Landscape Characterization of Numerical Optimization Problems Using Biased Scattered Data

Mario A. Muñoz*, Michael Kirley† and Saman K. Halgamuge*

*Department of Mechanical Engineering, The University of Melbourne, Melbourne, Australia
†Department of Computing and Information Systems, The University of Melbourne, Melbourne, Australia

e-mail: mariom@student.unimelb.edu.au, {mkirley,saman}@unimelb.edu.au

Abstract—The characterization of optimization problems over continuous parameter spaces plays an important role in optimization. A form of “fitness landscape” analysis is often carried out to describe the problem space in terms of modality, smoothness and variable separability. The outcomes of this analysis can then be used as a measure of problem difficulty and to predict the behaviour of a given algorithm. However, the metric value estimates of the landscape characterization are dependent upon the representation scheme adopted and the sampling method used. Consequently, the development of a complete classification of problem structure and complexity has proven to be challenging. In this paper, we continue this line of research. We present a methodology for the characterization of two dimensional numerical optimization problems. In our approach, data extracted during the search process is analyzed and the dependency of the results to the nominated sampling method is corrected. We show via computational simulations that the calculated metric values using our approach are consistent with the results from random experiments. As such, this study provides a first step towards the on-line calculation of fitness landscape characterization metrics and the development of empirical performance models of search algorithms. Advances in these areas would provide answers to the algorithm selection and portfolio configuration problems.

Index Terms—Data analysis, Heuristic algorithms, Information entropy, Optimization, Search problems

I. INTRODUCTION

Search algorithms based on stochastic and heuristic rules have been used to solve a wide range of optimization problems, particularly those problems where there is neither sufficient knowledge about the objective function nor sufficient time for intense investigation of the problem [1]–[3]. However, an adequate explanation of why the algorithms were successful when tackling the given optimization problems is lacking. Subsequently, it is risky to assert that the algorithm under study would work effectively on problems beyond those in the test set. Indeed, it would be very optimistic to expect that an algorithm would work reasonably well across a wide range of problems unless some restrictions are placed [4]. Hence, it is necessary to be careful when generalizing results based on empirical tests.

Ideally, measures are required to quantify problem difficulty and to help predict whether a search algorithm might be successful on a given optimization problem. Landscape characterization refers to the development of methods, which provide metrics related to the attributes of an optimization problem such as modality, smoothness and variable separability and thus problem difficulty [5]. Landscape characterization methods have been thoroughly investigated over the past twenty years [6]–[22]. However, their use is not widespread because it is easy to find counterexamples [5]. In addition, characterization metrics typically require precise calculations using larger volumes of data [23], the results of which are dependent on the representation and the sampling method used [24]. Nevertheless, if the only accessible information available are freely chosen observations, an estimation of these metrics would be useful when characterizing the problem difficulty. We also have to acknowledge that a single measure would not be useful to detect every single attribute of the “fitness landscape”.

In this paper, we present a methodology for the characterization of two dimensional numerical optimization problems. Our methodology analyses data extracted during the search process, hence it removes the necessity for a separated sampling experiment. Importantly, our approach corrects the dependency of the results from the parameters of the sampling algorithm. The estimated metrics can be used to quantify the modality, smoothness and variable scaling of the optimization problem. We show via computational simulations that our approach produces metric values that are consistent with the results from random experiments. As such, this work represents a first step towards the on-line calculation of these metrics and the development of empirical performance models of search algorithms. Advances in these areas would provide some answers to the algorithm selection and portfolio configuration problems.

The paper is organized as follows: In Section II we describe the conceptual framework for landscape characterization and analyze the reasons that make optimization problems hard. In Section III we formulate our method for landscape characterization. In Section IV we describe the experiments carried out and the obtained results. Finally, we discuss our results and propose further work in Section V.

