Motif Difficulty (MD): A Predictive Measure of Problem Difficulty for Evolutionary Algorithms using Network Motifs

Jing Liu

jing.liu@adfa.edu.au

School of Engineering and Information Technology, The University of New South Wales at the Australian Defence Force Academy, Canberra, ACT 2600, Australia

Hussein A. Abbass

h.abbass@adfa.edu.au

School of Engineering and Information Technology, The University of New South Wales at the Australian Defence Force Academy, Canberra, ACT 2600, Australia

David G. Green

david.green@monash.edu

Clayton School of Information Technology, Monash University, Clayton 3800, Australia

Weicai Zhong

w.zhong@adfa.edu.au

School of Engineering and Information Technology, The University of New South Wales at the Australian Defence Force Academy, Canberra, ACT 2600, Australia

Abstract

One of the major challenges in the field of Evolutionary Algorithms (EAs) is to characterise which kinds of problems are easy and which are not. Researchers have been attracted to predict the behaviour of EAs in different domains. Based on fitness landscape networks (FLNs) built by operators satisfying specific requirements, we define a new predictive measure, namely Motif Difficulty (MD), for comparison-based EAs. Since exhaustive computation on whole networks becomes quickly impractical, we propose a sampling technique for calculating the approximate MD. Extensive experiments on binary search spaces are conducted to show both advantages and limitations of MD. Multidimensional knapsack problems (MKPs) are also used to validate the performance of approximate MD on FLNs with different topologies. The effect of two representations, namely binary and permutation, on the difficulty of MKPs is analysed.

Keywords

Evolutionary algorithm, problem difficulty, predictive measure, fitness landscape, complex network, network motif.

1 Introduction

Intrinsically, evolutionary algorithms (EAs) (Forgel, 1999; Bäck et al., 1997; Goldberg, 1989) are still stochastic algorithms. Thus, one of the major challenges in this field is to characterise which kinds of problems are easy for a given algorithm to solve and which are not. In this direction, researchers have been attracted to predict the behaviour of EAs in different domains and have proposed predictive measures to quantify problem difficulty. The major tool used in available predictive measures is fitness landscapes (FLs). The concept of a FL was introduced in theoretical genetics (Wright, 1932) as a way to visualise evolutionary dynamics. FLs were connected to EAs via a neighbourhood structure based on operators used in EAs, which highlights the association

between search spaces and fitness spaces. With the property of neighbourhood structure in mind and at some level of granularity, each FL can form a network. In such a network, each node corresponds to a point in the search space, and each edge connects one node to one of its neighbours. The fitness value can be viewed as the weight of each node. Therefore, different problems correspond to different networks, and the process of EAs solving problems is equivalent to navigating through these networks. From this viewpoint, problem difficulty can be predicted by analysing the features of FL networks (FLNs). Although it is well-known that FLs are related to networks, no predictive measure has been proposed based on the features of FLNs.

A number of global features of complex networks, such as the small-world property, the clustering property, and the scale free property have been studied thoroughly. In addition to these global features, “network motifs” (Milo et al., 2002) were proposed to uncover the structural design principles and resulted in a deeper understanding of complex networks. Motifs are connected sub-graphs occurring in complex networks at numbers that are significantly higher than those in random networks. It has been shown that network motifs exist widely in various complex networks, and different complex networks have different types of motifs. Thus, network motifs are in fact an intrinsic property of complex networks, and can be used to differentiate different networks. For example, Ghoneim et al. (2008) used network motifs to discriminate two-player strategy games. For EAs, FLNs corresponding to various problems are expected to be different in some intrinsic properties. Therefore, we propose a predictive difficulty measure for EAs, namely Motif Difficulty (MD), by extracting motif properties from directed FLNs. We define MD by synthesising the effect of different classes of distance motifs on the searching process. Our experimental results show that MD can quantify the difficulty of different problems into the range of -1.0 (easiest) to 1.0 (most difficult), and performs especially well on some counterexamples for other measures.

This paper is organised as follows. Section 2 discusses related work. Preliminaries on fitness landscapes and network motifs are given in Section 3. Section 4 presents the definition of motifs in FLNs, and Section 5 presents a qualification of problem difficulty based on the motifs defined in Section 4. Experiments on exact and approximate MD are given in Sections 6 and 7, respectively. Finally, Section 8 summarises the work in this paper, and discusses both advantages and disadvantages of MD.

2 Related Work

In general, the study of factors affecting the performance of EAs can be divided into two classes (Borenstein and Poli, 2005a). The first class focuses on the properties of a particular algorithm, while the second class focuses on the problem itself, and particularly on FLs. In the first class, the BB hypothesis (Goldberg, 1989), which is famous in the Genetic Algorithm (GA) community, states that a GA tries to combine low order and highly fit schemata. Following the BB hypothesis, the notion of deception has been defined (Goldberg, 1989; Forrest and Mitchell, 1993). Epistasis variance (Davidor, 1991) and epistasis correlation (Naudts, 1998) try to assess the GA-hardness of problems from the perspective of theoretical genetics.

In the second class, methods focus on using statistical properties of FLs to characterise problem difficulty. The first study in this class proposed isolation (needle-in-a-haystack) (Forrest and Mitchell, 1993) and multimodality (Davidor, 1991). The other popular method is Fitness Distance Correlation (Jones and Forrest, 1995), which measures the hardness of a landscape according to the correlation between the distance from the optimum and the fitness value of the solution. However, none of the available

measures fully achieved success. Isolation might be sufficient, but it is not a necessary condition for a landscape to be difficult to search. Multimodality is neither necessary nor sufficient for a landscape to be difficult to search (Kallel et al., 2001). FDC has achieved some success, but is still not able to predict the performance in some scenarios (Naudts and Kallel, 2000; Jansen, 2001).

In addition to FLs, Borenstein and Poli (2005a) pointed out that a limitation of the original FL approach is that it does not provide a way to quantify the amount of information available in a landscape nor to assess its quality. Thus, they proposed Information Landscapes (ILs) based on tournament selection in GAs. Using ILs, they proposed a method to predict GA hardness and a theoretical model to study search algorithms (Borenstein and Poli, 2005b,c). Based on the observation that FLs can actually form networks, Ochoa et al. (2008) proposed a network characterisation of combinatorial FLs, and use the well-known family of NK landscapes as an example, and exhaustively extract local optima networks on NK landscape instances. This work is the first attempt at using network analysis techniques in connection with the study of FLs and problem difficulty. However, they did not propose predictive measures.

He et al. (2007) gave a rigorous definition of difficulty measures in black-box optimisation, and proved that in general predictive difficulty measures that run in polynomial time do not exist unless certain complexity-theoretical assumptions are wrong. However, there are still some successful applications of using predictive difficulty measures to guide the design of new algorithms. For example, Merz and Freisleben (2000) and Tavares et al. (2008) conducted a FL analysis for quadratic assignment problems and multidimensional knapsack problems (MKPs), respectively; Yang et al. (2006) presented an attempt at characterising the search space difficulties in red teaming by using fitness landscapes.

3 Preliminaries

3.1 Fitness Landscapes

Formally, a fitness landscape *FL* is defined by a tuple of three components:

$$FL = (S, f, N) \quad (1)$$

where S is the search space. In this paper, we consider only a finite discrete search space for combinatorial optimisation problems. The fitness function $f: S \rightarrow \mathbf{R}$ assigns a numeric value to each configuration in S , and we consider maximisation problems here. N is a neighbourhood structure defined over S as follows.

$$\forall s \in S, N(s) = \{y \in S | P(y \in \text{operator}(s)) > 0\} \quad (2)$$

where $P(e)$ denotes the occurrence probability of event e , and $\text{operator}(s)$ denotes the set of configurations that can be obtained by performing *operator* on s , namely the set of neighbours of s . Since EAs are determined by various operators, the above neighbourhood structure defined over operators reflects the features of different EAs, and correspondingly, these features can be reflected in the resulted FLs.

If we treat each configuration in the search space as a node, and link each node to all its neighbours, then each FL can be mapped into a network, namely a fitness landscape network *FLN*, which is also defined by a tuple of three components:

$$FLN = (V, E, f) \quad (3)$$

Figure 1: All 13 types of three-node connected subgraphs (Milo et al., 2002).

where V is the finite set of nodes and each node corresponds to a configuration in S . f is the same as that in (1). E is a set of undirected edges and each edge connects one configuration to one of its neighbours; that is,

$$E = \{\{x, y\} | x, y \in S \wedge (x \in N(y) \vee y \in N(x))\} \quad (4)$$

3.2 Network Motifs

Network motifs are patterns of inter-connections, which can be reflected by connected subgraphs (Milo et al., 2002). Milo et al. (2002) showed that there are 13 types of three-node connected subgraphs (Figure 1) for all directed networks. In a directed network, a pattern of inter-connections can be viewed as a network motif only when its number of occurrences in this network is significantly higher than that in the corresponding random network. More specifically, network motifs are those connected subgraphs for which the probability of appearing in a random network an equal or greater number of times than in the real network is lower than a cutoff value 0.01 (Milo et al., 2002).

To detect n -node network motifs in a network, the network was scanned for all of the possible n -node subgraphs, and the number of occurrences of each subgraph was recorded (Milo et al., 2002). Milo et al. (2002) showed that several networks (i.e., gene regulation, neurons, food webs, electronic circuits, and world wide web) exhibit different types of network motifs, and frequencies of different network motifs vary from one network to another. Motifs may thus define universal classes of networks, and are basic building blocks of most networks. Therefore, network motifs have been widely used in studying complex systems and in characterising features on the system level by analysing locally how the substructures are formed.

