A Polynomial Time Computation of the Exact Correlation Structure of k-Satisfiability Landscapes

Andrew M. Sutton L. Darrell Whitley Adele E. Howe

Department of Computer Science
Colorado State University
Fort Collins, CO 80523
{sutton,whitley,howe}@cs.colostate.edu

ABSTRACT

The *autocorrelation function* and related *correlation length* are statistical quantities that capture the *ruggedness* of the fitness landscape: a measure that is directly related to the hardness of a problem for certain heuristic search algorithms. Typically, these quantities are estimated empirically by sampling along a random walk. In this paper, we show that a polynomial-time Walsh decomposition of the k-satisfiability evaluation function allows us to compute the *exact* autocorrelation function and correlation length for any given k-satisfiability instance. We also use the decomposition to compute a theoretical expectation for the autocorrelation function and correlation length over the ensemble of instances generated uniformly at random. We find that this expectation is invariant to the constrainedness of the problem as measured by the ratio of clauses to variables. However, we show that *filtered* problems, which are typically used in local search studies, have a bias that causes a significant deviation from the expected correlation structure of unfiltered, uniformly generated problems.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Theory

Keywords

Combinatorial Optimization, Fitness Landscapes

1. INTRODUCTION

The correlation structure of the fitness landscape is an important measure of how "smooth" or "rugged" the search space appears to certain search methods. Stadler and Schnabl [17] have conjectured that, under fairly reasonable conditions, the correlation length of the landscape is directly related to the number of local optima. Several researchers have also proposed using these ruggedness values to classify and compare NP-hard optimization problems [7, 1].

The behavior of local methods such as local search and $(1 + \lambda)$ evolution strategies directly depends on the correlation structure of the search space. An instance with high correlation between states that are separated by a small number of steps or mutations will appear in some sense "smoother" to a search algorithm than one of low correlation. Intuitively, this is because nearby states will tend to have a more similar fitness. On the other hand, an instance with low correlation will appear highly rugged with a large number of local peaks. This information can be captured by the *autocorrelation function* $r(s)$ which measures the correlation between points separated by s random steps or mutations. Similarly the *correlation length* ℓ is a measure of how far along a random walk states tend to be correlated.

Hoos et al. [6] have recently studied the local search space structure and its effect on the performance of local search algorithms on k-satisfiability, maximum-k-satisfiability, and weighted maximum-k-satisfiability. The authors measure the correlation length empirically along a sampled random walk using a subset of the Hamming neighborhood. This method was first proposed by Weinberger [18]. Mathematically, the *exact* autocorrelation function is well-defined for the entire landscape [15]; however, for general functions, computing the exact autocorrelation requires enumerating the entire search space.

In this paper, we show that the autocorrelation function (and the correlation length) for the complete Hamming neighborhood can be computed exactly for any given k-satisfiability instance in polynomial time. We use a well-known result from Stadler [15] that relates the correlation structure of a problem instance to the coefficients in a decomposition of the fitness function. We show that the Walsh decomposition of a k-satisfiability instance provides these coefficients (when properly normalized) and use a result by Rana et al. [12] that every k-satisfiability instance has a tractable Walsh decomposition.

To study the expected behavior of correlation information over an entire problem distribution, we present a theoretical *expectation* for the autocorrelation function and correlation length over uniformly generated random k-satisfiability instances with n variables and m clauses. We show that

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

this expectation is invariant to clause-variable ratio m/n on randomly generated problems. However, on *filtered* problem sets, i.e., ensembles of randomly generated problems for which unsatisfiable instances are discarded, there is a significant deviation from this expectation. This result has important implications since most empirical studies for local search are performed on *filtered* sets [3, 21, 11, 13].

This paper is organized as follows. In the next section we define the correlation structure of problems and how it can be measured by the autocorrelation function and the correlation length. We show that, on k-satisfiability instances, the exact versions of these functions can be computed in polynomial time. In Section 3 we extend this analysis to classes of problems generated uniformly at random. In Section 4 we present numerical results. We conclude the paper in Section 5.

2. CORRELATION STRUCTURE

In the context of molecular biology, Eigen et al. [4] first studied correlation functions to measure the local structure of state-space models of certain molecular configurations. This idea has since been extended to studying the search space explored by local methods on combinatorial optimization problems [14, 1, 9, 6].

The concept of correlation structure extends naturally to combinatorial search methods that operate by perturbing complete candidate solutions. Local methods select from elements of a predefined *neighborhood* given by a *mutation* or *move* operator. Elements of this neighborhood are generated systematically or at random and then selected based on their fitness.

The neighborhood relationship among states imposes a connectivity on the search space. A state y is reachable from a state x in s steps if there exist exactly s moves or mutations that can transform x into y. The statistical relationship between this reachability and the fitness function defines the *correlation structure* of the landscape.

This structure can be measured by the *autocorrelation function* which captures the strength of the relationship between fitness and the number of steps from any particular point, and the *correlation length* which, informally, measures the expected range of correlation among states.

These measures define the *ruggedness* of the fitness landscape. A landscape with low correlation will tend to be rugged and have many local optima. On the other hand, a landscape with high correlation will tend to be "smoother" and more amenable to local methods. This phenomenon is illustrated in Figure 1. The autocorrelation function $r(s)$ quantifies the strength of the relationship between points separated by s steps.

2.1 The k-satisfiability landscape

An instance of the k-satisfiability problem consists of a formula that is the conjunction of m disjunctive clauses containing k literals each

$$(l_{1,1} \vee \ldots \vee l_{1,k}) \wedge (l_{2,1} \vee \ldots \vee l_{2,k}) \wedge \ldots \wedge (l_{m,1} \vee \ldots \vee l_{m,k})$$

where $l_{i,j}$ is taken to be one of the n variables or its negation.

