Estimating Bounds on Expected Plateau Size in MAXSAT Problems*

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

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

Abstract. Stochastic local search algorithms can now successfully solve MAXSAT problems with thousands of variables or more. A key to this success is how effectively the search can navigate and escape plateau regions. Furthermore, the solubility of a problem depends on the size and exit density of plateaus, especially those closest to the optimal solution. In this paper we model the plateau phenomenon as a percolation process on hypercube graphs. We develop two models for estimating bounds on the size of plateaus and prove that one is a lower bound and the other an upper bound on the expected size of plateaus at a given level. The models' accuracy is demonstrated on controlled random hypercube landscapes. We apply the models to MAXSAT through analogy to hypercube graphs and by introducing an approach to estimating, through sampling, a key parameter of the models. Using this approach, we assess the accuracy of our bound estimations on uniform random and structured benchmarks. Surprisingly, we find similar trends in accuracy across random and structured problem instances. Less surprisingly, we find a high accuracy on smaller plateaus with systematic divergence as plateaus increase in size.

1 Introduction

The success of stochastic local search algorithms on satisfiability problems is attributed in part to their exploitation of equal or "sideways" moves in the search neighborhood [1]. In many cases, this strategy results in an empirical improvement in generated solutions [2] and a theoretical improvement in the approximation ratio on special cases [3]. Accepting equal moves can result in "plateau behavior" of search [4]: potentially long epochs during which any discrete "gradient" information is absent, and search algorithms must either perform a random walk on the plateau or attempt to search it systematically until an improving move is found.

* 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.

T. Stützle, M. Birattari, and H.H. Hoos (Eds.): SLS 2009, LNCS 5752, pp. 31–45, 2009.

The two characteristics that determine the hardness of escaping a plateau are its *exit density*: the number of strictly improving moves incident to plateau solutions, and its *size*: the number of solutions belonging to the plateau. Since the progress of a stochastic local search algorithm is ultimately connected to how well it can escape plateaus, plateau characteristics are intimately related to problem difficulty for local search [4,5,6].

Not all plateaus contain exits. In the worst case, the entire plateau must be expanded before determining whether it is escapable or not. On a MAXSAT problem with n variables and m clauses there must exist a set of equal value solutions (not necessarily connected) that has cardinality $\Omega\left(\frac{2^n}{m}\right)$. This means that plateaus tend to be intractable to enumerate on average. Methods to determine plateau size need to be extremely efficient and not rely on enumeration.

In this paper we take a first step in predicting plateau characteristics for problem instances by focusing on plateau size. We develop methods for estimating upper and lower bounds on the expected plateau size in MAXSAT problems. Such bounds can benefit search algorithms in two ways: first by providing an estimate of how hard a problem instance is likely to be for stochastic local search, and second by predicting when the expected size of a plateau is likely to be too large to systematically search.

Under some simplifying assumptions on the distribution of equal valued solutions in the search space, we construct a correspondence between *plateaus* in MAXSAT problems and *percolation clusters* in hypercube graphs. We present models for bounding the expected size of plateaus from above and below. Furthermore, we introduce a method for estimating the probability that nearby points belong to the same level set by locally sampling the region of a point.

We find that the trends in accuracy for prediction are surprisingly uniform across random and structured problem sets. As we expected, the lower bound diverges in a consistent manner with respect to plateau size due to an approximation term in the prediction expression.

1.1 Related Work

Hampson and Kibler [5] empirically investigated the plateau behavior of local search on satisfiability problems. They discovered that many plateaus at high evaluation levels were intractably large and restarting was more beneficial than extensive plateau search in some cases. They found the exit density of plateaus is inversely proportional to the number of variables, and conjectured that the expected time to search these plateaus would increase linearly in the problem size. Most importantly, they found that the size of plateaus increased exponentially in the number of variables.

Frank et al. [4] studied the properties of plateau regions across several classes of MAXSAT problem. They used GSAT to locate solutions at the top evaluation levels and performed breadth-first search to exhaustively expand the plateaus to which each solution belonged. They collected statistics on the distribution of plateaus with and without exits. They found that different problem classes may be harder for local search because plateau characteristics differ across such classes.

In a more general setting, Hoos and Stützle [7] extended the plateau concept to general combinatorial search spaces and defined metrics for plateau characteristics (e.g., width). They developed plateau connection graphs: directed acyclic graphs that capture connectivity between plateau regions and associated transition probabilities.

Smyth [8] examined plateau characteristics for uniform random 3-SAT instances. He found that solutions on lower level sets tended to cluster together in one common large plateau where solutions on better level sets belonged to many smaller plateaus. He also studied the internal structure of plateau regions, finding that the graphs had very low branching factors and diameter greater than or equal to the number of variables.