4 Motifs in Directed Fitness Landscape Networks

In EAs, when an individual moves from one node to its neighbour under the effect of an operator, it is equivalent to exploring the local structures or subgraphs. Thus, subgraphs, which can be reflected by motifs, that an EA has visited during the whole evolutionary process have a close relationship with its performance. Since network motifs are just connected subgraphs, obviously, the number of possible motifs in undirected networks is much lower than that in directed networks. Moreover, the topology of FLNs built by the same operator is identical if the fitness value of each node is ignored. Thus, we first convert undirected FLNs to directed FLNs (DFLNs) by making use of fitness values so that more different types of network motifs can be extracted. Second, we propose another way to consider the 13 types of possible three-nodes motifs presented in Figure 1, and define six types of basic motifs. Finally, based on these basic motifs, a new type of motifs, namely distance motifs, is designed by taking global optima as references.

4.1 Directed Fitness Landscape Networks and Basic Motifs

Definition 1: Define a fitness landscape network as $FLN = (V, E, f)$, then the corresponding **Directed Fitness Landscape Network** is:

$$\overrightarrow{FLN} = (V, \overrightarrow{E}) \quad (5)$$

where $\overrightarrow{E} = \{(x, y) | \{x, y\} \in E \wedge f(x) \leq f(y)\}$ is a set of directed edges.

Given two nodes x and y , y is the neighbour of x as long as the probability of converting x to y by *operator* is larger than 0, then the edge $\{x, y\}$ exists in the corresponding network. In this way, a complete graph will be generated when using the mutation operator in which each bit will be flipped with probability $1/n$ for a binary string of length n (labelled as $1/n$ mutation in the following text), and the probabilities of transforming this string to different neighbours are not always the same. These incur some problems in detecting network motifs. First, if an operator leads to a complete graph, the topology of fitness landscapes for all problems is the same. Second, if the probabilities of transforming a string to different neighbours are not always the same, which is equivalent to the fact that edges have different weights, the edges in a motif may have different importance, and we cannot just count the number of each type of motifs. Therefore, in the following text, for the sake of simplicity, the operator used to build DFLNs must satisfy the conditions in (6) and (7). That is to say, the number of neighbours of each node must be much smaller than the number of nodes in the network, and the probabilities of converting one node to different neighbours must be identical. Edges in DFLNs use only the relative difference between the fitness values of two nodes rather than the absolute fitness values. Thus, DFLNs are suitable for analysing comparison-based search algorithms, such as (1+1)EA (Droste et al., 2002).

$$\forall x \in V, |N(x)| \ll |V| \quad (6)$$

$$\forall y, z \in N(x), P(\text{operator}(x) = y) = P(\text{operator}(x) = z) \quad (7)$$

The original network motifs are defined on the difference in frequency from random networks (Milo et al., 2002). However, random FLNs are built from a random fitness function. The difficulty of this random fitness function for EAs is also a part of study, so we cannot take it as a reference to detect network motifs in DFLNs. In fact, if we further check the 13 types of possible three-node motifs listed in Figure 1, we can find that motifs of types 1, 2, 3, 4, 7, and 8 are subsets of motifs of types of 5, 6, 9, 10, 11, 12, or 13. Therefore, in this study, for a three-node subgraph, we consider only the edges between two pairs of nodes, namely motifs of types 1, 2, 3, 4, 7, and 8 in Figure 1, which are named as Basic Motifs.

Definition 2: Given a directed fitness landscape network $\overrightarrow{FLN} = (V, \overrightarrow{E})$, a **Basic Motif**, labelled as M^b , in $\overrightarrow{FLN}$ is a connected three-node subgraph,

$$M^b = \{V_{M^b} = \{n_1, n_2, n_3\} \subset V, \overrightarrow{E}_{M^b} = \{(n_1, n_2), (n_2, n_1), (n_2, n_3), (n_3, n_2)\} \cap \overrightarrow{E}\} \quad (8)$$

To let M^b be a connected subgraph, $\overrightarrow{E}_{M^b}$ should satisfy

$$((n_1, n_2) \in \overrightarrow{E}_{M^b} \vee (n_2, n_1) \in \overrightarrow{E}_{M^b}) \wedge ((n_2, n_3) \in \overrightarrow{E}_{M^b} \vee (n_3, n_2) \in \overrightarrow{E}_{M^b}) \quad (9)$$

Table 1: The relationship between basic motifs and distance motifs.

Type of basic motifs		Class of Distance Motifs			
No.	Motifs	Condition	Class	Condition	Class
Group₀		Neutral			
Group₁		$d_3 \geq d_2$	Guide	$d_3 \geq d_1$	Core Guide
		$d_3 < d_2$	Deceptive	$d_3 > d_1$	Core Deceptive
Group₂		$d_2 \geq d_3$	Guide	$d_1 \geq d_3$	Core Guide
		$d_2 < d_3$	Deceptive	$d_1 > d_3$	Core Deceptive
Group₃		$d_1 \geq d_3$	Guide	$d_1 \geq d_2$ and $d_2 \geq d_3$	Core Guide
		$d_1 < d_3$	Deceptive	$d_1 > d_2$ and $d_2 > d_3$	Core Deceptive
Group₄		$d_2 \geq d_1$ and $d_2 \geq d_3$		Guide	
		$d_2 < d_1$ or $d_2 < d_3$		Deceptive	
Group₅		$d_1 \geq d_2$ and $d_3 \geq d_2$		Guide	
		$d_1 < d_2$ or $d_3 < d_2$		Deceptive	

Clearly, based on the edges between $\{n_1, n_2\}$ and $\{n_2, n_3\}$, there are six types of basic motifs in total for all DFLNs, which are labelled as **Group₀** - **Group₅** and shown in the first two columns of Table 1. These six types of basic motifs can be viewed as basic building blocks of a DFLN.

4.2 Distance Motifs

An selection mechanism in EAs, such as binary tournament selection, causes the search heuristic to prefer high fitness regions. Thus, the problems would be easy if high fitness regions are close to global optima. On the contrary, if high fitness regions are far from global optima, the selection mechanism may lead the search heuristic in a wrong direction. This indicates that the distance between candidate solutions and global optima is another important factor that impacts EAs' performance. Thus, we use the distance information to refine basic motifs. There are a number of ways to define the distance between a candidate solution and a global optimum. Naturally, the one that is most suitable for the searching process on fitness landscapes is the shortest path length. To calculate the shortest path length, we need to identify the nodes in a network that correspond to the global optima. However, for indirect encoding methods, like the permutation encoding for MKPs, the computational cost to match the points in the search

space to the global optima may be quite high. For binary encoding, since the expected number of bits that are flipped in the $1/n$ mutation is 1, we use the 1-bit flip operator to build the DFLN, and then the Hamming distance is used as the distance measure, which is equivalent to the shortest path length in this case. For other encoding methods, if the computational cost to identify the nodes corresponding to global optima and calculate the shortest path length is feasible, the shortest path length can be used as the distance measure; otherwise, an approximate way should be designed.

Definition 3: Given a basic motif, $M^b = \{V_{M^b} = \{n_1, n_2, n_3\}, \vec{E}_{M^b}\}$, the corresponding **Distance Motif** is $M^d = \{V_{M^d}, \vec{E}_{M^d}, \{d_1, d_2, d_3\}\}$, where d_1 , d_2 , and d_3 are respectively the distances attached to n_1 , n_2 , and n_3 .

When the searching process visits a basic motif from n_1 to n_2 to n_3 , we can check d_1 , d_2 , and d_3 to see whether the searching process is heading towards the right direction or not. Clearly, when we check a direction is correct or not, we should take into account the information of both fitness and distance. Since the information of fitness is reflected by edges in DFLNs, we first define paths in a distance motif. Then, based on their contributions to the searching process, distance motifs are divided into three classes.

Definition 4: Given a distance motif, $M^d = \{V_{M^d} = \{n_1, n_2, n_3\}, \vec{E}_{M^d}, \{d_1, d_2, d_3\}\}$. For $\forall n_i, n_j \in V_{M^d}$, if n_i can reach n_j by only visiting edges in $\vec{E}_{M^d}$, then there is a **Path** between n_i and n_j . Furthermore, for each edge on a path in M^d , if the edge with inverse direction does not exist, then this path is an **Effective Path**.

Definition 5: Given a distance motif, $M^d = \{V_{M^d}, \vec{E}_{M^d}, \{d_1, d_2, d_3\}\}$. If there is no effective paths in M^d , then M^d is a **Neutral Motif**. If all effective paths with the largest length (the number of edges in the path) in M^d satisfies that the distance of the start node is not less than that of the end node, then M^d is a **Guide Motif**; otherwise, it is a **Deceptive Motif**.