In the *decision* variant (k-satisfiability), the objective is to determine the existence of an assignment to all n variables such that the formula evaluates to true. In the optimization variant (maximum-k-satisfiability), the objective is to find

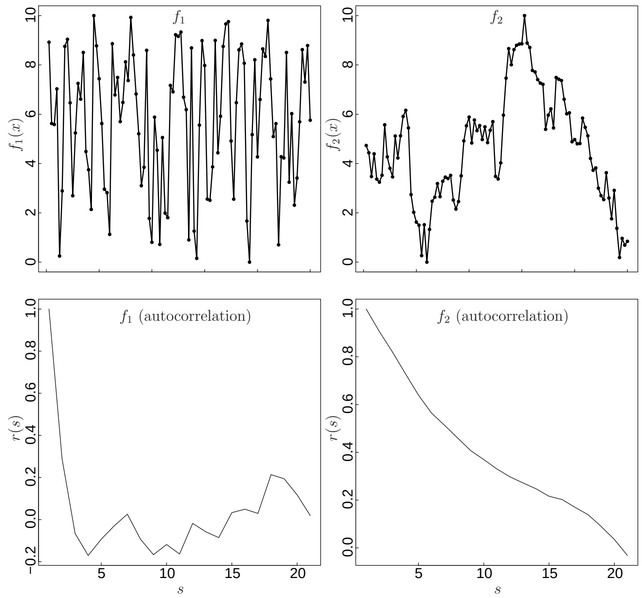

Figure 1: Two landscapes: f_1 [top left] with low correlation structure [bottom left] and f_2 [top right] with high correlation structure [bottom right] as measured by the autocorrelation function.

an assignment to all n variables that maximizes the number of clauses that evaluate to true.

Incomplete heuristic search algorithms cannot, in principle, solve the decision variant, e.g., they cannot prove an instance does not have a satisfying assignment. Thus, most studies of incomplete algorithms that search the space of complete candidate assignments by performing local perturbations (moves or mutations) typically view both variants from the perspective of optimization. Thus the fitness landscape for *both* the decision variant (k-satisfiability) and the optimization variant (maximum-k-satisfiability) are identical and our results apply to both.

Let $\{c_1, c_2, \ldots, c_m\}$ be the set of m clauses in an instance. Let $\{v_1, v_2, \ldots, v_n\}$ denote the set of n variables. Each state is a complete assignment to all n variables. The set of all possible assignments, which we denote X, is isomorphic to the set of binary sequences of length n. In other words, the complete assignment x corresponds directly to a sequence $(x[1], x[2], \ldots, x[n])$ with

$$x[b] = \begin{cases} 1 & \text{if variable } v_b \text{ is set to true in the assignment} \\ 0 & \text{if variable } v_b \text{ is set to false} \end{cases}$$

Thus we can express the fitness function in terms of binary sequences of length n:

$$f : \{0,1\}^n \mapsto \{0, 1, \ldots, m\}$$

where $f(x)$ counts the number of clauses satisfied in the complete assignment to all n variables given by the binary sequence x. We denote by $N(x)$ the *Hamming neighborhood* of a state x: the set of all binary sequences of x that differ in exactly one position. The state space X together with the fitness function f and the neighborhood N, which provides the connectivity on X give the fitness landscape of the k-satisfiability problem.

2.2 Exact correlation structure

We define the $2^n \times 2^n$ *random walk transition matrix* $\mathbf{T}$ as

$$\mathbf{T}_{xy} = \begin{cases} \frac{1}{n} & \text{if } y \text{ is a Hamming neighbor of } x \\ 0 & \text{otherwise} \end{cases}$$

Each component of the matrix $\mathbf{T}_{xy}$ can be viewed as the unbiased random walk transition probability from state x to state y. We may also view $\mathbf{T}$ as a linear operator over functions $g : X \mapsto \mathbb{R}$. In particular, since such a function g is over a discrete domain, we can characterize it as a vector $g \in \mathbb{R}^{|X|}$. The matrix-vector product $\mathbf{T}g \in \mathbb{R}^{|X|}$ can in turn be viewed as a function over X. This function evaluated at state x gives the average of the function g over the Hamming neighborhood of x:

$$\mathbf{T}g(x) = \frac{1}{n} \sum_{y \in N(x)} g(y) \tag{1}$$

For example, $\mathbf{T}f(x)$ gives the average fitness of the Hamming neighbors of x. In general, $\mathbf{T}^s g(x)$ gives the expectation of g evaluated at a point s random steps from x.

Weinberger [18] showed the correlation information can be measured using a finite sequence of states $(x_1, x_2, \ldots, x_q)$ with $x_{t+1} \in N(x_t)$ generated by performing a random walk on the hypercube induced by the Hamming neighborhood. We can then measure the lag-s autocorrelation of the "time series" of evaluations $(f(x_1), f(x_2), \ldots, f(x_q))$. This empirical lag-$s$ random walk autocorrelation function $\hat{r}(s)$ has an *expectation* of

$$r(s) = \frac{E[f(x_t)f(x_{t+s})] - E[f(x_t)]^2}{E[f(x_t)^2] - E[f(x_t)]^2}$$

where $E[\cdot]$ is the expectation over all times and initial states. Intuitively, $r(s)$ is the expected statistical correlation in fitness between two points separated by s steps. Stadler [15] showed that this expression simplifies.

$$r(s) = \frac{2^{-n}\langle f, \mathbf{T}^s f\rangle - \left(2^{-n}\sum_{x\in X} f(x)\right)^2}{2^{-n}\sum_{x\in X} f(x)^2 - \left(2^{-n}\sum_{x\in X} f(x)\right)^2} \tag{2}$$

where $\langle \cdot, \cdot \rangle$ is the standard inner product.