II. BACKGROUND

A. Black-box Numerical Optimization

Numerical optimization problems have a discrete domain of feasible solutions when solved in a digital computational machine. This means that both the search set, \( X \subset \mathbb{R}^n \), and the set of possible objective values, \( Y \subset \mathbb{R} \), although very large, are finite. Let \( f \) be a function such that:

- U.S. Government work not protected by U.S. copyright
\( f : \mathcal{X} \to \mathcal{Y} \)

Let \( x_i \in \mathcal{X} \) be a point with a function value \( y_i = f(x_i) \in \mathcal{Y} \). In this type of problem, the pairs \((x_i, y_i), i = 1, \ldots, p\) are the only accessible information and they can be chosen freely. Without losing generality, assume that the goal is to minimize \( f \). This means that we need to find one or more candidate solutions \( x_o \in \mathcal{X}, y_o = f(x_o) \), such that \( |y_o - y| \ll \delta \), where \( \delta \to 0 \) and \( x^* = f(x^*) \) are the location of the global optimum and its value, which may or may not be available for representation in a digital computer. This restriction is made because, most of the time, obtaining the global optimum is neither feasible nor essential in practice [25]. It also appears appropriate to assume that any \textit{a priori} information available about the problem has been used in defining an appropriate set of variables, feasibility ranges or representation [26]. Therefore, it is acknowledged that the expertise of the designer in correctly specifying the problem cannot be replaced but complemented.

There are many methods available to search for \( x_o \). Ideally, the method selected does not suffer of \textit{premature convergence}. This means that, after a bounded number of function evaluations, the method is not able to generate solutions outside a small area under examination and the solution obtained is unacceptable.

### B. Search Landscapes

The search landscape is a commonly used metaphor to describe how a search method solves a particular optimization problem. In a two dimensional search space, it is possible to make an analogy of the structure of the problem to a surface composed of ridges, valleys and basins, each one of them strongly related to its neighbors. In this surface the local or global optimal points are located on the highest or lowest points, respectively. In this sense the global optima are located on the highest or lowest points, respectively. Therefore, if the landscape exhibits large fluctuations between points, presenting several local optima and steep ascents and descents, it is often the result of a bad specification of the optimization problem [29].

A closely related concept to multimodality is the one of \textit{smoothness}, which refers to the amount of fluctuation in the landscape and can be informally classified as rugged, smooth or neutral. A \textit{rugged} landscape exhibits large fluctuations between points, presenting several local optima and steep ascents and descents. A \textit{neutral} landscape exhibits large flat areas or plateaus, where changes in the input do not generate significant changes in the output. These extremes make both gradient and correlation information less useful and makes it difficult to assert if one region is worth exploring. Non-smooth landscapes are usually the result of weak causality. While there is no viable method that can directly mitigate weak causality, it is often the result of a bad specification of the optimization problem [29].

Sometimes \( \mathcal{X} \) defines a global structure, which can be exploited to find \( x_o \). Therefore if the structure has fluctuations or modality, the problem becomes difficult to solve [30]. It is also possible to find problems who exhibit \textit{deceptiveness} — the global structure and gradient information leads the algorithm away from the optima, rendering the optimization algorithm less efficient than random search or exhaustive enumeration [29].

Another definition derived from the landscape metaphor is the \textit{basin of attraction} [27]. If we consider that a local search can be thought of as a function:

\[
\mu : \mathcal{X} \to \mathcal{X}
\]

where \( x_i \) is the initial point, \( \mu(x_i) \) is the local optimum that it reaches. Then, the basin of attraction of \( x_i \) can be defined as the set \( B(x_i) = \{ x : \mu(x_i) = x \} \). Hence, the search space is split into areas each with its own basin [13]. The shape and distribution of the basins are key attributes. A landscape with homogeneous basins is one such that \( |B(x_i)| = \frac{1}{n^d}, \forall x_i \in \mathcal{X} \), a desirable characteristic [30]. Also, a landscape can have a \textit{deterministic configuration} if its basins have the same size, (i.e., \( |B(x_i)| \approx |\mathcal{X}|/|\mathcal{X}|, \forall x_i \in \mathcal{X} \)); or a \textit{random configuration} if its basins have different sizes [13].

Variable scaling also has an effect on the shape of the basin of attraction. If a problem behaves differently in each dimension, it might be necessary to perform small steps on some dimensions and larger ones in others. Usually this is due to anisotropic or non-spherical basins. A problem can be even more difficult to solve if the variables are non-separable,
which means that the problem cannot be partitioned into subproblems of lower dimensionality each of which is easier to solve. Therefore, variations on the step size must be all encompassing.

It is possible that the search landscape possesses different properties at different locations, e.g. highly multimodal in some regions, while flat and neutral in others. Such globally inhomogeneous landscape would be difficult to characterize, since global analysis would generate contradictory conclusions. Hence, it is important that the developed methods are statistically isotropic, which means that they do not depend on the particular order of sampling nor the initial point where the search was begun [11], [19].

