Polynomial Time Summary Statistics for a Generalization of MAXSAT

Robert B. Heckendorn
Department of Computer Science
Colorado State University
Fort Collins, CO 80523 USA
heckendo@cs.colostate.edu
(970) 484-9903

Soraya Rana
Department of Computer Science
Colorado State University
Fort Collins, CO 80523 USA
rana@cs.colostate.edu
(970) 491-6453

Darrell Whitley
Department of Computer Science
Colorado State University
Fort Collins, CO 80523 USA
whitley@cs.colostate.edu
(970) 491-5373

Abstract

MAXSAT problems are notoriously difficult for genetic algorithms to solve. NK-landscapes are often used as test problems of scalable difficulty for genetic algorithms. In this paper we exploit the similar structure of the two problems to create an encompassing class of problems called embedded landscapes. Then we use Walsh analysis to explore the nonlinear bit interactions of these important test functions. We show that by applying Walsh analysis to embedded landscapes, several important summary statistics can be generated in polynomial time. We then use these techniques to discuss the statistical "shape" of both MAXSAT and NK-landscapes.

1 INTRODUCTION

MAXSAT problems are notoriously difficult for genetic algorithms to solve. Even relatively old algorithms such as Davis-Putnam [Davis and Putnam, 1960] which are deterministic and exact are orders of magnitude faster than GAs. Understanding what makes MAXSAT so difficult for GAs gives us important clues about mechanisms of problem difficulty in general. NK-landscapes, a class of problems used in theoretical biology[Kauffman, 1993], has become one of the classic problem classes used for testing genetic algorithms. One of the reasons for this is they provide tunably difficult terrain for testing GA performance. The similarity in structure between MAXSAT and NK-landscapes has led to the development of generalization of these two problems called **embedded landscapes** that contains both problem classes. This new class has applicability to a broad range of constraint satisfaction problems and problems based on nonlinear bit interactions known as epistasis.

Walsh analysis is often used to measure the nonlinear relationships between bits in the domain of discrete functions defined over bit strings. The heart of Walsh analysis is the Walsh transform which allows the quantification of all possible bit interactions. Unfortunately, the Walsh transform typically requires exponential time to compute with respect to the number of bits in the domain. However, recent work has shown that both MAXSAT and NK-landscapes have tractable Walsh analysis [Rana et al., 1998, Heckendorn and Whitley, 1997]. In fact, all embedded landscapes offer a polynomial time Walsh transform if the maximum number of variable interactions is bounded[Heckendorn et al., 1998].

We extend this work to show that Walsh analysis offers a polynomial time method for computing summary statistics for embedded landscapes and hence for both MAXSAT and NK-landscapes. We apply this technique to computing summary statistics for a set of randomly generated MAXSAT problems and discuss our results.

2 MAXSAT, NK-LANDSCAPES AND EMBEDDED LANDSCAPES

A SAT problem is stated as a logical expression in conjunctive normal form (CNF). The problem is considered solved when an instantiation of variables is found such that the expression is true or it can be proven that no such instantiation exists. SAT can be transformed into an optimization problem by summing the truth values (0 or 1) of the disjunctive clauses rather than ANDing them. This optimization counterpart to SAT is known as MAXSAT[Papadimitriou, 1994]. A version of MAXSAT which restricts the length of the clauses to have k literals is called MAXkSAT. The truth value

assignments for an N variable MAXSAT problem can be represented by a string of N bits. In this way SAT becomes the optimization problem of maximizing the N-bit function $f : \mathcal{B}^N \rightarrow \{0, 1, \ldots, C\}$ where C is the number of clauses in the CNF. For example: a 4-variable, 4-clause MAX2SAT problem might be:

$$f(x) = (x_0 \vee \overline{x_1}) + (x_1 \vee x_2) + (\overline{x_0} \vee \overline{x_2}) + (x_1 \vee x_3)$$

where the values of the 4 variables are assigned by the values of the bits in the bit string $x \in \mathcal{B}^4$. Throughout this paper we will assume the variables are labeled right to left in the bit string. In the example the bits of x would be associated with variables $x_3x_2x_1x_0$. Hence, for the sample variable assignment string, 0101: $x_3 = 0$, $x_2 = 1$, and so on. The value of $f(0101)$ is $1 + 1 + 0 + 0 = 2$.

In order to define MAXSAT problems more precisely, we first define two functions. Let $bc(i)$ be a **bit count** function which returns the number of 1's in i. Let **pack**:$\mathcal{B}^L \times \mathcal{B}^L \rightarrow \mathcal{B}^M$ where $M \leq L$ be a function. $pack(x, m)$ takes the bits in x and masks them (using AND) with an L bit mask $m : bc(m) = M$ and packs the bits selected by the mask in the result. For example: $pack(10101, 01101) \rightarrow 011$. The *pack* function will allow us to relate functions of different dimensions. We use the term **dimension** of a function to mean the number of bits in the domain.