In a path, the node with lower fitness value always points to the node with higher fitness value. When the distance of an end node on a path is not larger than that of the other end node, then it means that when the fitness value increases, the distance decreases. In such a case, the motif has a positive effect on the searching process, thus, it is a guide motif. Table 1 illustrates how the three classes of distance motifs are related to the six types of basic motifs. For **Group₀**, since the set of effective paths is empty, all motifs in **Group₀** are neutral motifs. For **Group₁**, the only effective path is between n_3 and n_2 . Thus, if $d_3 \geq d_2$, then they are guide motifs; otherwise, deceptive motifs. The case for **Group₂** is similar. For **Group₃**, there are three effective paths, namely the paths between n_1 and n_2 , n_2 and n_3 , and n_1 and n_3 , whose lengths are 1, 1, and 2, respectively, and only the path with the largest length need to be considered. Thus, if $d_1 \geq d_3$, then they are guide motifs; otherwise, deceptive motifs. For **Group₄**, there are two effective paths, namely the paths between n_2 and n_1 , n_2 and n_3 , whose lengths are both 1, and both of them need to be considered. Thus, if both $d_2 \geq d_1$ and $d_2 \geq d_3$, then they are guide motifs; otherwise, deceptive motifs. The case for **Group₅** is similar.

If we further analyse guide and deceptive motifs, paths with length two exist in some cases. On such paths, if the difference in distances and fitness values of each pair of nodes connected by an edge is consistent with that of two end nodes, then the corresponding motifs have a more explicit impact on the searching process. Thus, we further define two sub-classes of guide and deceptive motifs.

Definition 6: Given a guide motif, $M^G = \{V_{M^G}, \vec{E}_{M^G}, \{d_1, d_2, d_3\}\}$. If there exists a path with length two, labelled as $\{(n_i, n_k), (n_k, n_j)\}$, where $n_i, n_k, n_j \in V_{M^G}$ and $d_i \geq d_j$, satisfies the conditions that $d_i \geq d_k$ when $(n_k, n_i) \notin \vec{E}_{M^G}$ and $d_k \geq d_j$ when

$(n_j, n_k) \notin \vec{E}_{M^G}$, then M^G is a **Core Guide Motif**.

Definition 7: Given a deceptive motif, $M^D = \{V_{M^D}, \vec{E}_{M^D}, \{d_1, d_2, d_3\}\}$. If there exists a path with length two, labelled as $\{(n_i, n_k), (n_k, n_j)\}$, where $n_i, n_k, n_j \in V_{M^D}$ and $d_i < d_j$, satisfies the conditions that $d_i < d_k$ when $(n_k, n_i) \notin \vec{E}_{M^D}$ and $d_k < d_j$ when $(n_j, n_k) \notin \vec{E}_{M^D}$, then M^D is a **Core Deceptive Motif**.

According to Definitions 6 and 7, none of the motifs in **Group₄** and **Group₅** is core guide or deceptive motifs because no path with length two exists. The relationship with other groups of basic motifs is also shown in Table 1.

5 Predictive Difficulty Measures based on Distance Motifs

Guide motifs can mostly help the searching process head towards the right direction. This is because when the searching process goes from nodes with lower fitness values to nodes with higher fitness values, it naturally and smoothly approaches global optima. On the contrary, deceptive motifs mostly lead the searching process in the wrong direction since when the searching process goes from nodes with lower fitness values to nodes with higher fitness values, it deviates away from global optima. For neutral motifs, fitness values of the three nodes are the same, thus, neither deceptive nor guide information is provided. Next, we first analyse DFLNs of some well-studied fitness functions to show how the motifs composing a DFLN are related to problem difficulty. Then, based on the analyses, predictive difficulty measures are proposed.

5.1 DFLNs of Well-Studied Fitness Functions

5.1.1 Effect of the Amount of Different Types of Motifs on Problem Difficulty

Three classical fitness functions defined over the binary search space are used, namely *ONEMAX*, *NIAH*, and *TRAP* (Droste et al., 2002).

$$ONEMAX(s) = \sum_{i=1}^n s_i \quad (10)$$

$$NIAH(s) = \begin{cases} 1 & \text{If } s = s^* \\ 0 & \text{If } s \neq s^* \end{cases} \quad (11)$$

$$TRAP(s) = \sum_{i=1}^n s_i + (n+1) \cdot \prod_{i=1}^n (1 - s_i) \quad (12)$$

where $s = (s_1, s_2, \dots, s_n)$ and $s_i = 0$ or $1, i = 1, 2, \dots, n$. s^* is the global optimum.

Proposition 1: If a fitness landscape, $FL = \{S, f, N\}$, where S is composed of binary strings, $f = ONEMAX$, and the 1-bit flip operator is used to construct N , then distance motifs in the corresponding directed fitness landscape network are guide motifs.

Proof. Let a distance motif in the corresponding DFLN be $M^d = \{(n_1, n_2, n_3), \vec{E}_{M^d}, \{d_1, d_2, d_3\}\}$. If an edge $(n_i, n_j) \in \vec{E}_{M^d}$, then according to the 1-bit flip operator and *ONEMAX*, we have

$$f(n_j) = 1 + f(n_i) \Rightarrow f(n_j) > f(n_i) \Rightarrow (n_j, n_i) \notin \vec{E}_{M^d} \quad (13)$$

Thus, M^d belongs to **Group₃**, **Group₄**, or **Group₅**.

Case 1: M^d belongs to **Group₃**.

$$\begin{aligned} \vec{E}_{M^d} = \{(n_1, n_2), (n_2, n_3)\} &\Rightarrow f(n_1) < f(n_2) < f(n_3) \\ &\Rightarrow \#1(n_1) < \#1(n_2) < \#1(n_3) \end{aligned} \quad (14)$$

where $\#1(\cdot)$ denotes the number of '1' in a candidate solution. Since the global optimum is the string of all '1', we have $d_1 > d_2 > d_3$; that is, M^d is a core guide motif.

Case 2: M^d belongs to **Group**₄.

$$\begin{aligned} \vec{E}_{M^d} = \{(n_2, n_1), (n_2, n_3)\} &\Rightarrow f(n_2) < f(n_1) \text{ and } f(n_2) < f(n_3) \\ &\Rightarrow \#1(n_2) < \#1(n_1) \text{ and } \#1(n_2) < \#1(n_3) \end{aligned} \quad (15)$$

Then, we have $d_2 > d_1$ and $d_2 > d_3$; that is, M^d is a guide motif.

Case 3: M^d belongs to **Group**₅.

$$\begin{aligned} \vec{E}_{M^d} = \{(n_1, n_2), (n_3, n_2)\} &\Rightarrow f(n_1) < f(n_2) \text{ and } f(n_3) < f(n_2) \\ &\Rightarrow \#1(n_1) < \#1(n_2) \text{ and } \#1(n_3) < \#1(n_2) \end{aligned} \quad (16)$$

Then, we have $d_1 > d_2$ and $d_3 > d_2$; that is, M^d is a guide motif.

To summarise, in all the three cases, M^d is a guide motif. $\square$

Before we give Proposition 2, we first prove some properties of DFLNs built by the 1-bit flip operator.

Lemma 1: In a fitness landscape network built by the 1-bit flip operator, (1) the number of edges connecting any three nodes is less than three, (2) each node belongs to $\frac{3n(n-1)}{2}$ basic motifs where n is the dimension of the search space, (3) the number of all basic motifs is $2^{n-1}n(n-1)$.

Proof. (1) Suppose there are three nodes $\{n_1, n_2, n_3\}$ connected by three edges. Without loss of generality, let the set of edges connecting these three nodes be $\{\{n_1, n_2\}, \{n_2, n_3\}, \{n_3, n_1\}\}$, and the configurations corresponding to these three nodes be s_1, s_2, s_3 . Then, we have

$$H(s_1, s_3) = 1 \quad (17)$$

where $H(\cdot, \cdot)$ denotes the Hamming distance. But we also have

$$H(s_1, s_2) = 1 \text{ and } H(s_2, s_3) = 1 \Rightarrow H(s_1, s_3) = 0 \text{ or } 2 \quad (18)$$

Clearly, (17) and (18) contradict each other. Thus, the number of edges connecting any three nodes is less than three.

(2) The number of edges connected to each node is n . According to Lemma1(1), any three nodes can form only 1 basic motif at most. Thus, all distance motifs that a node n_1 belongs to can be divided into two cases; that is, (a) both n_2 and n_3 connect to n_1 and (b) n_1 connects to n_2 and n_2 connects to n_3 . The number of basic motifs that n_1 belongs to is

$$\binom{n}{2} + n(n-1) = \frac{3n(n-1)}{2} \quad (19)$$

(3) The proof for Lemma1(2) shows that basic motifs that each node belongs to can be divided into two cases. In fact, Case 2 for n_1 is equivalent to Case 1 for n_2 . Therefore, to prevent repetition, we need to count only Case 1 for each node to calculate

the number of basic motifs in the whole network. Since the number of nodes in the search space is 2^n , the number of all basic motifs is

$$2^n \binom{n}{2} = 2^{n-1} n(n-1) \quad (20)$$

□

Proposition 2: Given a fitness landscape, $FL = \{S, f, N\}$, where S is composed of binary strings with length n , $f = NIAH$, and the 1-bit flip operator is used to construct N . Then, the fraction of neutral motifs in the corresponding directed fitness landscape network is equal to or greater than $1 - 3 \times 2^{-n}$.

Proof. Given a distance motif M^d , and the configurations corresponding to three nodes in M^d are not the global optimum, then the fitness values of these three nodes are 0. Thus, M^d belongs to **Group**₀, namely a neutral motif.