Thus Equation (2) captures the *exact* statistical correlation of the entire landscape: the quantity which empirical random-walk autocorrelation is estimating.

2.3 Walsh decomposition

The *Walsh transform* is an analog of the Fourier transform which decomposes an arbitrary pseudo-Boolean function $g : \{0,1\}^n \mapsto \mathbb{R}$ into a superposition of Walsh functions:

$$g(x) = \sum_{i=0}^{2^n-1} w_i \psi_i(x)$$

where ψ_i is defined as follows.

Definition 1. *The Walsh function $\psi_i(x)$ in its* normalized *form is defined as*

$$\psi_i(x) = \frac{1}{\sqrt{2^n}}(-1)^{\langle i,x\rangle}$$

Note that the index i is interpreted as the length-n *bitstring* representation of i in the definition. The inner product in the exponent simply counts the number of 1 bits in the intersection of the bitstrings i and x.

In the worst case, a pseudo-Boolean function can be written as a superposition of $O(2^n)$ Walsh functions. However, Rana et al. [12] have shown that the evaluation function f of a k-satisfiability problem can be written as a linear combination of $O(2^k m)$ Walsh functions:

$$f(x) = \sum_i w_i \psi_i(x) \tag{3}$$

where w_i is the i^{th} Walsh coefficient. Since k is taken to be $O(1)$, the complexity is linear in the number of clauses.

For k-satisfiability, Rana et al. showed that each Walsh coefficient w_i is a sum of contributions from each clause:

$$w_i = \sum_{j=1}^{m} w_i(c_j)$$

where $w_i(c_j)$ is the contribution to w_i from clause c_j. This is defined as follows. Let $v(c_j)$ denote a bitstring of length n where

$$v(c_j)[b] = \begin{cases} 1 & \text{if variable } v_b \text{ appears in clause } c_j \\ 0 & \text{otherwise} \end{cases}$$

Similarly, let $u(c_j)$ be a bitstring of length n where

$$u(c_j)[b] = \begin{cases} 1 & \text{if variable } v_b \text{ appears } \textit{negated} \text{ in clause } c_j \\ 0 & \text{otherwise} \end{cases}$$

If x and y are bitstrings of length n, we say

$$x \subseteq y \iff (x[b] = 1 \implies y[b] = 1)$$

for $b = \{1, \ldots, n\}$. The contribution of clause c_j to Walsh coefficient w_i is

$$w_i(c_j) = \begin{cases} 0 & \text{if } i \not\subseteq v(c_j) \\ \sqrt{2^n}\,\frac{2^k-1}{2^k} & \text{if } i = 0 \\ -\sqrt{2^n}\,\frac{1}{2^k}\psi_i(u(c_j)) & \text{otherwise} \end{cases} \tag{4}$$

The Walsh coefficient w_i is simply a sum of $w_i(c_j)$ over all clauses c_j in the instance. All remaining Walsh coefficients are zero.

Intuitively, it should be clear that for $i \neq 0$, $w_i(c_j)$ can be computed for clause c_j as follows.

1. If the variables selected by the nonzero bits in the bitstring representation of i do *not* appear together in clause c_j ($i \not\subseteq v(c_j)$), then $w_i(c_j) = 0$.
2. If these variables *do* appear together in clause c_j then the *parity* of the count of negations of these particular variables in clause c_j gives $\pm\sqrt{2^n}\frac{1}{2^k}$

As an illustrative example, suppose $n = 4$ and $i = 5 = 0101$. Then,

$$w_{0101} = \sum_{j=1}^{m} w_{0101}(c_j)$$

Now, $w_{0101}(c_j) = 0$ iff $0101 \not\subseteq v(c_j)$, that is, variables v_2 and v_4 are not both in clause c_j. Otherwise, w_{0101} contributes $\pm\sqrt{2^n}\frac{1}{2^k}$ depending on the value of $(-1)^{\langle 0101, u(c_j)\rangle}$. Clearly, this contribution is *positive* if variables v_2 and v_4 have an odd number of negations (i.e., only one is negated), otherwise it is negative.

Summing over all clauses c_j gives the i^{th} Walsh coefficient. More precisely, w_i can be computed by counting the number of clauses L_i^{odd} where the variables specified by the bitstring i appear together and are negated an odd number of times, and the number of clauses L_i^{even} where the variables specified by i appear together and are negated an even number of times.

$$w_i = \sqrt{2^n}\,\frac{1}{2^k}(L_i^{odd} - L_i^{even}) \tag{5}$$

Clearly, the *order* of any Walsh coefficient w_i (the number of ones in the bitstring representation of i) is bounded by the number of variables that can appear together in a clause. Indeed, it is enough to specify $f(x)$ by computing the $O(m)$ non-zero Walsh coefficients and computing the superposition in (3).

We have introduced the Walsh functions in a slightly different way to satisfy the following important properties.

LEMMA 1. *The Walsh functions $\{\psi_i\}$ form an orthonormal basis.*

PROOF. The Walsh functions are clearly orthogonal. Furthermore, since we have included the normalization factor $1/\sqrt{2^n}$ in Definition 1, we have

$$\langle \psi_i, \psi_i \rangle = 1$$

□

This lemma provides us with the following two corollaries which will become useful shortly. Let δ denote the Kronecker delta function and let $\vec{1}$ denote the all-ones vector $\{1\}^{2^n}$.