Plateaus emerge in the presence of *neutrality*: the existence of neighboring states with equal evaluation. Reidys and Stadler [9] studied the nature of neutrality and developed an additive random model on which neutrality can be expressed as a random variable. They derived a probability mass function for the length of *neutral walks*: monotonic random paths of equal valued states which we will employ in this paper. Reidys and Stadler extended work originally done on RNA landscapes [10] where *neutral networks*, induced subgraphs of a landscape, are studied using the theory of random graphs.

2 Size Prediction

A combinatorial search problem is defined as a set X of candidate solutions and an objective function $f : X \to \mathbb{R}$ that assigns some value to each element of X. The solution set X for satisfiability problems is the set of true/false assignments to n variables which can be characterized as the set of strings $\{0,1\}^n$. For MAXSAT, the objective function f counts the number of satisfied clauses given by a particular solution x.

A local search algorithm defines some computationally tractable neighborhood function $N : X \to 2^X$ and, starting from an independently generated initial candidate solution, walks along the graph induced by the neighborhood function. That is, if $x \in X$ is the current candidate solution, in each iteration a new element $y \in N(x)$ is selected to become the new candidate solution according to a pivot rule. The behavior of local search can be characterized as a biased walk on the neighborhood graph $G(X,E)$ induced by N, that is, $(x,y) \in E \iff y \in N(x)$.

For MAXSAT problems, the seemingly most natural neighborhood N maps solutions to their set of Hamming neighbors: solutions that differ in exactly one variable. Thus $G(X,E)$ is isomorphic to a hypercube graph of order n. Since we are concerned only with the MAXSAT domain in this paper, we hereafter work only with this graph. The Hamming distance between two solutions x and y is denoted as $d(x,y)$ and represents the minimal distance between x and y on the hypercube.

Let $L \subseteq X$ be a maximal set of solutions such that $\forall x \in L, f(x) = \ell$. We refer to L as the *level set* at level ℓ. A neutral path $\mathcal{N}(x,y)$ in G is a sequence of distinct solutions $(x = x_1, x_2, \ldots, x_k = y)$ such that, for all $i \in \{1, \ldots, k-1\}$, the following conditions hold.

1. $x_{i+1} \in N(x_i)$
2. $f(x_i) = f(x_{i+1})$

A plateau is a *maximal* set P such that for all $x, y \in P$, $\exists\ \mathcal{N}(x, y)$. Thus a plateau is simply a connected component of the subgraph of the neighborhood graph G induced by a level set, and the set of all plateaus form a partition of G. The *size* of a plateau P is defined as its cardinality $|P|$. Note that our definition allows for $|P| = 1$. In other words, the set of all plateaus partition the search space X, and a vertex with no equal neighbors is a degenerate plateau. This definition is analogous to that in other studies, e.g., [4,7,8].

2.1 Estimating a Lower Bound: Hamming Path Set

We define a neutral Hamming path $\mathcal{N}_{\mathcal{H}}(x, y)$ in G between two solutions x and y is a particular case of a neutral path $\mathcal{N}(x, y) = (x = x_1, \ldots, x_k = y)$ with the added monotonicity constraint that $d(x, x_{i+1}) = d(x, x_i) + 1$ for all $i \in \{1, \ldots, k-1\}$.

Let x be an arbitrary solution in a plateau P. We define the Hamming path set H_x associated with x as

$$H_x = \{y \in P : \exists\ \mathcal{N}_{\mathcal{H}}(x, y)\}$$

Clearly, $H_x \subseteq P$ and thus $|H_x| \leq |P|$.

On a particular problem instance, we can consider H_x taken over all randomly selected $x \in X$. We can thus characterize $|H_x|$ as a random variable. Denote as $\mathbb{E}[|H_x|]$ its *expected value*. By linearity of expectation we have

$$\begin{aligned} \mathbb{E}[|H_x \cup (P \setminus H_x)|] &= \mathbb{E}[|P|] \\ \mathbb{E}[|H_x|] &\leq \mathbb{E}[|P|] \end{aligned}$$

In practice, the magnitude of the difference between the left hand and right hand side of the above relation will ultimately depend on our choice of $x \in P$.