That is to say, only motifs that include the node corresponding to the global optimum, labelled as n^* , can be a guide or deceptive motif. According to Lemma 1(2), the number of motifs that n^* belongs to is $\frac{3n(n-1)}{2}$. Thus, the number of neutral motifs is no less than

$$1 - \frac{\frac{3n(n-1)}{2}}{2^{n-1}n(n-1)} = 1 - \frac{3}{2^n} \quad (21)$$

□

Proposition 3: Given a fitness landscape, $FL = \{S, f, N\}$, where S is composed of binary strings with length n , $f = TRAP$, and the 1-bit flip operator is used to construct N ; then, the fraction of deceptive motifs in the corresponding directed fitness landscape network is equal to or greater than $1 - 3 \times 2^{-n}$.

Proof. Let a distance motif in the corresponding DFLN be $M^d = \{\{n_1, n_2, n_3\}, \vec{E}_{M^d}, \{d_1, d_2, d_3\}\}$, and the configurations corresponding to n_1, n_2, n_3 are not the global optimum, namely the string of all '0'. Let the configuration corresponding to $n_i, i = 1, 2, 3$, be $(s_{i1}, s_{i2}, \dots, s_{in})$, Then

$$f(n_i) = \sum_{j=1}^n s_{ij} \quad (22)$$

In fact, this is equivalent to *ONEMAX*, and according to the proof for Proposition 1, we have that M^d belongs to **Group**₃, **Group**₄, or **Group**₅.

Case 1: M^d belongs to **Group**₃. According to (14), we obtained that the number of '1' in n_1 is smaller than that of n_2 , and that of n_2 is smaller than that of n_3 . Since the global optimum is the string of all '0', we have $d_1 < d_2 < d_3$; that is, M^d is a core deceptive motif.

Case 2: M^d belongs to **Group**₄. According to (15), we obtained that the number of '1' in n_2 is smaller than those of n_1 and n_3 . Thus, we have $d_2 < d_1$ and $d_2 < d_3$; that is, M^d is a deceptive motif.

Case 3: M^d belongs to **Group**₅. According to (16), we obtained that both the number of '1' in n_1 and n_3 is larger than that of n_2 . Thus, we have $d_2 > d_1$ and $d_2 > d_3$; that is, M^d is a deceptive motif.

That is to say, only when a motif includes a node corresponding to the global optimum, can it be a guide or neutral motif. Then, being similar to the proof for Proposition 2, we have that the number of deceptive motifs is equal to or greater than $1 - \frac{3}{2^n}$. $\square$

The above propositions for *ONEMAX* and *TRAP* validate our knowledge that the more the number of guide motifs is, the easier the problem is, and the more the number of deceptive motifs is, the more difficult the problem is. Although most motifs in *NIAH* are neutral; that is, neither guide nor deceptive information exists, *NIAH* presents a level of difficulty for black-box algorithms (Droste et al., 2006).

5.1.2 Effect of the Spatial Distribution of Motifs on Problem Difficulty

In the above three fitness functions, DFLNs are dominated by one class of distance motifs. Next, two well-studied fitness functions, *LEADING_ONES* and *RIDGE* (Jansen, 2001) are used to show a different case; that is, DFLNs are composed of more than one class of distance motifs and their spatial distribution has a significant effect on problem difficulty.

$$LEADING_ONES(s) = \sum_{i=1}^n \prod_{j=1}^i s_j \quad (23)$$

$$RIDGE(s) = \begin{cases} n + 1 + \sum_{i=1}^n s_i & \text{If } \exists i \in \{0, 1, \dots, n\}, \text{ s.t., } s = 1^{(i)}0^{(n-i)} \\ n - \sum_{i=1}^n s_i & \text{Otherwise} \end{cases} \quad (24)$$

where $1^{(i)}$ denotes ‘11...1’ and $0^{(i)}$ is similar.

For *LEADING_ONES*, if $s_1 = 0$, then no matter what the values of the later $n - 1$ bits are, the fitness value is 0. For the search space with dimension n , the number of such nodes is 2^{n-1} , and all distance motifs composed of such nodes are neutral motifs. However, guide motifs also exist and connect to most nodes.

Proposition 4: Given a fitness landscape, $FL = \{S, f, N\}$, where S is composed of binary strings with length $n \geq 2$, $f = LEADING_ONES$, and the operator used in N is the 1-bit flip operator; then, each node whose Hamming distance to the global optimum is no less than 2 belongs to at least one core guide motif.

Proof. Let $s^* = (1, 1, \dots, 1)$ be the global optimum, and $s_1 = (s_{11}, s_{12}, \dots, s_{1n})$ be a candidate solution whose Hamming distance to the global optimum is no less than 2. Therefore, there are at least 2 bits in s_1 whose values are 0. Without loss of generality, let $s_{1i} = s_{1j} = 0$, $i, j \in \{1, 2, \dots, n\}$, $i < j$, and $s_{1k} = 1$ for $k < i$ and $i < k < j$. Let $s_2 = (s_{11}, s_{12}, \dots, s_{1(i-1)}, 1, s_{1(i+1)}, \dots, s_{1n})$ and $s_3 = (s_{11}, s_{12}, \dots, s_{1(i-1)}, 1, s_{1(i+1)}, \dots, s_{1(j-1)}, 1, s_{1(j+1)}, \dots, s_{1n})$. Then, we have

$$H(s_1, s_2) = H(s_2, s_3) = 1 \quad (25)$$

$$f(s_3) \geq j > j - 1 = f(s_2) > i - 1 = f(s_1) \quad (26)$$

Let n_1, n_2 , and n_3 be the nodes corresponding to s_1, s_2 , and s_3 , and d_1, d_2 , and d_3 be the distance of the three nodes to the global optimum. We have $d_2 = d_1 + 1$, $d_3 = d_2 + 1$; that is, $d_3 > d_2 > d_1$. Therefore, these three nodes form a core guide motif. $\square$

For *RIDGE*, the situation is different. Both deceptive and guide motifs exist, but only small number of nodes can form guide motifs.

Proposition 5: Given a fitness landscape, $FL = \{S, f, N\}$, where S is composed of binary strings with length n , $f = RIDGE$, and the 1-bit flip operator is used to construct N . Let $\bar{S}$ be the set of configurations that satisfy the first condition in (24). (1) Only distance motifs containing at least one node corresponding to the configuration in $\bar{S}$ can be guide motifs; (2) Each node whose corresponding configuration is in $\bar{S}$ belongs to at least one core guide motif.

Proof. (1) Let M^d be a distance motif where the configurations corresponding to the three nodes are not in $\bar{S}$. Then, the fitness value of each node decreases as the number of '1' increases. According to the proof for Proposition 3, M^d is a deceptive motif. This proves that only when at least one node's configuration is in $\bar{S}$, M^d can be any type of motifs except a deceptive type.

(2) Let $\bar{S} = \{\bar{s}_0, \bar{s}_1, \bar{s}_2, \dots, \bar{s}_n\}$, and $\bar{s}_i = 1^{(i)}0^{(n-i)}$, $i = 0, 1, \dots, n$. Then, we have

$$H(\bar{s}_j, \bar{s}_{j+1}) = 1, j = 0, 1, \dots, n-1 \quad (27)$$

$$H(\bar{s}_i, \bar{s}^*) = n - i \Rightarrow d_0 > d_1 > \dots > d_n \quad (28)$$

where $d_0, d_1, \dots, d_n$ are the distance to the global optimum of nodes corresponding to $\bar{s}_i, i = 0, 1, \dots, n$.

$$f(\bar{s}_i) = n + 1 + i \Rightarrow f(\bar{s}_0) < f(\bar{s}_1) < \dots < f(\bar{s}_n) \quad (29)$$

Therefore, each three connected nodes forms a core guide motif. $\square$

According to previous research (Jansen, 2001), *LEADING_ONES* is easy. Although Jansen (2001) also showed that *RIDGE* is an easy problem for a hill climber, Borenstein and Poli (2005b) showed that a simple GA with uniform crossover and mutation was unable to find the solution quickly. Propositions 4-5 show that although core guide motifs exist in both cases, their spatial distributions are completely different. For *LEADING_ONES*, core guide motifs nearly spread all over the whole networks while for *RIDGE*, they are only confined in a narrow space. This topological difference validates our experiences with the difficulty of these two functions to some extent.

5.2 Motif Difficulty

The most straightforward way to predict problem difficulty by motifs is to count the number of distance motifs. This also reflects the first aspect of the relationship between motifs and problem difficulty. Since this kind of difficulty measure is equivalent to making a statistic over the whole network, we name it as **Network Level Difficulty**.

Definition 8: For a directed fitness landscape network, $\vec{FLN} = \{V, \vec{E}\}$, its **Network Level Difficulty**, labelled as $MD_{\vec{FLN}}^e$, is defined as

$$MD_{\vec{FLN}}^e = \frac{\sum_{i=1}^{|V|-2} \sum_{j=i+1}^{|V|-1} \sum_{k=j+1}^{|V|} (M^D(n_i, n_j, n_k) - M^G(n_i, n_j, n_k))}{\sum_{i=1}^{|V|-2} \sum_{j=i+1}^{|V|-1} \sum_{k=j+1}^{|V|} M^d(n_i, n_j, n_k)} \quad (30)$$

where $n_i, n_j, n_k \in V$. $M^D(\cdot, \cdot, \cdot)$, $M^G(\cdot, \cdot, \cdot)$, and $M^d(\cdot, \cdot, \cdot)$ represent the number of deceptive, guide, and distance motifs that the three nodes can form, respectively.