COROLLARY 1. $\langle \psi_i, \psi_j \rangle = \delta_{ij}$

COROLLARY 2.

$$\langle \vec{1}, \psi_i \rangle = \begin{cases} 0 & \text{if } i \neq 0 \\ \sqrt{2^n} & \text{otherwise} \end{cases}$$

PROOF. Follows directly from $\psi_0 = \frac{1}{\sqrt{2^n}}\left(\vec{1}\right)$. □

Lemma 1 states that the Walsh functions form an orthonormal basis *in which we can represent f*. The power of this approach is revealed in the fact that the Walsh functions are involved in the eigendecomposition of the random walk transition matrix.

LEMMA 2. *The i^{th} Walsh function ψ_i is an eigenvector of the random walk transition matrix $\mathbf{T}$ with eigenvalue $\lambda_i = \left(1 - \frac{2\langle i,i\rangle}{n}\right)$*

PROOF. Let x be an arbitrary state.

$$\left(\mathbf{T}\psi_i\right)(x) = \frac{1}{n}\sum_{y \in N(x)} \psi_i(y) \qquad \text{by Eq. (1)}$$

A Hamming neighbor $y \in N(x)$ differs from x in exactly one bit position b. Now, by Definition 1 we have,

$$\psi_i(y) = \frac{1}{\sqrt{2^n}}(-1)^{\langle i,y\rangle}$$

We denote as $i[b]$ the b^{th} bit of the bitstring representation of i. If $i[b] = 0$ then $\langle i, y\rangle = \langle i, x\rangle$ and $\psi_i(y) = \psi_i(x)$. On the other hand, if $i[b] = 1$ then $|\langle i, y\rangle - \langle i, x\rangle| = 1$ and $\psi_i(y) = -\psi_i(x)$.

Since each Hamming neighbor differs from x in exactly one of each of the n possible bit positions, there are $\langle i, i\rangle$ elements of $N(x)$ that satisfy the first condition and $n - \langle i, i\rangle$ that satisfy the second. Thus we have

$$\begin{aligned}\frac{1}{n}\sum_{y \in N(x)} \psi_i(y) &= \frac{1}{n}\left(\left(n - \langle i,i\rangle\right)\psi_i(x) - \langle i,i\rangle\psi_i(x)\right) \\ &= \frac{1}{n}\left(n - 2\langle i,i\rangle\right)\psi_i(x) \\ &= \left(1 - \frac{2\langle i,i\rangle}{n}\right)\psi_i(x) \\ &= \lambda_i\psi_i(x)\end{aligned}$$

Since we chose x arbitrarily,

$$\mathbf{T}\psi_i = \left(1 - \frac{2\langle i,i\rangle}{n}\right)\psi_i = \lambda_i\psi_i$$

□

Lemma 2 means that the Walsh functions are eigenfunctions of the random walk transition matrix $\mathbf{T}$. We can write expressions involving both f and $\mathbf{T}$ in terms of the component Walsh functions ψ_i to simplify the expression.

2.4 Expressing $r(s)$ in terms of the decomposition

Following Stadler [15] we introduce the normalized (Walsh) amplitudes of order p

$$W^{(p)} = \sum_{\langle i,i\rangle = p} w_i^2 \Big/ \sum_{j \neq 0} w_j^2$$

We derive an exact expression for $r(s)$ entirely in terms of the normalized amplitudes.

PROPOSITION 1.

$$r(s) = \sum_{p \neq 0} W^{(p)}\left(1 - \frac{2p}{n}\right)^s \tag{6}$$

PROOF. We first simplify Equation (2) using the Walsh decomposition (3). We replace $f(x)$ by the decomposition to simplify each term.

$$\begin{aligned}\sum_{x \in X} f(x) &= \sum_{x \in X}\sum_i w_i\psi_i(x) \\ &= \sum_i w_i \sum_{x \in X}\psi_i(x) \\ &= \sum_i w_i\langle \vec{1}, \psi_i\rangle \\ &= w_0\sqrt{2^n} && \text{by Cor. 2}\end{aligned}$$

$$\begin{aligned}\sum_{x \in X} f(x)^2 &= \sum_{x \in X}\left(\sum_i w_i\psi_i(x)\right)^2 \\ &= \sum_{i,j} w_i w_j \sum_{x \in X}\psi_i(x)\psi_j(x) \\ &= \sum_i w_i w_j\langle \psi_i, \psi_j\rangle \\ &= \sum_i w_i^2 && \text{by Cor. 1}\end{aligned}$$

$$\sum_{x \in X} \langle f, \mathbf{T}^s f \rangle = \langle \sum_i w_i \psi_i, \sum_i w_i \mathbf{T}^s \psi_i \rangle$$

$$= \sum_{i,j} w_i w_j \langle \psi_i, \lambda^s \psi_j \rangle \qquad \text{by Lem. 2}$$

$$= \sum_{i,j} w_i w_j \langle \psi_i, \psi_j \rangle \lambda^s$$

$$= \sum_i w_i^2 \lambda^s \qquad \text{by Cor. 1}$$

Thus Equation (2) simplifies:

$$r(s) = \frac{2^{-n} \sum_i w_i^2 \lambda^s - (2^{-n} w_0 \sqrt{2^n})^2}{2^{-n} \sum_i w_i^2 - (2^{-n} w_0 \sqrt{2^n})^2}$$

Pulling out the $i = 0$ terms in the summation and simplifying gives us

$$= \frac{2^{-n} w_0^2 + 2^{-n} \sum_{i \neq 0} w_i^2 \lambda_i^s - 2^{-n} w_0^2}{2^{-n} w_0^2 + 2^{-n} \sum_{i \neq 0} w_i^2 - 2^{-n} w_0^2}$$

$$= \frac{\sum_{i \neq 0} w_i^2 \lambda_i^s}{\sum_{i \neq 0} w_i^2}$$

$$= \frac{\sum_{i \neq 0} w_i^2 \left(1 - \frac{2\langle i, i \rangle}{n}\right)^s}{\sum_{i \neq 0} w_i^2} \qquad (7)$$