A problem with landscape theory is that the existent models cannot be used as framework when an algorithm uses multiple operators, or even with different representations [26]. This is due to the fact that every operator used defines its own landscape, a view which is referred to as “one operator, one landscape” [26], [31]. For example, genetic algorithms with binary representation can define up to three different landscapes: a mutation landscape, a crossover landscape and a selection landscape. All of these landscapes have different characteristics for different problems, and usually are studied independently.

Overall, the landscape metaphor allows the description of specific attributes of the search space, which are considered influential in the performance of an optimization algorithm. These attributes can be described qualitatively in cases where the knowledge about the function is complete. To illustrate this point we present the Comparing Continuous Optimization (COCO) noiseless benchmark, which is a set of 24 scalable functions [32] whose well known qualitative characteristics are shown in Table I. But in cases where the only information available are the pairs \((x_i, y_i)\), it is necessary to develop a set of techniques that can provide a numerical attribute. This procedure is known as landscape characterization.

## III. Methodology

In this section, we describe a new approach for landscape characterization. The rationale behind our approach, is based on two key issues: Firstly, it is difficult to represent all the attributes of a problem through a single measure. Secondly, although accurate estimators might be impossible to obtain, a sufficiently precise estimation might be enough to classify a problem. Hence, if a set of measures can be extracted during or after an algorithm run, and this set is related to a performance metric of the algorithm, characterization would provide some answers to the algorithm selection and tuning problems.

A first step in this endeavor is to modify currently existing methods so scattered data can be used. However, it is important to acknowledge that search algorithms are designed to explore promising areas of \(X\), which produces a large concentration of observations in small regions of the space. Hence, if raw data obtained during an algorithm run is used to extract any statistic measures, the results would suffer of sampling bias. This is a systematic error that causes some members of a population to be underrepresented in a sample, making impossible to infer the real characteristics of the population. We provide methods for the removal of such bias, and also for calculating the Information Content [11], [20], [21] and Basin of Attraction [10], [13], [22] measures from scattered data.

### A. Removing bias from datasets

To correct the error produced by the sampling bias a weighted resample can be used. Let \(x_i\) be an observation in a sample from \(X\) of size \(p\). Assume that each dimension of \(X\) has been split into \(\alpha\) sections forming a partition of \(\alpha^m\) bins. Let \(\eta_{0}, u=1, \ldots, \alpha^n\) be the number of elements in the bin \(u\). If \(x_i\) is an element of the bin \(u\), define \(\omega_i\) as the weight of the point \(x_i\), which is calculated as follows:

\[
\omega(i) = \frac{1}{\eta_{0}} \left(1 - \frac{\eta_u}{p}\right) = \frac{p - \eta_u}{p \cdot \eta_{0}}
\]

Now, let \(w_i\) be the empirical probability for \(x_i\) to be selected in the weighted resample, which is calculated by:

\[
w_i = \frac{\sum_{j=1}^{i} \omega_j}{\sum_{j=1}^{p} \omega_j}
\]

We select \(q\) observations from the original \(p\) using the inverse transformation method [33]. Two issues arise by using this method: first, it is possible that some elements have repeats, and second, not all the data from the original \(p\) might be selected in the resample. These issues are considered in the next sections.

### B. Information content measures for continuous optimization landscapes

Information content is a demonstrated measure for ruggedness and multimodality of a function, but its implementation requires a set of observations extracted during a random walk with variable step size [20], [21]. The results show that the selected step size influences the measures obtained. To solve this limitation we devise the following procedure. Assume that the sample of \(q\) observations is ordered in a sequence of points following the rules:

- The first element of the sequence, \(x_1\), is randomly selected.
- The next elements are selected following the rule:

\[
x_i = \{x : x \notin \{x_1, \ldots, x_{i-1}\}, x = \arg \min_{x} d(x, x_{i-1})\}
\]

where \(d(x, x_{i-1}) = \|x - x_{i-1}\|\). From the ordered sample we can create a symbol sequence \(\Phi(\epsilon) = \{\phi_1, \ldots, \phi_q\}\) such that \(\phi_i \in \{0, 1\}\) given by \(\phi_i(\epsilon) = \Psi(i, \epsilon)\) where:

\[
\Psi(i, \epsilon) = \begin{cases} 
1 & \text{if } \frac{|y - y_{i-1}|}{d(x, x_{i-1})} < -\epsilon \\
0 & \text{if } \frac{|y - y_{i-1}|}{d(x, x_{i-1})} \leq \epsilon \\
1 & \text{if } \frac{|y - y_{i-1}|}{d(x, x_{i-1})} > \epsilon 
\end{cases}
\]