The value of network level difficulty is in the range of $[-1.0, 1.0]$. When MD^e is equal to -1.0 , all distance motifs are guide ones, so the problem is the easiest. On

the contrary, when MD^e is equal to 1.0, all are deceptive motifs, so the problem is the most difficult. To reflect the spatial distribution of each class of distance motifs in a predictive measure, we need to analyse how the nodes are involved in each class of distance motifs. Thus, we classify nodes into two different types.

Definition 9: For a node n , if one of the distance motifs that n is a part of is a core guide motif, moreover, n is not the middle node, and the fitness value of n is the smallest, then n is a **Core Guide Node**. If one of the distance motifs that n is a part of is a core deceptive motif, moreover, n is not the middle node, and the fitness value of n is the largest, then n is a **Core Deceptive Node**.

By intuition, any node that is a part of guide motifs is helpful for leading the searching process in a right direction, and any node that is a part of deceptive motifs is harmful. However, after further analysing motifs, we find that the three nodes play different roles. For example, for core guide motifs in **Group₃**, only n_1 is a core guide node. Because when the searching process visits this motif from n_1 to n_3 , it is heading towards the right direction. But when the searching process reaches n_3 , other motifs that n_3 is a part of may be deceptive ones, and the searching process cannot continuously head towards the right direction. Thus, we cannot determine whether the contribution of n_3 is positive or negative. Based on core guide and deceptive nodes, we design another measure, namely **Node Level Difficulty**.

Definition 10: For a directed fitness landscape network, $\overrightarrow{FLN} = (V, \overrightarrow{E})$, V_{CGN} is the set of core guide nodes, and V_{CDN} is the set of core deceptive nodes. Then, **Node Level Difficulty**, labelled as $MD^o_{\overrightarrow{FLN}}$, is defined as follows.

$$MD^o_{\overrightarrow{FLN}} = \frac{|(V - V_{CGN}) \cap V_{CDN}| - |V_{CGN}|}{|V|} \quad (31)$$

Here, $(V - V_{CGN}) \cap V_{CDN}$ denotes the set of core deceptive nodes which are not core guide nodes. According to Definition 9, any node can be a core guide or a core deceptive node simultaneously. However, the most important characteristic of the searching process of EAs is that it works under selection pressure. How selection pressure works is reflected by which kinds of paths are chosen. Thus, under selection pressure, guide motifs always have a priority since the high fitness region is always preferred. Even if a node is a part of both guide and deceptive motifs, the searching process may not visit deceptive motifs at all. This trend is reflected in node level difficulty by giving core guide nodes a priority. That is, once a node is a core guide node, its effect is taking into account, and only when a node is a core deceptive node but not a core guide node, its effect can be taken into account. The value of node level difficulty is still in the range of $[-1.0, 1.0]$.

Since MD^e is a statistic over the whole network, it can predict problem difficulty in general, but cannot reflect the detailed difference. Clearly, this detailed difference can be reflected by MD^o , since it takes into account the situation of each node. Therefore, we integrate MD^e and MD^o together to form a new measure, namely **Motif Difficulty**, whose value is still in the range of $[-1.0, 1.0]$, where -1.0 corresponds to the easiest problems while 1.0 to the most difficult ones.

Definition 11: For a directed fitness landscape network, $\overrightarrow{FLN}$, **Motif Difficulty**, labelled as $MD_{\overrightarrow{FLN}}$, is defined as follows.

$$MD_{\overrightarrow{FLN}} = \frac{MD^e_{\overrightarrow{FLN}} + MD^o_{\overrightarrow{FLN}}}{2} \quad (32)$$

6 Experiments on Exact Motif Difficulty

If a measure is calculated on the whole search space, it is said to be exact (Naudts and Kallel, 2000). Thus, the value of motif difficulty is calculated exhaustively on the whole search space in this section. First, a collection of reasonably well-studied artificial fitness functions of known difficulty with different characteristics and a non-artificial problem, namely *MAXSAT*, are used to validate MD's prediction under various situations. Then, MD is used to estimate the difficulty of three counterexamples for other difficulty measures. Finally, counterexamples for MD are discussed to show the limitation of MD. The problems' search spaces are binary strings with length 16. The FLNs are built using the 1-bit flip operator.

To make a fair comparison with other measures, we also implemented FDC (Jones and Forrest, 1995) and Information Landscape (IL) (Borenstein and Poli, 2005b) under the same conditions. Furthermore, to validate MD's prediction of EAs' behaviour, three types of EAs and a Local Search algorithm (LS) are used to observe the hardness of these fitness functions. The first EA we used is (1+1)EA (Droste et al., 2002), which is widely used in analysing the behaviour of EAs theoretically. The second EA is a GA with the $1/n$ mutation only, labelled as GA_M. The third EA is a GA with both the uniform crossover and the $1/n$ mutation, labelled as GA_CM. Both GA_M and GA_CM use binary tournament selection and have 100 population size. LS is the same as (1+1)EA except the 1-bit flip operator is used instead. For these four search algorithms, the maximum number of fitness function evaluations (NFFEs) is set to 10^5 . The NFFEs used to find the global optimum is used as the performance measure. If the global optimum is not found within the maximum NFFEs, then the performance is set to the maximum NFFEs. All results are averaged over 1000 independent runs.

6.1 Well-Studied Fitness Functions

ONEMAX, *LEADING_ONES*, and two other fitness functions with a variable level of difficulty are used. The first *TRAP_i* ($i = 1, 2, \dots, n$) (Borenstein, 2008) is an extension of *TRAP* (Droste et al., 2002) we used in Section 5.1.

$$TRAP_i(s) = \begin{cases} 2n & \text{If } \#1(s) = 0 \\ n - \#1(s) & \text{If } \#1(s) < i \\ \#1(s) & \text{If } \#1(s) \geq i \end{cases} \quad (33)$$

The second *MULTI_MODAL_i* (Borenstein, 2008) is a multi-modal function with a varying number of local optima. For *MULTI_MODAL_i*, the global optimum is first selected uniformly at random from the search space, and its fitness value is set to $2n$. Then, i different local optima are selected uniformly at random from the search space, and their fitness values are set to n . Finally, the fitness values of all other configurations are calculated as n minus the distance to the closest local or global optimum.

Experimental results for *ONEMAX* and *LEADING_ONES* are given in Table 2. As can be seen, the performance of four algorithms is consistent with the prediction of three measures. Both the performance of algorithms and the predictive measures show that *ONEMAX* is extremely easy, and *LEADING_ONES* is more difficult than *ONEMAX*, but still belongs to easy problems since MD is smaller than -0.5 .

MULTI_MODAL_i is selected to test how MD reflects the effect of multimodality on problem difficulty. The number of local optima varies from 1 to 30. Values of three predictive measures are given in Figure 2(a). There is no clear decreasing or increasing trend on FDC, IL, and MD with the number of local optima. The performance of four

Table 2: The values of three predictive measures and the averaged NFFEs of four search algorithms for *ONEMAX*, *LEADING_ONES*, *RIDGE*, *NIAH*, and *RAND*.

Fitness Functions	<i>ONEMAX</i>	<i>LEADING_ONES</i>	<i>RIDGE</i>	<i>NIAH</i>	<i>RAND</i>
FDC	-1.0000	-0.3535	0.9877	-0.0156	0.0038
IL	0	0.3813	0.8599	0.4300	0.5012
MD	-0.9999	-0.6167	0.9971	-0.0009	-0.1110
(1+1)EA	9	215	668	64153	98880
GA_M	1643	4401	98228	61125	57695
GA_MC	826	3820	99700	50796	51577
LS	43	129	295	53140	100000

search algorithms are given in Figure 2(d), in which we can see that the computational costs of GA_M and GA_MC have no explicit relationship with the number of local optima. Therefore, MD confirms the previous observations regarding GA hardness; that is, multimodality is neither necessary nor sufficient for a landscape to be difficult to search (Kallel et al., 2001).

To further quantitatively evaluate the predictive capability of MD, we calculate the correlation coefficient between values of three measures and the average NFFEs of four search algorithms for *MULTI_MODAL_i*, which is shown in Table 3. For (1+1)EA and LS, MD shows a high correlation, which reaches to 0.85, and is higher than those of FDC and IL. For GA_M, although the correlation coefficient of MD is lower than those of FDC and IL, it is still larger than 0.7, and shows a positive correlation. For GA_MC, these three measures show a low correlation. This is mainly due to the fact that all of them do not take into account the effect of the crossover operator.

TRAP_i is selected to test whether MD can reflect gradual changes in problem difficulty. When i varies from 1 to n , problem difficulty decreases gradually. In fact, this reflects the effect of isolation on problem difficulty; that is, the more the global optimum is isolated from the other high fitness solutions, the more difficult the problem is. Values of three predictive measures are given in Figure 2(b). Clearly, three measures reflect gradual changes in problem difficulty. Therefore, MD not only confirms the previous results about the effect of isolation on GA hardness, but also shows an ability in reflecting gradual changes of problem difficulty. The performance of four search algorithms is given in Figure 2(e), which shows the same phenomenon. The correlation coefficient between the values of three measures and the average NFFEs of four search algorithms is shown in Table 3. For the four search algorithms, these three measures show a relatively high correlation, and MD even reaches to 0.93 for (1+1) EA and LS.

6.2 MAXSAT Problems

To test MD on landscapes of non-artificial problems, 10 random 3-MAXSAT instances are considered as well. Each instance with 16 literals and 68 clauses is generated by choosing for each clause three different literals uniformly at random and then negating each of them with probability 0.5, and only one global optimum is considered; that is, if an instance has more than one global optimum, one is selected uniformly at random. What should be noted is the ratio of clauses to literals here is 4.25, which means the generated instances lie in the phase transition region. In such a case, the probability that an instance is satisfiable is neither near one nor near zero. Values of three predictive measures are given in Figure 2(c).