Equation (7) gives the proposed result. □

For example, the autocorrelation function for the well-studied 3-satisfiability problem can be expressed as follows.

$$r(s) = W^{(1)} \left(1 - \frac{2}{n}\right)^s + W^{(2)} \left(1 - \frac{4}{n}\right)^s + W^{(3)} \left(1 - \frac{6}{n}\right)^s$$

Using a similar derivation, we can express the correlation length strictly in terms of the nonzero Walsh coefficients.

$$\ell = \sum_{p \neq 0} \frac{n \cdot W^{(p)}}{2p} \qquad (8)$$

We can thus compute the exact autocorrelation function and correlation length in polynomial time by first computing the $O(m)$ nonzero *squared* Walsh coefficients and then summing them up and computing each order-p contribution to the normalized Walsh amplitude $W^{(p)}$ for $p = \{1, 2, \ldots, k\}$.

3. PROBLEM CLASSES

The above computation allows us to compute the correlation structure of a given problem *instance*. Stadler and Happel [16] have also defined the autocorrelation function over *random fields*: formalisms of problem distributions in which parameters are assigned using a statistical model. Many studies of k-satisfiability are performed on such problem distributions in which each instance is generated randomly according to parameters.

Let $\mathfrak{D}(n, m, k)$ be the set of all problems with n variables, m clauses, and k literals per clause generated as follows. For each of the m clauses, we select exactly k unique elements from the set of n variables and negating each with probability $\frac{1}{2}$. Thus the $2n$ possible literals (i.e., variables unnegated or negated) occur with equal probability in each clause. Each instance of $\mathfrak{D}(n, m, k)$ will have its own value for $r(s)$ and ℓ. In this section, we are interested in the *expectation* of the autocorrelation function and correlation length over the entire class of problems generated in this way.

Cheeseman et al. [2] have shown that with this problem class (and several other classes of NP-Hard problems) there is associated an "order parameter" that is ultimately linked to how difficult a problem is for complete search algorithms. For k-satisfiability, this order parameter is the probability of an instance being satisfiable. In particular, the class of problems $\mathfrak{D}(n, m, k)$ can be partitioned into two "phases": an underconstrained phase consisting of problems that have low constrainedness and are almost surely satisfiable, and an overconstrained phase consisting of highly constrained problems that are almost surely unsatisfiable. The degree of constrainedness of a k-satisfiability problem instance depends on the relationship between the number of variables and the number of clauses. Intuitively, an instance with many variables but relatively few clauses will be *underconstrained* since each variable is likely to appear in only a small number of clauses. On the other hand, an instance with relatively many clauses will be *overconstrained* since there are many more conditions to simultaneously satisfy. Thus the constrainedness can be described in a compact way by the *clause-to-variable* ratio $\alpha = \frac{m}{n}$. Instances with low α lie in the underconstrained phase and instances with high α lie in the overconstrained phase.

There exists a *critical value* of α that separates the two phases where the probability of a given instance being satisfiable drops suddenly to zero. Instances from this *phase transition* region have been found to be difficult for complete search algorithms [2, 8]. Instances in the under- and overconstrained phase are "typically" easy whereas instances in the phase transition region are "typically" hard. This phenomenon is called an *easy-hard-easy* pattern. For 3-satisfiability, the critical value has not been precisely determined analytically (though upper and lower bounds exist). Empirical results show that it lies between 4.2 and 4.3 [10, 8]. Finite-size scaling methods show that, as $n \to \infty$, the critical point is near $\alpha \approx 4.26$ [5].

A number of researchers have examined if the easy-hard-easy pattern extends to incomplete *local* algorithms [3, 21, 11]. However, since incomplete algorithms are, in general, unable to efficiently prove an instance is unsatisfiable, these studies are almost always performed on a modified problem class that is obtained by *filtering* instances from $\mathfrak{D}(n, m, k)$ such that all unsatisfiable instances are removed. On a filtered distribution, this easy-hard-easy pattern has been shown to occur for local algorithms as well. For example, the median search cost (in terms of number of moves until a solution is found) for the WALKSAT algorithm on a set of filtered instances is plotted in Figure 2. Note the peak in search cost in the phase transition region near the critical point. On *unfiltered* problems however, the search cost diverges (as measured in terms of time to find a solution with the provably maximal number of satisfied clauses).

3.1 Expectation over random problems

In order to compute the expectation of the autocorrelation function and correlation length over the uniform random k-satisfiability problem distribution $\mathfrak{D}(n, m, k)$, we characterize the squared Walsh coefficients as *random variables* and compute their theoretical expectation over $\mathfrak{D}(n, m, k)$. We then apply a *mean-field approximation* to estimate the expectation of $r(s)$ and ℓ. We denote the expectation of a random variable V over the problem distribution $\mathfrak{D}(n, m, k)$ as $\langle V \rangle$.

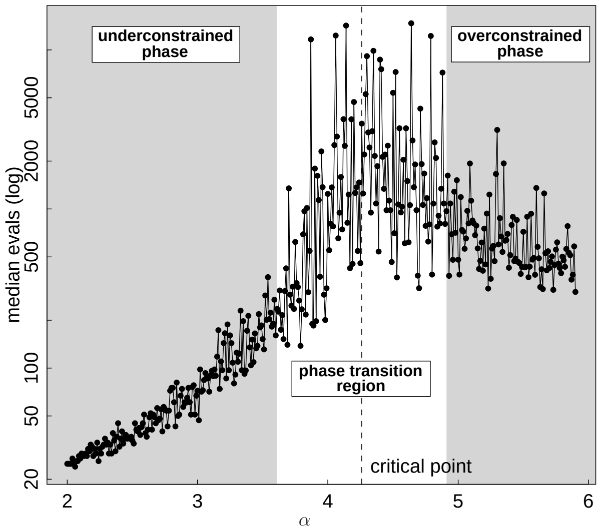

Figure 2: Median evaluations for local search (log scale) as a function of α. This demonstrates the easy-hard-easy pattern through the phase transition on *filtered* instances.