where \(\epsilon \geq 0\) is a sensitivity parameter used to control the degree of detail obtained. By including the euclidean distance
between \(x_i\) and \(x_{i-1}\), the step size is taken into account in the analysis. Now the information content, which characterizes the ruggedness of the landscape is given by:

\[
H(\epsilon) = -\sum_{a \neq b} F_{ab} \log_6 F_{ab}
\]

where \(a, b \in \{1, 0, 1\}\) and \(F_{ab}\) is the frequency of the substring \(ab\) in the symbol string \(\Phi(\epsilon)\). The number of possible substrings where \(a \neq b\) is \(3!\), hence the logarithm is taken with base six so the information content is in the interval \([0, 1]\). As this is an entropy measure, \(0\log_6 1 = 0\) [20].

The partial information content characterizes the modality of the landscape. If a new string \(\Phi'(\epsilon)\) is constructed from \(\Phi(\epsilon)\) by removing all the 0’s and any repeated symbols. This new string would have the form of “…1111…”. Then, the partial information content \(M(\epsilon)\) is defined as:

\[
M(\epsilon) = \frac{|\Phi'|}{q - 1}
\]

From this analysis, we define the following measures as points of interest [20]: Maximum information content (1) and Maximum information sensitivity (2) are intended to identify unimodal from multimodal problems; Settling sensitivity (3) is intended to identify variable scaling differences; Initial partial information (4) and Half partial information sensitivity (5) are intended to detect neutral areas. The measures also are expected to provide redundancy between each other.

\[
H_{\text{max}} = \max_{\epsilon} \{H(\epsilon)\}
\]

\[
\epsilon_{\text{max}} = \max_{\epsilon} \{\epsilon : H(\epsilon) = H_{\text{max}}\}
\]

\[
\epsilon_{s} = \min_{\epsilon} \{\epsilon : H(\epsilon) < 0.05\}
\]

\[
M_{0} = M(\epsilon = 0)
\]

\[
\epsilon_{0.5} = \max_{\epsilon} \{\epsilon : M(\epsilon) > 0.5M_{0}\}
\]

Figure 1 shows some result of this analysis. The top figure shows through markers the location of \(H_{\text{max}}, \epsilon_{\text{max}}, \epsilon_{s}, M_{0}\) and \(\epsilon_{0.5}\). The bottom figure shows the results from three different functions from Table I: A unimodal-smooth function, \(f_{1}\), a unimodal-neutral function, \(f_{7}\), and a multimodal function, \(f_{24}\). The solid line represents the \(H(\epsilon)\) curve and the dashed line the \(M(\epsilon)\) curve.

C. Basin of attraction measures

In this section, we use the concept of basin of attraction to extract measures from data obtained during a search algorithm run. We start with a sample of \(q\) points, which may or may not contain repeated elements. The first step is to remove any repeats from this dataset. Next, define a neighborhood structure using a mesh calculated through Delaunay triangulation. The points which are only one jump distant in the mesh are neighbors.

To start the algorithm, we select a random point in the mesh, and follow the steepest descent heuristic over the grid. If the
point is a local minimum, we define the basin of attraction as the set of points in the path followed to reach the local optima. The procedure is repeated until all points have been visited at least once. To avoid additional computational effort, if the point currently under evaluation has been visited previously, we add the current path to the basin of attraction of the current point. Figure 2 shows the result of the algorithm for the well known Six-hump Camel Back function. We use this function for this example as its low modality and similar basin sizes allow the generation of a clear graphic. The algorithm identifies effectively the four local optima and two global optima. We use this function from Table I with different qualitative characteristics.

From this analysis we extract the following measures:

- Number of Local Optima (NLO), which is an approximation of \(|X_i|\) and identifies if the problem is unimodal or multimodal.
- Basin Size Ratio (BSR), which is calculated by (6) and it characterizes the existence of a dominant basin.

\[
BSR = \frac{\max_{x_i} |B(x_i)|}{\min_{x_i} |B(x_i)|}
\]  

(6)

IV. EXPERIMENTAL PROCEDURE AND RESULTS

To test the performance of our methods, we use the COCO noisless benchmark from Table I with \(n = 2\) and three different sampling methods: Random sampling (RS), (1+1)CMA-ES [34], and the Particle Swarm Optimization (PSO) with Trelea’s second parameter set [35]. RS is the comparison algorithm, and the other two algorithms were selected based only in their popularity. For each algorithm fifteen runs of 2000 function evaluations are made. If the algorithm converges to an optimal point during its run before completing the allocated evaluations, the algorithm is restarted at random to complete its allocation.