Figure 2: The values of three predictive measures and the performance of four algorithms for Multi-Modal, Trap, and MAXSAT problems.

Although 10 instances are generated with the same number of literals and clauses and the same literal clause ratio, their difficulty varies from -0.5976 to -0.4520 on MD, from -0.4499 to -0.1247 on FDC, and from 0.3458 to 0.4573 on IL. The three measures show that Instances 6 and 7 are more difficult than others. This also confirms the previous results (Jansen, 2001) that different instances of the same problem might have different degrees of difficulty in black-box scenarios. The performance of four search algorithms is given in Figure 2(f), in which we can see that the performance of four algorithms also fluctuates from an instance to another, and all show a higher computational cost on Instances 6 and 7. The correlation coefficient between the values of three measures and the average NFFEs of four search algorithms is shown in Table 3. For GA_M and GA_MC, the results of FDC, IL, and MD are all above 0.55, and for (1+1)EA and LS, the results are smaller, but still larger than 0.33. Thus, a positive correlation is shown for the four algorithms.

The above experimental results show that MD is consistent with FDC and IL on both artificial and non-artificial problems, and confirm previous results on the problem difficulty, including isolation, deception, and multimodality.

6.3 Counterexamples for Other Difficulty Measures

Three classes of problems, namely Scaling, Constantness, and Irrelevant Deception problems, are used here. It has been shown that other difficulty measures fail to predict the difficulty of these three classes of problems. Epistasis measures, including epistasis variance and epistasis correlation, cannot correctly predict the difficulty of Constantness and Scaling problems. Although FDC can detect the presence of constantness in landscapes, FDC can be blinded by the presence of a large proportion of irrelevant deception since its construction is based on averaging (Naudts and Kallel, 2000).

Table 3: The correlation coefficient between three predictive measures and the performance of four search algorithms.

Problems	Measures	(1+1)EA	GA_M	GA_MC	LS
Multi-Modal	FDC	0.8139	0.7758	0.2921	0.7959
	IL	0.8481	0.7353	0.2154	0.8395
	MD	0.8580	0.7133	0.1800	0.8516
Trap	FDC	0.9261	0.8440	0.8447	0.9171
	IL	0.9369	0.8185	0.8756	0.9336
	MD	0.9345	0.8257	0.8738	0.9356
MAXSAT	FDC	0.4075	0.6204	0.6653	0.4059
	IL	0.4271	0.6195	0.6613	0.4178
	MD	0.3357	0.5571	0.6202	0.3992
Scaling	FDC	-0.1501	-0.1632	0.2688	0.3439
	IL	0	0	0	0
	MD	0	0	0	0
Constantness	FDC	0.5363	0.6151	0.8113	0.5313
	IL	0.3914	0.4690	0.6795	0.3866
	MD	0.5393	0.6225	0.8498	0.5339
Irrelevant Deceptive	FDC	-0.6855	-0.7809	-0.8200	-0.7412
	IL	0.4556	0.5227	0.5588	0.5063
	MD	0.5835	0.6697	0.7103	0.6397

6.3.1 Scaling Problems

Naudts and Kallel (2000) constructed a scaling problem, namely *NLM*, as follows.

$$NLM_m(s) = \left(\frac{\#1(s)}{n} \right)^m \quad (34)$$

Clearly, the larger the value of m is, the larger the fitness value of the global optimum, namely the string of all '1', gets isolated from the mass of fitness values. In fact, this class of functions is a modified version of *ONEMAX*. For a GA with comparison-based selection, the difficulty of this class of functions does not change with m . The values of three predictive measures for this function are given in Figure 3(a). Both MD and IL are stable, and equal to -0.9999 and 0 , respectively. However, FDC increases from -1.0000 to -0.0738 , which confirms the results in Naudts and Kallel (2000); that is, FDC goes to 0 for both m and n going to infinity. The performance of four search algorithms is given in Figure 3(d) in which all algorithms show a stable performance similar to that of *ONEMAX* for all m . Clearly, the values of MD and IL are consistent with the performance of four search algorithms. Therefore, MD has the advantage of being insensitive to nonlinear scaling.

The correlation coefficient between the values of three measures and the average NFFEs of four algorithms is shown in Table 3. Since no clear changes in both values of IL and MD and the performance of four algorithms, the correlation coefficient of IL and MD is 0 , while that of FDC fluctuates from negative to positive, which shows a failure in predicting the difficulty of this problem.

6.3.2 Constantness Problems

Naudts and Kallel (2000) constructed a generalized Royal Road function (Mitchell et al.,

Figure 3: The values of three predictive measures and the performance of four algorithms for Scaling, Constantness, and Irrelevant Deception problems.

1993) as follows.

$$GENERALIZED_ROYAL_ROAD_{m,m'}(s) = \sum_{k=m}^{m'} 2^k \left| \left\{ i | s \in H_i^{\log n, k} \right\} \right| \quad (35)$$

where $0 \leq m \leq \log n$, $0 \leq i < 2^{(\log n)-m}$, $m \leq m' \leq \log n$, and

$$H_i^{\log n, k} = *(2^k i)_1 (2^k) * (2^{\log n} - 2^k (i+1)) \quad (36)$$

where “*” stands for a wild-card symbol, which can be ‘0’ or ‘1’. The difficulty of this function is controlled by m and m' , and functions with the same m have the same difficulty. Here, since $\log n = 4$, experiments on all 15 pairs of (m, m') are conducted, and the values of three predictive measures are given in Figure 3(b), where the 15 pairs of parameters are ordered from easy to difficult (1 - 15) as follows on the X-axis: (0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4).

As can be seen, MD perfectly reflects the difficulty changes; that is, for $m=0, 1, 2, 3, 4$, the value of MD is constant for each level. Thus, MD can detect the presence of constantness in landscapes. FDC and IL also show this change in general. However, for $m = 0$ and 1 , FDC and IL increase with m' clearly while MD is stable in each level. The performance of four algorithms is given in Figure 3(e), which shows that the functions with former 12 pairs of parameters are relatively easy. MD is -0.8749 for $m = 1$ and -0.6025 for $m = 2$, which also indicate that the former 12 functions are easy.

The correlation coefficient between the values of three measures and the average NFEs of four algorithms is shown in Table 3. The correlation coefficient of FDC and MD for four search algorithms is similar, and in the range of $[0.5, 0.85]$. Moreover, that of MD is always higher than that of FDC. The correlation coefficient of IL is low, which is only about 0.4 for (1+1)EA, GA_M, and LS, and 0.6795 for GA_MC while those of

both FDC and MD are higher than 0.8.

6.3.3 Irrelevant Deception Problems

Naudts and Kallel (2000) constructed a problem with irrelevant deception, namely *DECEPTIVE_MIXTURE*, by mixing a *ONEMAX* and a number of *ZEROMAX*.

$$DECEPTIVE_MIXTURE_m(s) = \begin{cases} n - 1 - \sum_{i=2}^n (1 - s_i), & \text{If } s_1 = 0 \\ n - 2 - \sum_{i=3}^n (1 - s_i) + 2n, & \text{If } s_1 = 1 \text{ and } s_2 = 0 \\ \dots\dots\dots \\ n - m - \sum_{i=m+1}^n (1 - s_i) + 2(m-1)n, & \text{If } s_1 = s_2 = \dots = s_{m-1} = 1 \text{ and } s_m = 0 \\ \sum_{i=m+1}^n s_i + 2mn, & \text{If } s_1 = s_2 = \dots = s_m = 1 \end{cases} \quad (37)$$

The higher the mixture coefficient m is, the harder the problem is. But both FDC and sitewise optimisation measure (a generalisation for FDC and epistasis) (Naudts and Kallel, 2000) failed to correctly predict the performance for this function. Here, experiments on $m=1$ (easy), $2, \dots, n-1$ (difficult) are conducted, and the values of three predictive measures are given in Figure 3(c). As can be seen, the value of FDC decreases with m , which is in contradiction with the characteristics of this class of functions. For IL and MD, both of their values increase with m . The values of MD varies from -0.5311 to -0.0749 , which are always smaller than 0 and indicate that this class of functions is not very difficult for EAs. The performance of four search algorithms is given in Figure 3(f) in which all results show that although the difficulty of this class of functions increases with m , they are easy in general, and EAs can solve this class of functions easily. This confirms the analysis in (Naudts and Kallel, 2000) and the deceptive information in this class of functions is irrelevant to the searching process of EAs. Thus, MD can distinguish irrelevant information existing in landscapes.

The correlation coefficient between the values of three measures and the average NFFEs of four algorithms is shown in Table 3. Obviously, FDC shows a negative correlation. Although the above results show that both IL and MD correctly predict the changing trend of the difficulty, the correlation coefficient shows that MD is more accurate than IL since the values of the correlation coefficient for MD are in the range of $[0.58, 0.71]$, and are consistently larger than those of IL by 0.1.