Recall the definition of a Walsh coefficient w_i in the foregoing section. From Equation (5) we have

$$w_i = \sqrt{2^n}\frac{1}{2^k}(L_i^{odd} - L_i^{even}) = \sqrt{2^n}\frac{1}{2^k}(L_i)$$

We now view L_i^{even} and L_i^{odd} as discrete random variables that count clauses in each instance of $\mathfrak{D}(n, m, k)$ that contain the variables selected by bitstring i that have an odd or even number of negations, respectively. We have also introduced the random variable $L_i = L_i^{odd} - L_i^{even}$. We want to compute the expectation of the squared Walsh coefficients.

$$\langle w_i^2 \rangle = \frac{2^n}{4^k}\langle L_i^2 \rangle$$

Since L_i is the difference of clause counts, it is a discrete random variable over the integral domain in the interval $[-m, m]$. Thus we can compute the expectation for L_i^2 in terms of its probability mass function.

$$\langle L_i^2 \rangle = \sum_{z=-m}^{m} z^2 \Pr\{L_i = z\}$$

Note that $L_i = z$ iff $L_i^{odd} = z_1$ and $L_i^{even} = z_2$ and $m - (L_i^{odd} + L_i^{even}) = z_3$ such that $z_1, z_2, z_3 \geq 0$ and $z_1 + z_2 + z_3 = m$ and $z_1 - z_2 = z$. If we know the *probability* of any given clause contributing to the counts represented by z_1, z_2, and $z_3 = m - (z_1 + z_2)$ (and if these probabilities are independent) then we can explicitly compute this probability mass function as a sum over multinomial distributions

$$\Pr\{L_i = z\} = \sum_{z_1, z_2 \geq 0 : z_1 - z_2 = z} \frac{m! \pi_1^{z_1} \pi_2^{z_2} \pi_3^{m-(z_1+z_2)}}{z_1! z_2! (m - (z_1 + z_2))!}$$

where π_1 is the probability that a clause contributes to the L_i^{odd} count, π_2 is the probability that a clause contributes to the L_i^{even} count, and π_3 is the probability that a clause contributes to neither counts.

Over $\mathfrak{D}(n, m, k)$, each clause is generated independently so these probabilities will be independent. For a bitstring i, a single clause contributes to L_i^{odd} (L_i^{even}) if 1) all variables specified by the bitstring i are selected for the clause in the generation process *and* 2) an odd (even) number of them are negated. For any $p = \langle i, i \rangle$ variables, the probability that all p variables specified by i occur in a given clause is the number of valid configurations involving all p variables selected by the bitstring representation of i: $p!\binom{k}{p}$, divided by the probability the particular configuration is selected by the uniform random generation process: $p!\binom{n}{p}$. The probability that any p variables appear together in a clause is

$$\pi' = \binom{k}{p} / \binom{n}{p}$$

Thus, π_1 is π' times the probability that an odd number of them are negated. Since variables are negated with probability $\frac{1}{2}$ in $\mathfrak{D}(n, m, k)$, a clause contributes to L_i^{odd} with probability $\pi_1 = \frac{\pi'}{2}$. Similarly, $\pi_2 = \frac{\pi'}{2}$. The probability that a clause does not contribute to either L_i^{odd} or L_i^{even} is simply the probability that not all variables specified by i occur in the clause. Thus we have $\pi_3 = 1 - \pi'$. This gives the theoretical expectation for squared Walsh coefficients.

3.2 A mean-field approximation over $\mathfrak{D}(n, m, k)$

Over uniformly generated problems, the expectation value $\langle L_i^2 \rangle$ only depends on the *order* of i (the number of one bits in its bitstring representation) since the probability of selecting any $p = \langle i, i \rangle$ variables does not depend on our specific choice of p variables. Let $L^{(p)}$ denote the value of L_i for all $\langle i, i \rangle = p$. By linearity of expectation we have

$$\left\langle \sum_{\langle i,i \rangle = p} w_i^2 \right\rangle = \binom{n}{p} \frac{2^n}{4^k} \sum_{z=-m}^{m} z^2 \Pr\{L^{(p)} = z\}$$

If we assume that statistical fluctuations are small, then we can approximate the expectation of a function of random variables with the function of the expectation of random variables (called a mean-field approximation in [19]). Under this assumption, we calculate the following approximation.

$$\langle W^{(p)} \rangle = \langle \sum_{\langle i,i \rangle = p} w_i^2 / \sum_{j \neq 0} w_j^2 \rangle \simeq \langle \sum_{\langle i,i \rangle = p} w_i^2 \rangle / \langle \sum_{j \neq 0} w_j^2 \rangle$$

and compute the expectation values for $\langle r(s) \rangle$ and $\langle \ell \rangle$ by substituting the approximated expectation value $\langle W^{(p)} \rangle$ into the following equations. In particular, the expectations of the autocorrelation function and correlation length over problems generated uniformly at random are given by

$$\langle r(s) \rangle = \sum_{p \neq 0} \langle W^{(p)} \rangle \left(1 - \frac{2p}{n}\right)^s \quad (9)$$

$$\langle \ell \rangle = \sum_{p \neq 0} \frac{n \cdot \langle W^{(p)} \rangle}{2p} \quad (10)$$

where $\langle W^{(p)} \rangle$ is computed using the approximation in (9).