Under a simplifying assumption which we will make in the following section, we will find that the probability that a solution belongs to H_x depends only on its distance from x. Denote as

$$h_x(r) = \Pr\{y \in H_x\} \text{ for any } y : d(x, y) = r$$

the probability that a solution y at distance r from x belongs to H_x. On the hypercube of order n, there are $\binom{n}{r}$ solutions at distance r from an arbitrary vertex. Thus we derive the expected size of the Hamming path set (and therefore our lower bound on plateau size) as

$$\mathbb{E}[|H_x|] = \sum_{r=0}^{n} \binom{n}{r} h_x(r) \tag{1}$$

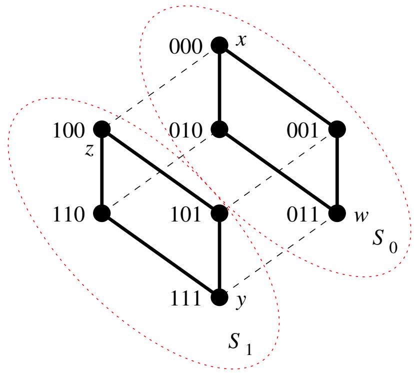

Fig. 1. Partitioning of a hypercube into S_0 and S_1

We develop an estimate of $h_x(r)$ (and so $\mathbb{E}[|H_x|]$) using a percolation approach. Let C_n be a hypercube graph of order n. Each vertex in C_n corresponds to a string $\{0,1\}^n$. Let $x = (000\ldots0)$ and $y = (111\ldots1)$. We refer to x and y as the *corner* vertices. A vertex is *active* if it belongs to the same level set as x. We define the *concentration* as the probability p that a vertex is on the same level set as x, and thus active. We assume this probability is constant and independent across all vertices. In other words, p depends only on the level set under consideration. Note that x is a *fixed active* vertex since it trivially belongs to its own level set. We say the cube *percolates from y to x* if there is a monotonic path $(y = x_1, x_2, \ldots, x_k = x)$ such that all x_i are active.

Let $c(n,p)$ denote the probability that C_n percolates with concentration p from y to the fixed active vertex x.

Proposition 1. *For some real number* $0 \leq g(n) \leq 1$

$$c(n,p) = p \cdot \left(2 \cdot c(n-1,p) - c(n-1,p)^2\right) + g(n)$$

Proof. We partition the vertices C_n into two disjoint sets S_0 and S_1. S_0 consists of the vertices represented by the strings $(0 * * \ldots *)$. S_1 consists of the vertices represented by the strings $(1 * * \ldots *)$.

Note that S_0 and S_1 form hypercubes (see Figure 1). Each subcube shares one of its corner vertices with C_n. In the case of S_0, one of its corner vertices is x, while the opposite corner is a vertex $w = (0,1,1,\ldots,1)$. In the case of S_1, one of its corner vertices is y and the opposite corner is a vertex $z = (1,0,0,\ldots,0)$.

All percolating paths from y to x must pass from S_1 to S_0 exactly once and cannot pass back from S_0 to S_1 (since at each step of the path, the number of ones in the bitstring must decrease by exactly one).

We refer to paths that pass from S_1 to S_0 through edges (y,w) or (z,x) as *external crossing paths*. We refer to the remaining paths as *internal crossing paths*. Let E_{ex} be the event that at least one external crossing path percolates from y to x. Let E_{in} be the event that at least one internal crossing path percolates.

If y is inactive, then C_n does not percolate from y to x. Now suppose y is active. The probability that S_0 percolates from w to the fixed active vertex x is $c(n-1,p)$. Now we consider S_1. Note that $x \notin S_1$, but y takes the role of the fixed active vertex (since we have assumed it is active). The probability of S_1 percolating from z to y is $c(n-1,p)$.

Percolation is direction invariant. Thus if S_1 percolates from z to y, there is a percolating external crossing path from y to x through the edge (z,x), and thus C_n percolates. Similarly, if S_0 percolates from w to x, since we have assumed y is active there is a percolating external crossing path from y to x through the edge (y,w). Thus if either S_0 or S_1 percolate, then C_n must percolate. These events are not mutually exclusive, so the probability that there is a percolating external crossing path through either subcube is $2 \cdot c(n-1,p) - c(n-1,p)^2$. We multiply this expression by p, the probability that y is active, to obtain the probability that C_n percolates via an external crossing path.

$$\Pr(E_{ex}) = p \cdot \left(2 \cdot c(n-1,p) - c(n-1,p)^2\right) \tag{2}$$

Now consider the internal crossing paths. Clearly we have,

$$\Pr(E_{in}) - \Pr(E_{ex} \cap E_{in}) = g(n) \tag{3}$$

where $0 \leq g(n) \leq 1$ is a real number that depends on n. The probability that C_n percolates can be expressed as $\Pr(E_{ex}) + \Pr(E_{in}) - \Pr(E_{ex} \cap E_{in})$. Substituting Equations (2) and (3) gives the result. □

We thus ignore the internal crossing paths and bound the percolation probability.