The above experimental results show that FDC fails in detecting irrelevant deceptive information and is sensitive to nonlinear scaling. Although IL can detect irrelevant deceptive information and is insensitive to nonlinear scaling, IL is not as stable as MD in detecting the presence of constantness, and the correlation coefficient between the values of IL and the performance of search algorithms is much lower than that of MD. Moreover, in predicting Irrelevant Deceptive problems, MD is always more accurate than IL. In general, MD performs very well on the three classes of counterexamples for other difficulty measures, and is insensitive to nonlinear scaling, which is consistent with GA, can detect the presence of constantness, and is robust to irrelevant deceptive information.

6.4 Counterexamples for Motif Difficulty

In addition to the success that MD achieved in the above experiments, MD still has limitations. In this section, three counterexamples for MD, namely *RIDGE*, *NIAH*, and *RAND*, are presented. The global optima of *NIAH* and *RAND* are selected uniformly at random from the search space. For *RAND*, the fitness value of the global optimum

is set to 2^n while those of all other configurations are chosen uniformly at random from $[1, 2^n - 1]$. The values of three predictive measures and the performance of four search algorithms are given in Table 2.

The performance of four algorithms on *RIDGE* confirms previous discussion (Jansen, 2001; Borenstein, 2008); that is, *RIDGE* is an easy problem for a hill climber, but a simple GA with uniform crossover and mutation has difficulty in solving it. However, MD (0.9971) indicates that *RIDGE* is an extremely difficult problem, which fails in predicting the performance of (1+1)EA and LS. As we discussed in Section 5.1, the number of guide motifs in the DFLN for *RIDGE* is relatively small and distributes only in a narrow space, so it shows difficulty in both node level and network level difficulty, and naturally shows difficulty in MD since MD is actually a combination of the two statistics. The reason that GA fails in solving *RIDGE* is mostly due to the fact that binary tournament selection regularly loses individuals from the ridge. Thus, the right prediction of MD on the performance of GA_M and GA_MC is mainly due to the fact that the population size is large here. *RIDGE* clearly exposes a limitation of MD; that is, when a class of distance motif in a DFLN is small in amount and narrow in spatial distribution, MD may not predict its difficulty correctly for (1+1)EA, local search or GA with small populations since MD is based on two statistics over the whole network.

For *NIAH* and *RAND*, MD shows that they have similar difficulty, which is neither easy nor difficult. However, Droste et al. (2006) already showed that *NIAH* is extremely difficult for black-box algorithms. As we discussed in Section 5.1, most motifs in *NIAH* are neutral; that is, neither guide nor deceptive information exists. In fact, when the search space is large, the lack of guide information presents a fatal problem for black-box algorithms. Although the amount of guide motifs is similar to the amount of deceptive motifs for *RAND*, they distribute randomly, and no explicit guide or deceptive information presents at all. Thus, another limitation of MD is the prediction on the situation that neither guide nor deceptive information presents in an explicit way.

7 Experiments on Approximate Motif Difficulty

If a measure is computed on a sample of the search space, its values are called approximate (Naudts and Kallel, 2000). Since exhaustive computation on whole networks becomes quickly impractical, good sampling techniques on networks are required. In this section, we first propose a sampling technique for calculating the approximate MD, and then give the approximate MD for problems in Section 6. Additionally, since the search spaces of all problems in Section 6 are binary, MKPs are also used to validate the performance of MD on FLNs with different topologies.

7.1 Sampling Technique for Approximate MD

First, a subset of different nodes, V_s , is selected randomly from V . Then, for any pair of nodes, (x, y) in V_s , a sequence of nodes, $z_1, z_2, \dots, z_n \in V$ exists which satisfies

$$z_1 \in N(x), z_2 \in N(z_1), z_3 \in N(z_2), \dots, z_n \in N(z_{n-1}), y \in N(z_n), n \geq 0 \quad (38)$$

Then, the set of distance motifs, $M_{(x, y)}^d$ in this sequence is

$$M_{(x, y)}^d = \{(x, z_1, z_2), (z_2, z_3, z_4), \dots, (z_{n-2}, z_{n-1}, z_n), (z_{n-1}, z_n, y)\} \quad (39)$$

The sets of guide and deceptive motifs in this sequence are labelled as $M_{(x, y)}^G$

Table 4: Approximate MD for the five fitness functions in Section 6.

Fitness Functions	<i>ONEMAX</i>	<i>LEADING_ONES</i>	<i>NIAH</i>	<i>RAND</i>	<i>RIDGE</i>
MD	-0.9999	-0.6167	-0.0009	-0.1110	0.9971
$\widetilde{MD}$	-0.9997	-0.5198	0	-0.2303	0.9998
$\pm$ Standard	± 1.17	± 1.21	± 0	± 1.45	± 6.67
Deviation	$\times 10^{-4}$	$\times 10^{-5}$		$\times 10^{-2}$	$\times 10^{-5}$

and $M_{(x,y)}^D$, respectively. Then, the approximate network level difficulty, $\widetilde{MD}^e$, is calculated as follows.

$$\widetilde{MD}^e = \frac{\sum_{x,y \in V_s} |M_{(x,y)}^D| - \sum_{x,y \in V_s} |M_{(x,y)}^G|}{\sum_{x,y \in V_s} |M_{(x,y)}^d|} \quad (40)$$

In $M_{(x,y)}^d$, (x, z_1, z_2) and (z_{n-1}, z_n, y) are called as End Distance Motifs. Thus, any node x in V_s is a core guide node over V_s if one of the end distance motif that x is a part of belongs to core guide motifs. The definition of a core deceptive node over V_s is similar. Then, the approximate node level difficulty, $\widetilde{MD}^o$, is calculated as follows.

$$\widetilde{MD}^o = \frac{|(V_s - V_s^{CGN}) \cap V_s^{CDN}| - |V_s^{CGN}|}{|V_s|} \quad (41)$$

Finally, the approximate motif difficulty, $\widetilde{MD}$, is calculated as follows.

$$\widetilde{MD} = \frac{\widetilde{MD}^e + \widetilde{MD}^o}{2} \quad (42)$$

Clearly, the computational cost of calculating $\widetilde{MD}$ lies in finding the sequence $z_1, z_2, \dots, z_n$, for any pair of nodes in V_s . Since this sequence depends on the neighbourhood structure, the method to find it depends on the operator. For the 1-bit flip operator, the simplest method is to find the bits with different values in the pair first, and changes the value one by one. For the swap operator under permutation encoding, one of the method is to do the following operations for each position with different values in the pair until the two nodes are the same. Suppose $x = (x_1, x_2, \dots, x_n)$ and $y = (y_1, y_2, \dots, y_n)$, and $x_i \neq y_i$. If $x_j = y_i$, then swap x_i and x_j . What should be also noted is for a pair of nodes, this sequence may not be unique, and we need to use only one for the sake of the sampling technique.

7.2 Experiments on Approximate MD

In the following experiments, the size of the search space is 32, $|V_s|$ is 1000, and the values of approximate MD are averaged over 10 independent samplings. Table 4 lists average approximate MD and standard deviations for the five fitness functions in Section 6. As can be seen, approximate MD is consistent with MD for these five functions. Moreover, the standard deviations are quite small.

To further illustrate the performance of approximate MD on testing gradual changes in difficulty, $TRAP_i$ and Constantness problems are used, where i varies from 1 to 32, and the X-axis of Constantness problems is on all 21 pairs of (m, m') ; that is, $(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 2), (2, 3), (2, 4),$

Figure 4: Average approximate MD and standard deviations (in the form of error bars).

(2, 5), (3, 3), (3, 4), (3, 5), (4, 4), (4, 5), (5, 5). Experimental results are shown in Figure 4, which confirms that the sampling technique can not only reflect gradual changes in difficulty, but also detect the presence of constantness in landscapes. Moreover, the standard deviations are very small, which illustrates that approximate MD is stable.

7.3 Multidimensional Knapsack Problems

To validate the performance of MD on FLNs with different topologies, we analyse the effect of two representations on the difficulty of MKPs. The first is the binary representation, where each candidate solution is represented by a characteristic bit vector and each bit is mapped to an item. A bit set to 1 indicates that the corresponding item is packed into the knapsack. For this representation, we use the 1-bit flip operator to build FLNs, and the fitness of each configuration is calculated by the penalty function (*pCOR*) in (Gottlieb, 2001).

The second is the permutation representation, where each configuration is represented by a permutation of all items $\pi = (\pi_1, \pi_2, \dots, \pi_n)$. For this representation, we use the swap operator to build the FLNs; that is, two different items are first selected from π and then swapped. Since the global optimum is represented as a binary string and the computational cost to identify the permutations corresponding the global optimum is high, the shortest path length cannot be used as the distance. However, the fitness of each configuration needs to be calculated by decoding the permutation into a binary string using a first-fit heuristic (Tavares et al., 2008), so we use the Hamming distance for the corresponding decoded binary string to form distance motifs. This is also the method used to calculate the distance in (Tavares et al., 2008). The MKPs are selected from OR-Library¹. Experimental results are shown in Table 5.

As can be seen, the values of approximate MD for the permutation representation are always smaller than those for the binary representation. This confirms the results in (Tavares et al., 2008); that is, the permutation representation with a first-hit heuristic is better than the binary representation. To further illustrate the difference between these two representations, (1+1)EA and GA_M are used to solve these problems. For the permutation representation, the $1/n$ mutation operator is replaced by the swap operator. The maximum NFFEs is set to 5×10^5 , and other parameters are the same as those of the above experiments. The results are given in Table 5. For the five problems, the average NFFEs for the permutation representation is always smaller than that for the binary representation, which is consistent with the results of approximate MD.