The MAXSAT problem can now be expressed as:

$$f(x) = \sum_{i=1}^{C} c_i(pack(x, m_i))$$

where C is the number of clauses in the problem, $c_i : \mathcal{B}^{bc(m_i)} \rightarrow \mathcal{B}$ is the i^{th} disjunctive clause and m_i is the i^{th} mask selecting the variables needed by c_i. m_i is referred to as an **interaction mask**. In the specific 4-variable, 4-clause MAX2SAT example, the clauses are: $c_0 = (x_0 \vee \overline{x_1})$, $c_1 = (x_1 \vee x_2)$, $c_2 = (\overline{x_0} \vee \overline{x_2})$, $c_3 = (x_1 \vee x_3)$ and their corresponding masks are: $m_0 = 0011$, $m_1 = 0110$, $m_2 = 0101$, $m_3 = 1010$ respectively.

NK-landscapes are another important class of test problems for genetic algorithms. They derive from Stuart Kauffman's work in theoretical biology [Kauffman, 1993]. An NK-landscape, $f : \mathcal{B}^N \rightarrow \mathbb{R}$, can be expressed as:

$$f(x) = \frac{1}{N} \sum_{i=0}^{N-1} r_i(pack(x, m_i))$$

where N represents the number of the number of bits in the domain of the function. For each of the N bits, K other bits are chosen as influencing bits. Each of the N combinations of $K + 1$ bits (a bit at its K influencing bits) can be represented by an interaction mask m_i such that the i^{th} bit is set in m_i and $bc(m_i) = K + 1$. Each $r_i : \mathcal{B}^{K+1} \rightarrow [0, 1)$ represents a table of 2^{K+1} random values between 0 and 1. For example the following tables completely define the 4 subfunctions and associated masks for an example NK-landscape with $N = 4$, $K = 1$:

	$m_0 = 0011$	$m_1 = 0110$	$m_2 = 0101$	$m_3 = 1010$
y	$r_0(y)$	$r_1(y)$	$r_2(y)$	$r_3(y)$
00	0.3	0.1	0.9	0.4
01	0.7	0.2	0.3	0.6
10	0.2	0.5	0.7	0.9
11	0.6	0.8	0.4	0.1

So, for example, $f(0101) = \frac{1}{4}(0.7 + 0.5 + 0.4 + 0.4) = 0.3$.

NK-landscape [Kauffman, 1989] problems bare a striking resemblance to the general form of MAXSAT [Heckendorn et al., 1998, Weinberger, 1990]. Both problems are a summation of smaller **subfunctions** that are defined over subsets of bits selected by interaction masks. With this observation in mind we designed an encompassing class of problems called embedded landscapes. **Embedded landscapes**, $f : \mathcal{B}^N \rightarrow \mathbb{R}$, model a broad class of combinatorial and constraint satisfaction problems. They can be expressed as the sum of Q subfunctions;

$$f(x) = \sum_{j=1}^{Q} g_j(pack(x, m_j)).$$

There are no restrictions on the number of subfunctions, Q, or the number of 1 bits in each interaction mask, $m_j \in \mathcal{B}^N$, or the values returned by the **interaction functions** $g_j : \mathcal{B}^{bc(m_j)} \rightarrow \mathbb{R}$. The g_j generally have lower dimensional domains than f and are hence considered to be **embedded** in higher dimensional space. In many cases we only consider functions in which the maximum dimension of the g_j is bounded by some k. These problems form a difficult set of problems. In fact, the optimization problems posed by the set of embedded landscapes, with the subfunctions of bounded dimension of at least 3, have been shown to be NP-complete [Heckendorn et al., 1998].

2.1 WALSH ANALYSIS

The **Walsh transform** is the analog to the discrete Fourier transform but for functions whose domain is a bit string. Every real valued function f over an L-bit string, $f : \mathcal{B}^L \rightarrow \mathbb{R}$, can be expressed as a weighted sum of a set of 2^L orthogonal functions called **Walsh**

functions.

$$f(x) = \sum_{j=0}^{2^L-1} w_j \psi_j(x) \tag{1}$$

where the Walsh Functions are denoted $\psi_j : \mathcal{B}^L \rightarrow \{-1, 1\}$. The Walsh functions play the role that sine and cosine play in the Fourier transform. The weights $w_j \in \mathbb{R}$ are called **Walsh coefficients**. In this paper, the indices of both Walsh functions and coefficients may be expressed as either binary or the numerical equivalent.