The required parameters were set as follows: the number of splits per dimension, \(\alpha\), was set at 50, the number of resamples was set at 100, and the number of points per resample, \(q\), was set as 2000. For the Information content method, \(\epsilon\) have values of 0, and logarithmically increasing from \(10^{-5}\) to \(10^{15}\). The number of different values tested for \(\epsilon\) is 1000.

A. Resampling algorithm

To evaluate the performance of the resampling method, we use the Pearson \(\chi^2\) goodness of fit test [36] over sample of 2000 observations in partition with \(\alpha = 5\). The test is equal to one if the sample is uniformly distributed and it is zero otherwise. We carry out the experiment over the 100 samples per function and averaged the total results. To evaluate the efficiency of the method, we compare five different scenarios: RS sampling, PSO without removing the bias, PSO removing the bias, CMA-ES without removing the bias and CMA-ES removing the bias. The results, shown in Table II, confirm that applying our method produces a uniformly distributed sample.

To illustrate the effect of the resampling algorithm, we present a histogram for 1000 points extracted from the PSO dataset over the \(f_1\) function, from biased (Fig. 3(a)) and debiased (Fig. 3(b)) datasets. It is noteworthy the difference between the two distributions, where the biased dataset has a high concentration of data in one sector. This sector is the location of the global optima found during the experiments with PSO.

Through these experiments, it is demonstrated that the resampling algorithm performs adequately. As the diversity in the dataset is higher, it is possible to obtain a uniform dataset. This step is necessary if we are to obtain adequate landscape

**TABLE II**

<table>
<thead>
<tr>
<th>No.</th>
<th>RS Biased</th>
<th>PSO Biased</th>
<th>CMA-ES Biased</th>
<th>RS Debiased</th>
<th>PSO Debiased</th>
<th>CMA-ES Debiased</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>1.0000</td>
<td>0.1717</td>
<td>1.0000</td>
<td>0.1600</td>
<td>0.9946</td>
<td></td>
</tr>
</tbody>
</table>

- Maximum distance between basins (\(d_{\text{max}}\)), and Distance between the global optimum and the local optimum with the largest attraction basin (\(d_{\text{opt}}\)), which detect deceptive characteristics of the landscape.

- Basins Size Ratio (BSR), which is calculated by (6) and it characterizes the existence of a dominant basin.

- Maximum distance between basins (\(d_{\text{max}}\)), and Distance between the global optimum and the local optimum with the largest attraction basin (\(d_{\text{opt}}\)), which detect deceptive characteristics of the landscape.
The results show that the resampling effectivity reduces the difference between results, with exception of $M_0$. Larger error in the basin of attraction measures is not as relevant, since they can be used in relative ranking. It is noteworthy that the PSO algorithm produces less biased results, when compared with $(1+1)$CMA-ES. This has a simple explanation. Since PSO is a population algorithm, it samples at different positions concurrently. Therefore, the distribution of the observations is closer to uniform, which was also demonstrated by the data from Table II.

C. Measure results and analysis

The measurements obtained through RS are shown in Table IV. To interpret this table it is convenient to use the categorization in Table I and the descriptions by Finck, et al. [37]. Functions $f_1$ and $f_5$ are similar according to their qualitative characteristics and present close values for $\varepsilon_{\text{max}}$, $\varepsilon$, $M_0$, and $\varepsilon_{0.5}$. While $f_2$ has the same qualitative characteristics, this function has a large anisotropic basin of attraction that forms a narrow valley, and it also possesses minor smooth local irregularities. The existence of the anisotropic basin produces differences between displacements in different dimensions. This difference is expressed by higher values of $\varepsilon_{\text{max}}$ and $\varepsilon$. Non smooth problems are usually represented by values of $H_{\text{max}}$ higher than 0.7, although an exception is function $f_3$. In particular, Functions $f_{10}$, $f_{19}$, $f_{23}$ and $f_{24}$ which can be considered almost noisy present the highest values of $H_{\text{max}}$. Function $f_1$ is the only one that presents plateaus, which is detected by low $\varepsilon_{\text{max}}$ and $M_0$, and can be identified as non smooth by $H_{\text{max}}$. The existence of anisotropic basins of attraction, such as those presented in functions $f_5$, $f_6$, $f_8$ and $f_9$ is detected by a high value of $\varepsilon_{0.5}$. Function $f_{12}$, which has the most anisotropic narrow basin of attraction has the highest value of them all. Functions $f_8$ and $f_9$, $f_3$ and $f_{10}$, and $f_3$ and $f_{15}$ are rotated versions of themselves, and as such they share similar measurements values.