¹<http://people.brunel.ac.uk/~mastijb/jeb/info.html>

Table 5: Approximate MD and averaged NFFEs of search algorithms for MKPs (n =the number of items and m =the number of knapsacks).

Instance			Representation	$\widehat{MD}$ $\pm$ Standard Deviation	(1+1)EA	GA_M
No.	n	m				
1	28	10	Binary	$-0.3580 \pm 1.11 \times 10^{-4}$	219093	391745
			Permutation	$-0.5819 \pm 3.64 \times 10^{-4}$	1011	3147
2	39	5	Binary	$-0.4025 \pm 1.20 \times 10^{-3}$	499631	499917
			Permutation	$-0.5329 \pm 2.79 \times 10^{-4}$	265997	247055
3	50	5	Binary	$-0.4047 \pm 6.30 \times 10^{-4}$	497970	500000
			Permutation	$-0.5436 \pm 1.09 \times 10^{-4}$	72116	188585
4	60	5	Binary	$-0.4256 \pm 3.19 \times 10^{-3}$	225695	500000
			Permutation	$-0.7548 \pm 5.85 \times 10^{-4}$	145309	398969
5	60	30	Binary	$-0.5321 \pm 5.86 \times 10^{-4}$	484465	500000
			Permutation	$-0.6365 \pm 1.49 \times 10^{-3}$	147402	148265

8 Discussion and Conclusion

In this study, we have proposed a new predictive difficulty measure using network motifs. Based on properties of DFLNs, distance motifs were first designed, and three classes of distance motifs based on their contributions to the searching process of EAs were proposed. The new difficulty measure, namely MD, was defined by synthesising the features of three classes of distance motifs, namely guide, deceptive, and neutral motifs. A sampling technique for computing approximate MD was also designed. Extensive experiments were conducted to validate the performance of both exact and approximate MD, and the experimental results have shown that MD is not only consistent with previous results on problem difficulty in terms of isolation, deception, multimodality, gradual changes of problem difficulty, and the effect of different representations on the difficulty of MKPs, but also worked very well for three counterexamples for other difficulty measures. MD manifested the advantages of being insensitive to nonlinear scaling, detecting the presence of constantness, and being robust to irrelevant deceptive information.

The comparison with two other measures, namely FDC and IL, also illustrated the good performance of MD. First, FDC failed in Scaling problems and Irrelevant Deceptive problems, while both MD and IL succeed. Second, although the performance of MD and IL is competitive, MD is more accurate than IL for Constantness problems and Irrelevant Deceptive problems since its correlation coefficient is much higher than that of IL. The limitations of MD were also discussed. The first one is due to the fact that MD is based on two statistics over the whole network, MD may fail in predicting the performance of (1+1)EA or local search or GA with small populations when a class of distance motif in a DFLN is small in amount and narrow in spatial distribution. The second is the prediction of MD on the situation that neither guide nor deceptive information presents explicitly.

It is well-known that fitness landscapes can be related to networks, but few studies have been done to design predictive difficulty measures using network properties. Since network motifs are basic building blocks of complex networks, problem difficulty is expected to be related to these building blocks, and our work confirmed this. In fact, network motifs can be seen as the features of problems, and problem difficulty is deter-

mined by these features. Thus, to predict problem difficulty is equivalent to learning the relationship between these features and problem difficulty, which can be seen as a machine learning problem. This study represents our attempt towards finding such relationships in a simple way.

More work remains to be done. First, taking global optima as references is impractical for practical applications; thus, other reference points need to be investigated. Second, the current version of MD has constraints on the operator used to build FLNs. That is, the operator should be unary and small in the size of neighbourhood, and the probability of converting one node to different neighbours should be the same. Since edges in DFLNs are created using only the relative difference between fitness values of two end nodes, MD is suitable only for analysing comparison-based search algorithms that use the operator satisfying the above conditions. Another drawback of MD is that the ideal distance, namely the shortest path length, may not work for some encoding methods, and we had to find other ways to calculate the distance. Moraglio and Poli (2004) and Moraglio (2007) assigned distances to search operators following the geometric framework, in which both mask-based crossovers and bit-wise mutations for binary strings are associated with the Hamming distance. This work is really useful in helping MD break through current limitations on operators and distance, and it is also one of our future works. To study problem difficulty is only a step towards the design of more efficient algorithms. We thus hope that MD will be useful in the design of efficient search techniques or operators.

Acknowledgment. This work is partially funded by an Australian ARC Discovery Project Grant. The authors would like to thank the reviewers for their helpful comments and valuable suggestions.

References

- Bäck, T., Hammel, U., and Schwefel, H.-P. (1997). Evolutionary computation: Comments on the history and current state. *IEEE Trans. Evolutionary Computation*, 1(1):3–17.
- Borenstein, Y. (2008). *Problem Hardness for Randomized Search Heuristics with Comparison-based Selection: A Focus on Evolutionary Algorithms*. PhD Thesis, University of Essex, U.K.
- Borenstein, Y. and Poli, R. (2005a). Information landscapes. In *Proceedings of 2005 Genetic and Evolutionary Computation Conference*, pages 1515–1522. ACM Press.
- Borenstein, Y. and Poli, R. (2005b). Information landscapes and problem hardness. In *Proceedings of 2005 Genetic and Evolutionary Computation Conference*, pages 1425–1431. ACM Press.
- Borenstein, Y. and Poli, R. (2005c). Information landscapes and the analysis of search algorithms. In *Proceedings of 2005 Genetic and Evolutionary Computation Conference*, pages 1287–1294. ACM Press.
- Davidor, Y. (1991). Epistasis variance: A viewpoint on ga-hardness. In Rawlins, G. J. E., editor, *Foundations of Genetic Algorithms*, pages 23–35, San Mateo, CA. Morgan Kaufmann.
- Droste, S., Jansen, T., and WegenerPeter, I. (2002). On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81.
- Droste, S., Jansen, T., and WegenerPeter, I. (2006). Upper and lower bounds for randomized search heuristics in black-box optimization. *Theory of Computing Systems*, 39:525–544.
- Forgel, D. (1999). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. New York: Wiley, 2 edition.

- Forrest, S. and Mitchell, M. (1993). Relative building block fitness and the building block hypothesis. In Whitley, D., editor, *Proceedings of Foundations of Genetic Algorithms*, pages 109–126, San Mateo, CA. Morgan Kaufmann.
- Ghoneim, A., Abbass, H. A., and Barlow, M. (2008). Characterizing game dynamics in two-player strategy games using network motifs. *IEEE Trans. Systems, Man, and Cybernetics, part B*, 38(3):682–690.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley: Reading, MA.
- Gottlieb, J. (2001). On the feasibility problem of penalty-based evolutionary algorithms for knapsack problems. In *Proceedings of EvoWorkshops Appl. Evol. Comput.*, pages 50–59.
- He, J., Reeves, C., Witt, C., and Yao, X. (2007). A note on problem difficulty measures in black-box optimization: Classification, realizations and predictability. *Evolutionary Computation*, 15(4):435–443.
- Jansen, T. (2001). On classifications of fitness functions. In Kallel, L., Naudts, B., and Rogers, A., editors, *Theoretical Aspects of Evolutionary Computing*, pages 371–385. Springer-Verlag.
- Jones, T. and Forrest, S. (1995). Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In Eshelman, L. J., editor, *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 184–192, San Mateo, CA. Morgan Kaufmann.
- Kallel, L., Naudts, B., and Reeves, C. R. (2001). Properties of fitness functions and search landscapes. In Kallel, L., Naudts, B., and Rogers, A., editors, *Theoretical Aspects of Evolutionary Computing*, pages 175–206. Springer-Verlag.
- Merz, P. and Freisleben, B. (2000). Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Trans. Evolutionary Computation*, 4(4):337–352.
- Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827.
- Mitchell, M., Forrest, S., and Holland, J. H. (1993). The royal road for genetic algorithms: Fitness landscapes and ga performance. In Varela, F. J. and Bourgine, P., editors, *Proceedings of the 1st European Conference on Artificial Life*, pages 245–254, Cambridge, MA. M.I.T Press/Bradford.
- Moraglio, A. (2007). *Towards a Geometric Unification of Evolutionary Algorithms*. PhD Thesis, University of Essex, U.K.
- Moraglio, A. and Poli, R. (2004). Topological interpretation of crossover. In *Proceedings of 2004 Genetic and Evolutionary Computation Conference*, pages 1377–1388.
- Naudts, B. (1998). *Measureing GA-hardness*. PhD Thesis, University of Antwerp, Belgium.
- Naudts, B. and Kallel, L. (2000). A comparison of predictive measures of problem difficulty in evolutionary algorithms. *IEEE Trans. Evolutionary Computation*, 4(1):1–15.
- Ochoa, G., Tomassini, M., Vérel, S., and Darabos, C. (2008). A study of nk landscapes’ basins and local optima networks. In *Proceedings of 2008 Genetic and Evolutionary Computation Conference*, pages 555–562. ACM Press.
- Tavares, J., Pereira, F. B., and Costa, E. (2008). Multidimensional knapsack problem: A fitness landscape analysis. *IEEE Trans. Systems, Man, and Cybernetics, part B*, 38(3):604–616.
- Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In *Proceedings of the 6th Congress on Genetics*, page 365.
- Yang, A., Abbass, H. A., and Sarker, R. (2006). Characterizing warfare in red teaming. *IEEE Trans. Systems, Man, and Cybernetics, part B*, 36(2):268–285.