The j^{th} Walsh function can be defined:

$$\psi_j(x) = (-1)^{bc(j \wedge x)}$$

where $j, x \in \mathcal{B}^L$. Thus, if $bc(j \wedge x)$ is odd, then $\psi_j(x) = -1$ and if $bc(j \wedge x)$ is even, then $\psi_j(x) = 1$. The j^{th} Walsh function looks at the parity of the bits selected by j. Hence there are 2^L Walsh functions.

An important property of Walsh coefficients is that w_j measures the contribution to the evaluation function by the interaction of the bits indicated by the positions of the 1's in j. Thus, w_{0001} measures the linear contribution to the evaluation function associated with bit position 0, while w_{0101} measures the nonlinear (multiple bit) interaction between the bits in positions 0 and 2, and so on. Therefore Equation 1 says that any function over L bit space can be represented as a weighted sum of all possible 2^L bit interaction functions ψ_j. This nonlinearity is an important feature in determining problem difficulty for genetic algorithms [Goldberg, 1989a, Goldberg, 1989b, Reeves and Wright, 1995].

The 2^L Walsh coefficients can be computed by a Walsh transform:

$$w_j = \frac{1}{2^L} \sum_{x=0}^{2^L-1} f(x)\psi_j(x) \tag{2}$$

The calculation of Walsh coefficients can be thought of in terms of matrix multiplication. Let $\vec{f}$ be a column vector of 2^L elements where the i^{th} element is the evaluation of function $f(i)$. Similarly define a column vector $\vec{w}$ for the Walsh coefficients. If M is a $2^L \times 2^L$ matrix where $M_{i,j} = \psi_j(i)$, also known as a Hadamard matrix, then:

$$\vec{w} = \frac{1}{2^L} \vec{f}^{\mathsf{T}} M$$

For example, if we have a 3 bit function with the 2^3 function evaluations labeled $f_0..f_7$, then the Walsh coefficient calculation would be:

$$\vec{w} = \frac{1}{8} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \tag{3}$$

3 WALSH ANALYSIS OF EMBEDDED LANDSCAPES

Since understanding the bit interactions of both MAXSAT and NK-landscapes will help us better understand the mechanisms of difficulty for these problems doing a Walsh Analysis of embedded landscapes, which contain both problems, should be useful. Equation 2 suggests that computing the Walsh coefficients of a function requires the complete enumeration of the function space. In general, this makes Walsh analysis impractical for analyzing bit interactions for problems of nontrivial size. This is not the case for embedded landscapes. The fact that an embedded landscape is composed of a sum of generally much smaller subfunctions will give us leverage in computing the Walsh coefficients. If the dimension of the subfunctions is bounded by some k then the Walsh transform can be computed in polynomial time with respect to the number of bits in the domain of the function [Heckendorn et al., 1998].

We outline this process as follows: By definition, an embedded landscape, $f : \mathcal{B}^L \rightarrow \mathbb{R}$, is a sum of subfunctions:

$$f(x) = \sum_{j=1}^{Q} g_j(pack(x, m_j)).$$

The Walsh coefficients for each subfunction $g_j : \mathcal{B}^{bc(m_j)} \rightarrow \mathbb{R}$ can be computed:

$$w_i^{g_j} = \frac{1}{2^{bc(m_j)}} \sum_{y=0}^{2^{bc(m_j)}-1} g_j(y)\psi_i(y) \qquad i \in \mathcal{B}^{bc(m_j)}$$

where w^{g_j} are the Walsh coefficients of g_j. It has been shown [Heckendorn and Whitley, 1997] for a function composed of just a single subfunction, that is:

$$f(x) = g(pack(x, m)),$$

that exactly the Walsh coefficients of g appear as Walsh coefficients in the higher dimensional function, f, and the remaining are all zero.

$$w_i^f = \begin{cases} w^g_{pack(i,m)} & \text{if} \quad i \subseteq m \\ 0 & \text{otherwise} \end{cases} \qquad i \in \mathcal{B}^N \tag{4}$$

where m is the interaction mask for the subfunction. $i \subseteq m$, denotes that i is **contained in** m, which means that all of the positions in i that contain a 1 also have a 1 in m. Said another way $i \wedge m = i$. Equation 4 shows that the only possible nonzero Walsh coefficients are those whose bit patterns are contained in the interaction mask for the function. It is known that the Walsh coefficients for a k bit function g can be computed in $\mathcal{O}(k2^k)$ time using a **fast Walsh transform**. Therefore, the Walsh coefficients of function f above can be computed in the same time.