4. NUMERICAL RESULTS

To determine the accuracy of the mean-field approximation we generated 391 instances uniformly at random from

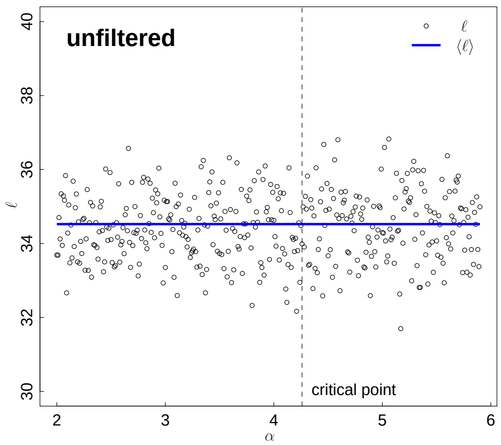

Figure 3: ℓ and $\langle \ell \rangle$ over unfiltered distribution.

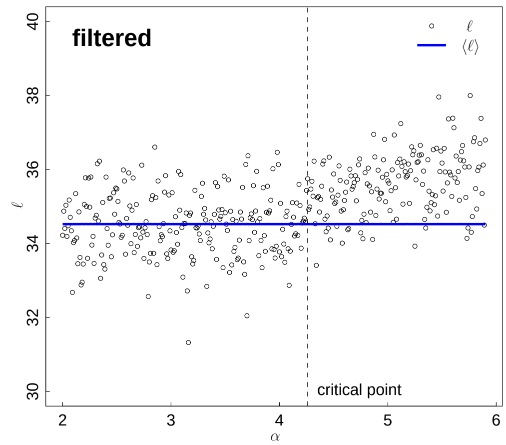

Figure 4: ℓ and $\langle \ell \rangle$ over filtered distribution.

the distribution $\mathfrak{D}(100, m, 3)$ varying m to obtain a clause-variable ratio $\alpha = \frac{m}{n}$ from 2.00 to 5.90 at steps of 0.01. To examine the correlation structure on *filtered* instances, we repeated the procedure with the modification that, for each value of α, a problem instance would be accepted if it is found to be satisfiable using a complete search algorithm. Thus all instances in the filtered distribution are satisfiable. For each generated instance we computed the squared Walsh sums of each order.

The expectation values for order-p squared Walsh sums

$$\sum_{\langle i,i \rangle = p} \langle w_i^2 \rangle$$

are related linearly to α, i.e., orders 1 and 2 give a value of 4.6875α and order 3 gives a value of 1.5625α. Since α cancels in Equation (9), for fixed n, $\langle r(s) \rangle$ and $\langle \ell \rangle$ are α-invariant across $\mathfrak{D}(n, m, k)$. In particular, we calculate $\langle \ell \rangle = 34.5238$ for $\mathfrak{D}(100, m, 3)$ as discussed above.

Given the squared Walsh sums for each instance, we can compute the exact correlation length given by Equation (8) and compare this to $\langle \ell \rangle$. The results are plotted in Figures 3 and 4. The *critical point* of $\alpha = 4.26$ is marked on the plots. The α-invariance of the mean-field approximation of Equation (9) is reflected in the steady behavior of $\langle \ell \rangle$ on the unfiltered problems. To quantify the correspondence between empirical and theoretical expectation we calculate the absolute deviation between $\langle \ell \rangle$ and the empirical mean of ℓ which we denote $\mu(\ell)$ for the subcritical and supercritical phases (points below and above the critical point, respectively). The empirical means $\mu(\ell)$ for each set and phase along with the absolute deviation of $\mu(\ell)$ from the theoretical expectation value are reported in Table 1. On the unfiltered set we see relatively small deviations in both phases, suggesting the correspondence is tight. On the filtered set, we see a marked divergence from the expectation value in the supercritical phase. This trend can also be seen in Figure 4. The results for $r(s)$ are analogous, but omitted in the interest of brevity.

To examine this trend more carefully, we observe the expectation values for the sum of squared Walsh coefficients (where the mean-field approximation is no longer necessary). For each sum of squared Walsh coefficients of order p, we compute the deviation from expectation

$$\delta = \sum_{\langle i,i \rangle = p} w_i^2 - \left\langle \sum_{\langle i,i \rangle = p} w_i^2 \right\rangle$$

We find a divergence from expectation *only* in the first-order squared Walsh coefficients for the filtered problems (see Figure 5). Recall the squared Walsh coefficients expectation has the following proportionality

$$\langle w_i^2 \rangle \propto \langle (L_i^{odd} - L_i^{even})^2 \rangle$$

Since each such w_i is *first-order* (that is, $\langle i,i \rangle = 1$), the random variable L_i^{odd} is counting the number of clauses in which the single *particular variable* specified by i appears negated. Similarly, the random variable L_i^{even} counts the number of clauses in which the particular variable specified by i appears unnegated. On the unfiltered distribution, each variable is negated with *a priori* probability 1/2.

Thus, the above deviation from expectation on filtered instances suggests that for each variable, the absolute value of the difference between negated and non-negated appearances of variables is diverging. The implication is that, on the filtered set, the *a posteriori* negation probability is no longer necessarily equal to our assumption of 1/2. In other words, filtering for satisfiable instances induces a selection bias toward problems with non-uniform negation probability. The bias emerges at the critical point and intensifies in the overconstrained phase.