Unimodal functions with isotropic basin of attraction such as $f_1$, $f_3$ and $f_{14}$ are easily identified by the NLO measure. Unimodal functions with anisotropic basins are identified as having several local optima, such as $f_2$, $f_{19}$, $f_{10}$ and $f_{12}$. This effect is a result of the meshing procedure, which divides the unique basin into sections. However, since the number of divisions is low, it is possible to avoid confusing this type of problems with medium modality problems, such as Functions $f_{20}$ and $f_{22}$, or highly multimodal problems such as Functions $f_3$, $f_4$, $f_{19}$ and $f_{24}$. Function $f_7$, which is the only one with neutral areas, is identified as having several local optima. Because the local search is unable to move after it hits a neutral area, this is considered the end of the search. Hence, this increases the number of local optima found.

Some clues about the distribution of the basin sizes can be obtained by analyzing both NLO and BSR. If a space has relatively uniform basins then $BSR \rightarrow 1$. If this ratio becomes smaller, e.g. on $f_{20}$, it is possible that the algorithm explores one area of the search space, as its basin is more attractive than others. If a highly multimodal problem presents a large
value of $d_{opt}$ then it is possible that the problem has some deceiving characteristics. The sampling bias imposed by the algorithms would change significantly our conclusions. If we consider $H_{max}$ and the error imposed by the bias was incremental, this would mean that smooth functions such as $f_1$, $f_2$ and $f_3$, could be classified as rugged. However, if the error was decremental, highly rugged functions such as $f_{23}$ and $f_{24}$ would be classified as smooth. The change in the position of $\epsilon_{max}$ also would indicate a possible anisotropic problem for function $f_1$.

These results show that the method can be used to identify effectively ruggedness and neutrality in the functions. The inspection of the structure of the function allows the correlation between some values and the shape of the basin of attraction.

V. DISCUSSION AND FURTHER WORK

Developing a complete method to characterize a numerical optimization problems is a complex task. A key element of this endeavor is to acknowledge that a single measure cannot encapsulate all characteristics that make a problem difficult. Hence, it is necessary to develop alternative measures tailored for specific characteristics. In this work, we have demonstrated that it is possible to extract such measures, not only from random data, but also from data extracted during an algorithm run. Although it is necessary to pre-process the information to avoid sampling bias, this eliminates the need for additional experimentation.

In this paper, we have presented a new landscape characterization method for numerical optimization problems as a proof of concept. However, to create a complete characterization method there are several issues to be addressed. Firstly, we acknowledge that the COCO Benchmark does not have sufficient types of problems. Specifically, there are not sufficient problems with neutral areas. The methods developed in this paper detect neutrality with a low $M_0$, low $\epsilon_{max}$, and high $NLO$ for the Step Ellipsoidal function. At this point, it is not possible to state definitely that all neutral problems have the similar measurement values. The inclusion of more random test functions to the database would be ideal. Secondly, it is necessary to scale the methods to higher dimensions as practical problems have dimensionality much higher than two. Some difficulties to be addressed for this purpose are the selection of a structure for neighborhood definition and the limited number available observations. Thirdly, it is necessary to develop other measures that identify characteristics not evaluated by this paper. Variable separability is one of those
characteristics which has major influence in the performance of an algorithm. However, as the analysis methods become more detailed, some measures might become redundant and unnecessary to calculate. The pruning of those measures would make the calculations more effective.

Finally, we expect to predict the performance of search algorithms in a function given a set of characteristics. For that it is possible to define a function $h: \mathcal{C} \mapsto \mathcal{P}$, where $\mathcal{C}$ is the space of characteristics of the problem and $\mathcal{P}$ is the space of performance of the algorithm. A nonlinear estimator of this function might be able to predict the possible performance of an algorithm if the training examples are sufficient and the input variables relevant. We will devote our further work to explore all of these ideas.

REFERENCES