Since the Walsh transform is a simple linear transformation, the Walsh transform of a sum of functions is the sum of the Walsh transform of the functions. Therefore, the Walsh transform of an embedded landscape with Q subfunctions can be treated as a sum of the Walsh transforms of each individual subfunctions [Heckendorn et al., 1998]. Therefore:

$$w_i^f = \sum_{j=1}^{Q} w_{pack(i,\ m_j)}^{g_j} \tag{5}$$

It follows from Equations 4 and 5 that there are at most 2^kQ nonzero Walsh coefficients. This means the number of nonzero Walsh coefficients is linear in Q. But Q must be bounded by the number of possible subsets of k bits. Hence for an L bit function: $Q \leq \binom{L}{k}$. That is, Q polynomially bounded by L for fixed values of k. Equation 5 tells us that all the Walsh coefficients for an embedded landscape, f, can be computed in $\mathcal{O}(k2^kQ)$ time by using the fast Walsh transform. Hence the Walsh coefficients can be computed in polynomial time with respect to L for fixed values of k. For example: in the case of Walsh coefficients for a MAX3SAT problem with C clauses, there are at most $8C$ nonzero Walsh coefficients. They can be computed in $\mathcal{O}(3 \cdot 8C)$ time.

To show how these equations work, Table 1 shows an example of Walsh coefficient calculations for a small MAX3SAT function $f : \mathcal{B}^4 \rightarrow \mathbb{R}$ with $f(x) = f_1 + f_2 + f_3$ and

$$f_1 = (\overline{x_2} \vee x_1 \vee x_0)$$
$$f_2 = (x_3 \vee \overline{x_2} \vee x_1)$$
$$f_3 = (x_3 \vee \overline{x_1} \vee \overline{x_0}).$$

Since each clause is defined over 3 variables, their corresponding Walsh coefficients can be computed using Equation 3. When a clause is evaluated over all possible variable assignments, each clause will produce seven 1's and a single 0. This property can be exploited to produce the Walsh coefficients for MAXSAT clauses directly from the clause description[Rana et al., 1998]. The Walsh coefficients for each clause are listed in Table 1 as $W(f_1)$, $W(f_2)$ and $W(f_3)$. According to equation 5, we simply sum $W(f_1)$, $W(f_2)$ and $W(f_3)$ to produce $W(f(x))$. Note that, unlike this example, for most embedded landscapes k is much less than the number of bits in the domain.

Table 1: Walsh Coefficients broken down by clause.

x	w_i	$W(f_1)$	$W(f_2)$	$W(f_3)$	$W(f(x))$
0000	w_0	0.875	0.875	0.875	2.625
0001	w_1	−0.125	0	0.125	0
0010	w_2	−0.125	−0.125	0.125	−0.125
0011	w_3	−0.125	0	−0.125	−0.250
0100	w_4	0.125	0.125	0	0.250
0101	w_5	0.125	0	0	0.125
0110	w_6	0.125	0.125	0	0.250
0111	w_7	0.125	0	0	0.125
1000	w_8	0	−0.125	−0.125	−0.250
1001	w_9	0	0	0.125	0.125
1010	w_{10}	0	−0.125	0.125	0
1011	w_{11}	0	0	−0.125	−0.125
1100	w_{12}	0	0.125	0	0.125
1101	w_{13}	0	0	0	0
1110	w_{14}	0	0.125	0	0.125
1111	w_{15}	0	0	0	0

Since all nonzero Walsh coefficients for any MAX3SAT problem can be computed in P-time with respect to number of bits in the domain and MAX3SAT is NP-complete [Papadimitriou, 1994], then if $P \neq NP$, knowing the exact linear and nonlinear interactions of a function cannot be sufficient for inferring the global optimum.

We briefly discussed how Walsh coefficients can indicate the bitwise nonlinearity in a problem that can lead to problem difficulty for GAs. We have shown how the Walsh coefficients can be quickly computed for any embedded landscape with bounded subfunction size. In the next section we show how Walsh analysis can also probe the statistical nature of embedded landscapes by calculating summary statistics in polynomial time.

4 STATISTICS FOR PROBLEM INSTANCES

Walsh analysis can be used to compute summary statistics for **fitness distributions** of discrete optimization problems. Note that the fitness distribution is the distribution formed by evaluating all possible inputs to a problem. So, for a problem defined over 2^L possible inputs, the distribution would be composed of all 2^L evaluations of the inputs. Clearly, computing summary statistics for arbitrary fitness distributions would require exponential time. Goldberg and Rudnick [Goldberg and Rudnick, 1991] have used

Walsh coefficients to calculate fitness variance for fitness distributions and schemata; however, the calculations were intended for enumerable functions.

In this section, we show how higher order statistics such as skew and kurtosis can be also be computed from the Walsh coefficients by using a general formula for computing the r^{th} moment for any embedded landscape fitness distribution and for any other problem where all nonzero Walsh coefficients are known. Since Walsh analysis can be performed for embedded landscapes in polynomial time with respect to L, these summary statistics can also be computed in polynomial time.

