and Ackley benchmarks in 30 dimensions is shown in Table IV. Although the approximate values for ruggedness measure R_f changed in some cases, the ordering of the 30-dimensional functions based on R_f corresponds to the ordering of the one-dimensional functions. This seems to indicate that the ruggedness measure scales well to increasing dimensions.

V. Conclusions

Despite extensive work being done in trying to predict which problems will be hard for population-based algorithms, no satisfactory problem difficulty measure has been found. Although certain characteristics (such as the level of modality of the problem landscape) can affect problem difficulty, no characteristic on its own is sufficient as a predictor of problem difficulty. Rather than trying to predict problem hardness, this research aims to analytically characterise problems as a first step towards attempting to link problem types with the algorithms best suited to solving them.

An existing information theoretic technique for investigating the ruggedness with respect to neutrality of discrete problem landscapes was modified to work in continuous domains. In addition, a single measure for quantifying the ruggedness of a function is proposed. Preliminary tests on benchmark functions show that the proposed measure behaves as predicted and scales well to higher dimensions. Further work will include testing the technique for more problems and contrasting this measure against other ways of quantifying ruggedness (such as correlation measures), adapted to work in continuous domains.

TABLE III

Approximate values of ruggedness measure R_f for a range of 1-dimensional functions

Function f	R_f
Table function	0.06
Straight	0.27
Half Spherical (1D)	0.29
Spherical (1D)	0.45
Rastrigin (1D)	0.53
Step (1D)	0.54
Ackley (1D)	0.88

TABLE IV

Values of ruggedness measure R_f for 30-dimensional functions

Function f	R_f
Half Spherical (30D)	0.28
Spherical (30D)	0.52
Rastrigin (30D)	0.59
Step (30D)	0.64
Ackley (30D)	0.87

Acknowledgment

This work was supported in part by the National Research Foundation.

References

[1] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, April 1997.

[2] Y. Borenstein and R. Poli, "Information landscapes," in *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*. New York, NY, USA: ACM Press, 2005, pp. 1515–1522.

[3] ——, "Information landscapes and problem hardness," in *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*. New York, NY, USA: ACM Press, 2005, pp. 1425–1431.

[4] K. Deb and D. E. Goldberg, "Sufficient Conditions for Deceptive and Easy Binary Functions," *Annals of Mathematics and Artificial Intelligence*, vol. 10, no. 4, pp. 385–408, December 1994.

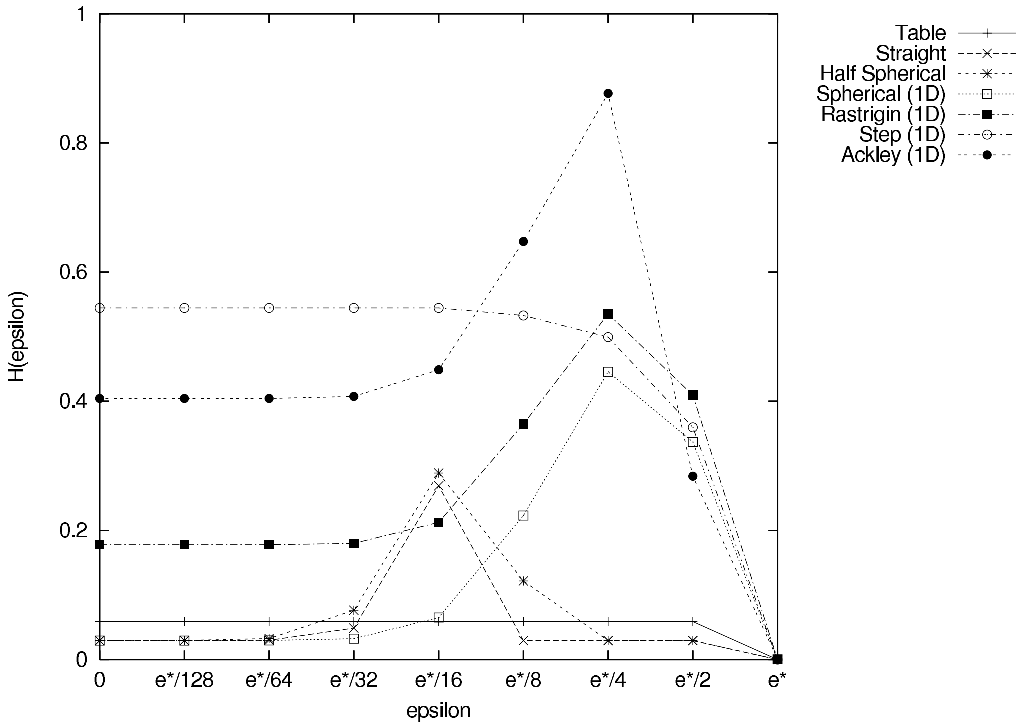


Fig. 3. $\overline{H}(\varepsilon)$ over different values of ε for seven different 1-dimensional functions, where the values of ε depend on ε^*
.