Our results also show that very different correlation structure begins to emerge in problems that have been filtered for satisfiability. We also conjecture that other selection biases arise from the filtering process that are ultimately related to problem hardness. Empirically, problem hardness depends on this filtering process. For example, Xu et al. [20] have shown that problem hardness is easier to model if the distribution is filtered for (un)satisfiability. Furthermore, these hardness models tend to have vast differences depending on whether they are trained on models filtered for satisfiability versus unsatisfiability.

5. CONCLUSION

The contributions of this paper are threefold. First, we have shown how to exactly compute correlation structure

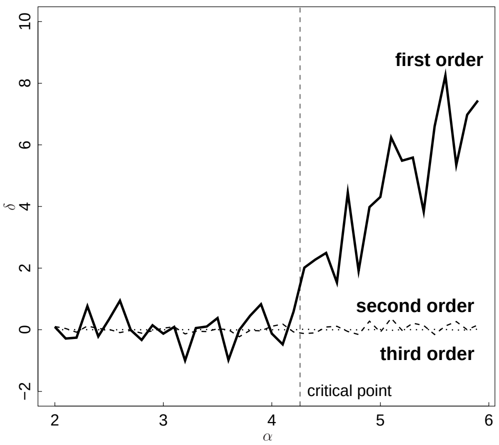

Figure 5: Deviation δ from expectation of sum of squared Walsh coefficients on filtered set.

set	phase	$\mu(\ell)$	$\sigma(\ell)$	$\lvert\mu(\ell) - \langle\ell\rangle\rvert$
unfiltered	subcritical	34.402	0.876	0.121
	supercritical	34.495	0.979	0.028
filtered	subcritical	34.498	0.861	0.026
	supercritical	35.614	0.826	1.090

Table 1: ℓ on $\mathfrak{D}(100, m, 3)$: mean μ, standard deviation σ, and absolute deviation of empirical mean from theoretical expectation.

as measured by $r(s)$ and ℓ in polynomial time on any k-satisfiability instance. Second, we can approximate the *expectation* of these quantities over the uniform random distribution of problems. Our approximation agrees well numerically on the unfiltered distribution. We have also shown that, on unfiltered problems, the expectation of ℓ does not depend on the clause to variable ratio α. This is an interesting result since local search cost empirically depends on α [3], suggesting that search cost depends on deeper structural features than simple correlation structure.

Finally, we have uncovered a selection bias on filtered problems showing that selecting problems for satisfiability, as is typically done in local search studies, impels problem instances toward non-uniform negation probabilities which is detected by the first-order Walsh coefficients.

6. ACKNOWLEDGMENTS

This research was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number FA9550-08-1-0422. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

7. REFERENCES

[1] E. Angel and V. Zissimopoulos. On the classification of NP-complete problems in terms of their correlation coefficient. *Discrete Applied Mathematics*, 99:261–277, 2000.

[2] P. Cheeseman, B. Kanefsky, and W. M. Taylor. Where the Really Hard Problems Are. In *Proc. of IJCAI-1991* Sydney, Australia, pages 331–337, 1991.

[3] D. A. Clark, J. Frank, I. P. Gent, E. MacIntyre, N. Tomov, and T. Walsh. Local search and the number of solutions. In *CP-1996*, pages 119–133, 1996.

[4] M. Eigen, J. McCaskill, and P. Schuster. Molecular quasispecies. *J. of Physical Chemistry*, 92(24):6881–6891, Dec 1988.

[5] A. K. Hartmann and M. Weigt. *Phase Transitions in Combinatorial Optimization Problems*. Wiley-VCH, 2005.

[6] H. H. Hoos, K. Smyth, and T. Stützle. Search space features underlying the performance of stochastic local search algorithms for MAX-SAT. In *Proc. of PPSN-8*, pages 51–60, Sept. 2004. Springer.

[7] W. Hordijk and P. F. Stadler. Amplitude spectra of fitness landscapes. *J. of Complex Systems*, 1:39–66, 1998.

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

[9] P. Merz and B. Freisleben. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE-EC*, 4(4):337–352, November 2000.

[10] D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of SAT problems. In *Proc. of AAAI-1992*, 1992.

[11] A. J. Parkes. Clustering at the phase transition. In *Proc. of AAAI-1997*, 1997.

[12] S. Rana, R. B. Heckendorn, and L. D. Whitley. A tractable Walsh analysis of SAT and its implications for genetic algorithms. In *Proc. of AAAI-1998*, pages 392–397, 1998.

[13] J. Singer, I. P. Gent, and A. Smaill. Backbone fragility causes the local search cost peak. *JAIR*, 12:235–270, 2000.

[14] P. F. Stadler. Toward a theory of landscapes. In R. Lopéz-Peña, R. Capovilla, R. García-Pelayo, H. Waelbroeck, and F. Zertruche, editors, *Complex Systems and Binary Networks*, pages 77–163. Springer Verlag, 1995.

[15] P. F. Stadler. Landscapes and their correlation functions. *J. of Mathematical Chemistry*, 20:1–45, 1996.

[16] P. F. Stadler and R. Happel. Random field models for fitness landscapes. *J. of Mathematical Biology*, 38(5):435–478, 1999.

[17] P. F. Stadler and W. Schnabl. The landscape of the traveling salesman problem. *Physics Letters A*, 161:337–344, 1992.

[18] E. D. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63:325–336, 1990.

[19] C. Williams and T. Hogg. Exploiting the deep structure of constraint problems. *Artificial Intelligence*, 70:73–117, 1994.

[20] L. Xu, H. H. Hoos, and K. Leyton-Brown. Hierarchical hardness models for SAT. In *In CP-2007*, pages 696–711, 2007.

[21] M. Yokoo. Why adding more constraints makes a problem easier for hill-climbing algorithms: Analyzing landscapes of CSPs. In *CP-1997*, pages 356–370, 1997.