Given the mean, the formula used to compute the r^{th} moment, denoted μ_r, for a discrete random variable X is:

$$\mu_r \;=\; E[(X-\mu)^r] \;=\; \sum_{x\in X}(x-\mu)^r p(x)$$

where $p(x)$ is the probability of x occurring. For our purposes, the function $p(x)=\frac{1}{2^L}$ since we are enumerating a function over all L bit binary strings. Since $\psi_0 = 1$ for all inputs we see from Equation 2 that Walsh coefficient w_0 is the mean of all fitnesses. The function then becomes:

$$\mu_r = \sum_{x\in X}\frac{(x-\mu)^r}{2^L}$$

If X represents a real valued function over an L bit domain then:

$$\mu_r = \frac{1}{2^L}\sum_{x=0}^{2^L-1}(f(x)-\mu)^r$$

We can substitute for f with the linear Walsh representation of f from Equation 1:

$$\mu_r = \frac{1}{2^L}\sum_{x=0}^{2^L-1}\left(\sum_{i=0}^{2^L-1} w_i\psi_i(x)-\mu\right)^r$$

Since $\mu = w_0$, and $\psi_0(x)=1\ \forall x$:

$$\mu_r = \frac{1}{2^L}\sum_{x=0}^{2^L-1}\left(\sum_{i=1}^{2^L-1} w_i\psi_i(x)\right)^r$$

We can now expand the exponential creating a set of r indices a_j where $a_j \in \mathcal{B}^L$:

$$\mu_r = \frac{1}{2^L}\sum_{x=0}^{2^L-1}\left(\sum_{a_1=1}^{2^L-1} w_{a_1}\psi_{a_1}(x)\right)\left(\sum_{a_2=1}^{2^L-1} w_{a_2}\psi_{a_2}(x)\right)\ldots$$
$$\ldots\left(\sum_{a_r=1}^{2^L-1} w_{a_r}\psi_{a_r}(x)\right)$$

Since the Walsh coefficients do not depend on x, the formula can be rewritten as:

$$\mu_r = \frac{1}{2^L}\sum_{a_1=1}^{2^L-1}\sum_{a_2=1}^{2^L-1}\ldots\sum_{a_r=1}^{2^L-1} w_{a_1}w_{a_2}\ldots$$
$$\ldots w_{a_r}\sum_{x=0}^{2^L-1}\psi_{a_1}(x)\psi_{a_2}(x)\ldots\psi_{a_r}(x)$$

Using the fact that for arbitrary p and q: $\psi_p(x)\psi_q(x) = \psi_{p\oplus q}(x)$:

$$\mu_r = \frac{1}{2^L}\sum_{a_1=1}^{2^L-1}\sum_{a_2=1}^{2^L-1}\ldots\sum_{a_r=1}^{2^L-1} w_{a_1}w_{a_2}\ldots$$
$$\ldots w_{a_r}\sum_{x=0}^{2^L-1}\psi_{a_1\oplus a_2\oplus..\oplus a_r}(x)$$

Now using the fact that:

$$\sum_{x=0}^{2^L-1}\psi_i(x) = \begin{cases} 0 & \text{if } i\neq 0 \\ 2^L & \text{if } i = 0\end{cases}$$

we see that **only when** $a_1\oplus a_2\oplus\ldots\oplus a_r = 0$ is the inner sum nonzero. Therefore,

$$\mu_r = \frac{1}{2^L}\sum_{a_1\oplus a_2\oplus\ldots\oplus a_r=0} w_{a_1}w_{a_2}\ldots w_{a_r}2^L, \qquad a_i\neq 0\ \forall\ i$$
$$= \sum_{a_1\oplus a_2\oplus\ldots\oplus a_r=0} w_{a_1}w_{a_2}\ldots w_{a_r}, \qquad a_i\neq 0\ \forall\ i \qquad (6)$$

To summarize, given the set of nonzero Walsh coefficients, we can compute the r^{th} moment for the fitness distribution using products of the Walsh coefficients such that the exclusive-or of the indices is zero.