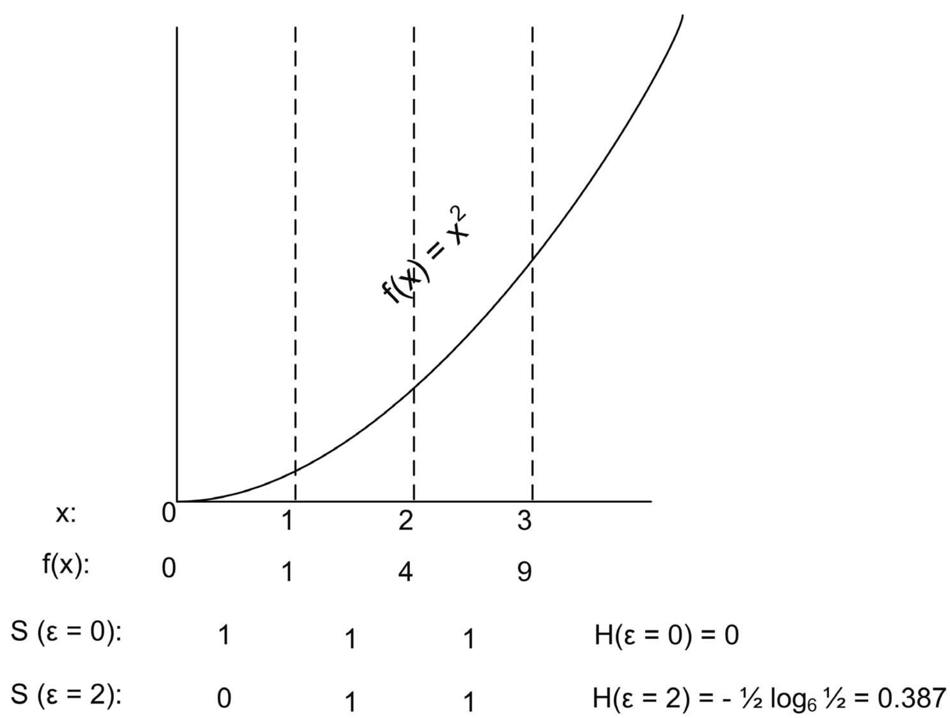


Fig. 4. A simple illustration of how the value of $H(\varepsilon)$ can increase with an increase in ε

[5] D. E. Goldberg, J. Horn, and K. Deb, "What Makes a Problem Hard for a Classifier System?" Department of General Engineering, 117 Transportation Building, 104 South Mathews Avenue, Urbana, IL 61801-2996, Tech. Rep. IlliGAL Report No 92007, 1992. [Online]. Available: citeseer.ist.psu.edu/goldberg92what.html

[6] D. E. Goldberg, "Making Genetic Algorithms Fly: A Lesson from the Wright Brothers," *Advanced Technology for Developers*, vol. 2, pp. 1–8, Feb. 1993.

[7] J. J. Grefenstette, "Deception considered harmful," in *Foundations of Genetic Algorithms 2*, D. L. Whitley, Ed. San Mateo, CA: Morgan Kaufmann, 1993, pp. 75–91.

[8] S. Forrest and M. Mitchell, "What Makes a Problem Hard for a Genetic Algorithm? Some Anomalous Results and Their Explanation," *Machine Learning*, vol. 13, pp. 285–319, 1993.

[9] B. Naudts and L. Kallel, "A Comparison of Predictive Measures of Problem Difficulty in Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 1, p. 1, April 2000. [Online]. Available: citeseer.ist.psu.edu/naudts00comparison.html

[10] T. Jansen, "On Classifications of Fitness Functions," Fachbereich Informatik, Universität Dortmund, 44221 Dortmund, Tech. Rep. CI–76/99, 1999. [Online]. Available: citeseer.ist.psu.edu/jansen99classification.html

[11] J. He, C. Reeves, C. Witt, and X. Yao, "A Note on Problem Difficulty Measures in Black-Box Optimization: Classification, Realizations and Predictability," *Evolutionary Computation*, vol. 15, no. 4, pp. 435–443, 2007.

[12] E. Weinberger, "Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference," *Biological Cybernetics*, vol. 63, no. 5, pp. 325–336, September 1990.

[13] P. F. Stadler, "Landscapes and Their Correlation Functions," *J. Math. Chem.*, vol. 20, pp. 1–45, 1996.

[14] J. Horn and D. E. Goldberg, "Genetic Algorithm Difficulty and the Modality of Fitness Landscapes," in *Foundations of Genetic Algorithms 3*, L. D. Whitley and M. D. Vose, Eds. San Francisco, CA: Morgan Kaufmann, 1995, pp. 243–269. [Online]. Available: citeseer.ist.psu.edu/article/horn94genetic.html

[15] G. Rudolph, "How Mutation and Selection Solve Long-Path Problems in Polynomial Expected Time," *Evolutionary Computation*, vol. 4, no. 2, pp. 195–203, April 1997.

[16] S. Droste, T. Jansen, and I. Wegener, "Upper and Lower Bounds for Randomized Search Heuristics in Black-Box Optimization," *Theory of Computing Systems*, vol. 39, no. 4, pp. 525–544, 2006.

[17] V. K. Vassilev, T. C. Fogarty, and J. F. Miller, "Information Characteristics and the Structure of Landscapes," *Evolutionary Computation*, vol. 8, no. 1, pp. 31–60, 2000.

[18] V. K. Vassilev, "Fitness Landscapes and Search in the Evolutionary Design of Digital Circuits," PhD Thesis, Napier University, 2000.

[19] V. K. Vassilev, T. C. Fogarty, and 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.

[20] S. Roman, *Coding and Information Theory*. 175 Fifth Avenue, New York, USA: Springer-Verlag, 1992.