This formula allows us to compute the variance, skew and kurtosis for any fitness distribution provided we are given the Walsh coefficients.

$$variance = \mu_2 = \sigma^2 \qquad skew = \frac{\mu_3}{\sigma^3} \qquad kurtosis = \frac{\mu_4}{\sigma^4}$$

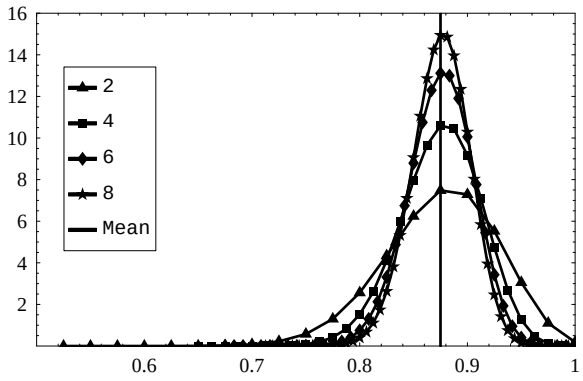

Figure 1: PDF of Fitness Distributions for 20 Variable MAX3SAT Problems with Clause/Var from 2 to 8.

For example, since $a_1 \oplus a_2 = 0$ if and only if $a_1 = a_2$ then the variance for any function can be computed

$$\sum_{i=1}^{2^L-1} w_i w_i$$

Of course, this computation of the moment around the mean, if done directly, would take $\mathcal{O}(2^{Lr})$ time. However, in the case of an embedded landscape, only a polynomial number of Walsh coefficients are nonzero and only the nonzero coefficients need be considered. Selecting the indices to have even parity would consist of selecting the first $r-1$ indices from the set of nonzero Walsh coefficients. The exclusive-or of these would be taken and would be used as the desired r^{th} index. The exclusive-or of the r indices would therefore be zero. Using this simple strategy it would take $\mathcal{O}(n^{r-1})$ time to compute the r^{th} moment given the Walsh coefficients, where n is the number of nonzero Walsh coefficients. In the case of embedded landscapes the nonzero Walsh coefficients can be identified and computed in polynomial time. Since our moment computation strategy is also polynomial time for fixed r, all the summary statistics can be computed from formula 6 in polynomial time.

4.1 SUMMARY STATISTICS FOR LARGE MAXSAT PROBLEMS

This statistical analysis applies to any embedded landscape including both NK-landscapes and MAXkSAT problems. So it is now possible to calculate summary statistics for even high dimensional examples of these important classes of GA test problems. As an example, in this section we will compute summary statistics for MAX3SAT problems including many that would be too large to enumerate. We will compare the computed skew versus a set of exhaustively enumerated smaller MAXSAT problems and see that the computed skew correctly predicts the shape of the distribution. We will also look for statistical indicators of the well known phase transition found in MAX3SAT problems [Kirkpatrick and Selman, 1994]. The **phase transition** occurs when the clause to variable ratio for a MAX3SAT problem approaches 4.3. At that point there is a spike in problem difficulty relative to many deterministic algorithms and the number of potential solutions for MAX3SAT problems drops off exponentially. We begin by discussing the mean and median of MAX3SAT problems.

The Walsh analysis of MAXSAT problems is related to polynomial time c-approximate algorithms ($0 \leq c \leq 1$) for NP-complete problems[Papadimitriou, 1994]. Approximate algorithms are polynomial time algorithms designed to guarantee solutions to NP-complete optimization problems that are at least a $1-c$ times the optimal solution, assuming maximization. The value for c for MAX3SAT is $\frac{1}{8}$[Trevisan, 1997, Papadimitriou, 1994]. Recently, Håstad has been proven that it is an NP-hard problem to *guarantee* a solution that is $\frac{7}{8}+\epsilon$, $\epsilon > 0$, of optimal[Håstad, 1997]. The basis of the $\frac{7}{8}$ limit for MAXSAT is that $\frac{7}{8}$ is simply the probability that each clause can be satisfied by randomly setting the variables. Not surprisingly, the mean (but not necessarily the median) of the fitness distribution of any MAX3SAT problem is simply $\frac{7}{8}$ times the number of clauses in the problem.

To better understand how function values are distributed for MAX3SAT problems, the average fitness distributions of all points in 50 randomly generated 20 variable MAX3SAT problems are examined as probability density functions. The probability distribution function (**pdf**) gives us insight into what kinds of output we can expect for randomly generated inputs; namely, do most solutions lie to the left or right of the mean ($\frac{7}{8}$ optimal)? Figure 1 is the composite **pdf** calculated by averaging the **pdf**s for 50 individual MAX3SAT problems. The figure shows four composite **pdf**s for clause to variable ratios of 2, 4, 6, 8. The x-axis corresponds to the ratio of satisfied to total number of clauses while the y-axis represents the frequency that particular ratio occurred. The optimum occurs at a ratio of 1.0 and the mean fitness is $\frac{7}{8}$ for all problem sets.

The set of **pdf**s illustrates that the fitness distributions are skewed to the left of the mean fitness, $\frac{7}{8}$. As the clause to variable ratio increases, the skew tends towards 0 but still remains negative. So, the majority

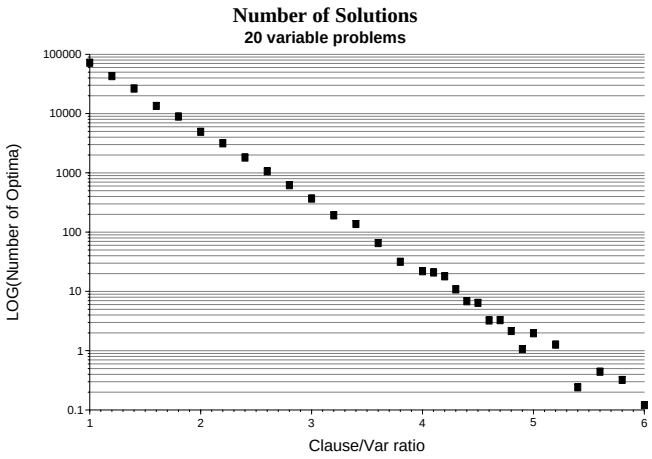

Figure 2: Counts of optimal solutions for 20-variable MAX3SAT problems using a log scale.

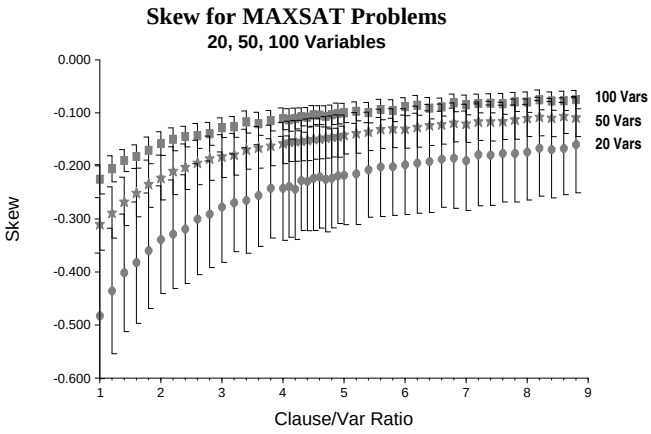

Figure 4: Skew of fitness distributions for MAX3SAT problems.

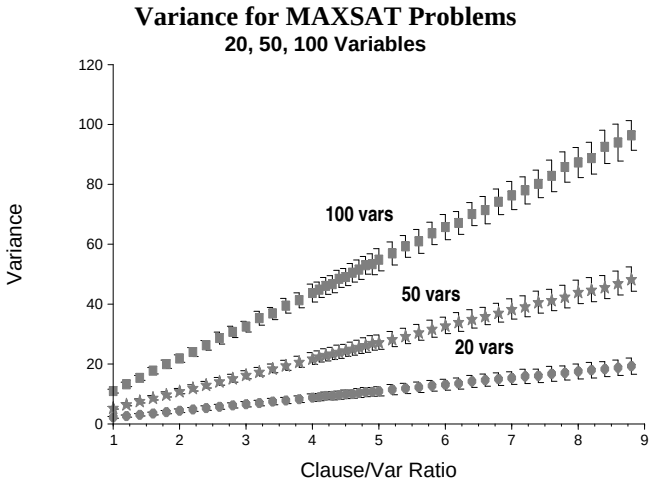

Figure 3: Variances of fitness distributions for MAX3SAT problems.

of fitness values are higher than $\frac{7}{8}$ optimal. However, the actual number of optimal solutions is decreasing as the clause to variable ratio increases; in fact, the number of solutions decreases at an exponential rate as illustrated in Figure 2.

The histograms for 20 variable problems provide some insight into the fitness distributions of higher dimensional MAX3SAT problems. We can use the Walsh formulas to compute the variance, skew and kurtosis for fitness distributions that are too large to be enumerated. Figures 3 and 4 respectively illustrate the average variance and skew of the fitness distributions for 20, 50 and 100 variable MAX3SAT problems. The plots for both the 20 and 50 variable problems were averaged over 500 problem instances. We used 100 problem instances at each point for the 100 variable problems. The region representing clause/variable ratios between 4 and 5 was more heavily sampled by using increments of 0.1, while the remaining points were computed at increments of 0.2. Since kurtosis calculations are $\mathcal{O}(n^3)$, we limited our kurtosis calculations to only the set of 500 20 variable problems. The kurtosis was approximately 3 regardless of the clause/variable ratio.

While no evidence of a phase transition is found in these statistics, the skew of the randomly generated MAXSAT problems was always negative which agreed with the results in Figure 1. For this rather large set of randomly generated MAXSAT problems, it would appear that the majority of solutions are actually greater than $\frac{7}{8}$ of optimal. This also implies that if randomly generated SAT problems are being used to test search algorithms, then we need to be aware of the biases that exist in that set of fitness distributions.

The summary statistics for problem instances of randomly generated MAX3SAT problems indicate that the majority of solutions are actually higher than average. For over one thousand MAX3SAT problem instances, the skew was negative. While there are cases of other MAX3SAT problem instances with positive skew, Crawford's parity function learning problems from the DIMACS benchmark problem set [Dim, 1993], the randomly generated MAX3SAT problems have biased fitness distributions. While there was no indication of phase transition behavior for the MAX3SAT summary statistics, it is interesting to note that the number of solutions drops exponentially with the clause to variable ratio.

5 CONCLUSIONS

Embedded landscapes are a broad class of functions encompassing both MAXSAT and NK-landscapes. As with MAXSAT and NK-landscapes, the Walsh transform can be performed on embedded landscapes in polynomial time with respect the number of bits in the function domain. Once obtained, the nonzero Walsh coefficients can be used to compute the summary statistics (i.e. mean, variance, skew and kurtosis) in polynomial time. Using our polynomial time algorithms we were able to compute the summary statistics for large MAXSAT problems and use this to look for statistical indicators of a phase transition.

The polynomial time Walsh analysis of embedded landscapes also means that we can exactly compute schema averages upto a fixed order in polynomial time [Rana et al., 1998] [Goldberg, 1989a] [Goldberg, 1989b]. Knowing exact schema averages and exact summary statistics for any particular problem instance actually provides a significant amount of information. On the other hand, despite having all this information, the results of Håstad [Håstad, 1997] indicates that in the general case, no search algorithm or exact method can be guaranteed to generate a solution that is better than μ, the average evaluation, unless P = NP.

6 ACKNOWLEDGMENTS

This work was supported by NSF grant IRI-9503366 and AFOSR grant F49620-97-1-0271. Soraya Rana was also supported by a National Physical Science Consortium fellowship awarded by NASA-SSC.

References

[Dim, 1993] (1993). The second dimacs international algorithm implementation challenge on clique, graph coloring and satisfiability. http://dimacs.rutgers.edu/pub/challenge.

[Davis and Putnam, 1960] Davis, M. and Putnam, H. (1960). A computing procedure for quantification theory. *Journal of the ACM*, 7:201–215.

[Goldberg, 1989a] Goldberg, D. (1989a). Genetic algorithms and walsh functions: Part i, a gentle introduction. *Complex Systems*, 3:129–152.

[Goldberg, 1989b] Goldberg, D. (1989b). Genetic algorithms and walsh functions: Part ii, deception and its analysis. *Complex Systems*, 3:153–171.

[Goldberg and Rudnick, 1991] Goldberg, D. E. and Rudnick, M. W. (1991). Genetic algorithms and the variance of fitness. *Complex Systems*, 5(3):265–278.

[Håstad, 1997] Håstad, J. (1997). Some optimal inapproximability results. In *Proceedings of the 29th ACM Symposium on Theory of Computation*, pages 1–10.

[Heckendorn and Whitley, 1997] Heckendorn, R. B. and Whitley, D. (1997). A walsh analysis of nk-landscapes. In Bäck, T., editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 41–48. Morgan Kaufmann.

[Heckendorn et al., 1998] Heckendorn, R. B., Whitley, L. D., and Rana, S. (1998). Test function generators as embedded landscapes. In *Foundations of Genetic Algorithms - 5*, Leiden, The Netherlands.

[Kauffman, 1989] Kauffman, S. A. (1989). Adaptation on rugged fitness landscapes. In Stein, D., editor, *Lectures in the Science of Complexity*, pages 527–618. Addison-Wesley.

[Kauffman, 1993] Kauffman, S. A. (1993). *Origins of Order*. Oxford Press.

[Kirkpatrick and Selman, 1994] Kirkpatrick, S. and Selman, B. (1994). Critical behavior in the satisfiability of random boolean expressions. *Science*, 264:1297–1301.

[Papadimitriou, 1994] Papadimitriou, C. H. (1994). *Computational Complexity*. Addison-Wesley Publishing, Co.

[Rana et al., 1998] Rana, S., Heckendorn, R., and Whitley, D. (1998). A tractable walsh analysis of SAT and its implications for genetic algorithms. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*.

[Reeves and Wright, 1995] Reeves, C. and Wright, C. (1995). An experimental design perspective on genetic algorithms. In Whitley, D. and Vose, M., editors, *Foundations of Genetic Algorithms - 3*, pages 7–22. Morgan Kaufmann.

[Trevisan, 1997] Trevisan, L. (1997). Approximating satisfiable satisfiability problems. In *Proceedings of the 5th European Symposium on Algorithms*, pages 472–485. Springer-Verlag.

[Weinberger, 1990] Weinberger, E. (1990). Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63:325–226.