Corollary 1. *Since* $0 \leq g(n) \leq 1$, $c(n,p) \geq p \cdot (2 \cdot c(n-1,p) - c(n-1,p)^2)$

We define $\hat{c}(n,p)$ as the lower bound on $c(n,p)$:

$$\begin{aligned} \hat{c}(1,p) &= p \\ \hat{c}(n,p) &= p \cdot \left(2 \cdot \hat{c}(n-1,p) - \hat{c}(n-1,p)^2\right) \end{aligned} \tag{4}$$

The above result allows us to place a lower bound on $\mathbb{E}[H_x]$.

Proposition 2. *Let x be an arbitrary solution in X. Suppose that for each element $y \in X$, $\Pr\{f(y) = f(x)\} = p$. Then $h_x(d(x,y)) = c(d(x,y),p)$.*

Proof. This follows directly from the definition of $c(n,p)$. Note that a vertex in a Hamming path from y to x must lie in the subcube of order $d(x,y)$ between x and y. If a vertex in the subcube is on the same level set as x, it is considered active. Since each vertex is active with probability p, a neutral Hamming path is simply a percolating path in the subcube of order $d(x,y)$. □

Thus, we have

$$\mathbb{E}[|H_x|] \geq \sum_{r=0}^{n} \binom{n}{r} \hat{c}(r,p) \tag{5}$$

If we know p for a particular level set, then we can bound the expected plateau size. Note that our premise that the concentration parameter p is independent across a given level set is a rather heavy simplifying assumption. In fact, we would expect in practice that solutions have distinct correlations among them. However, this assumption makes the analysis easier.

Finally, the elimination of the $g(n)$ term in the approximation expression will cause the lower bound to diverge as $n \to \infty$ since the approximation loses accuracy for each value of n. Thus, we expect the error to have superlinear growth with n since $g(n)$ is proportional to subcube size.

2.2 Estimating an Upper Bound: Bethe Lattice Approximation

We have characterized plateaus as connected clusters of active sites in the hypercube graph. In this section we will use an exact result from percolation theory to derive an upper limit on the expectation of plateau size for certain values of p. The Bethe lattice (or Cayley tree) of coordination number n is defined as a connected acyclic graph in which each vertex is connected to n neighboring vertices.

For a given concentration p, the expected size of connected clusters of active sites in the Bethe lattice will always be greater than or equal to the expected size of clusters of active sites in the hypercube graph. This can be shown by a simple counting argument. Since the Bethe lattice is acyclic, every site in the cluster rooted at a site b has exactly one path of active vertices to b. Thus the expected number of neighbors of a cluster site that extend the cluster a step further from b is $p \cdot (n-1)$. On the other hand, a vertex in the hypercube graph that belongs to a cluster rooted at some vertex x will have *at least* one path of active vertices to x since cycles are possible. The expected number of neighbors that extend the cluster further is therefore less than or equal to $p \cdot (n-1)$. Thus the expected size of connected active clusters in the Bethe lattice for a particular p is an upper bound on the expected plateau size in the hypercube graph.

The expected cluster size on the Bethe lattice has an exact solution. Let b be an arbitrary active site in the lattice. Let T be the expected size of the clusters rooted at each neighbor of b. By the symmetries of the lattice we have

$$T = p\,(1 + (n-1)T)$$

Solving for T we have $T = \frac{p}{1-(n-1)p}$ and the expected cluster size at arbitrary b is $1 + nT$:

$$\frac{1+p}{1-(n-1)p} \tag{6}$$

Since the Bethe lattice is an infinite system, its value as an approximation of the finite hypercube becomes poorer as p gets larger. In fact, there is a singularity in Equation (6) when $p = \frac{1}{n-1}$. This corresponds to the critical point at which an infinite cluster appears in the lattice and expected cluster size is no longer well-defined. Thus the Bethe lattice approximation is only valid in the subcritical region: values of p strictly less than $\frac{1}{n-1}$. A useful introduction to percolation theory can be found in [11].

2.3 Estimating Concentration: Neutral Walk Method

Except in synthetic cases, the concentration parameter p will not be known *a priori.* Thus we must determine a method to estimate p. One approach might be to simply sample points on the landscape until the proportion of solutions that belong to a particular level set is accurately represented. However, this approach is insufficient for the following reasons.

1. It may take exponential time to obtain an accurate estimate of the true proportion for smaller level sets.
2. The actual concentration p is likely to be correlated with distance. For example, in MAXSAT, solutions at Hamming distance one are more likely to be on the same level set than solutions an arbitrary distance away.

To address these points, we develop a method that uses a *neutral walk*: a polynomial time algorithm that locally samples around a solution. The concept of a neutral walk was introduced by Schuster et al. [12] to measure the extent of plateaus (which they refer to as components of a neutral network) for RNA landscapes. A neutral walk is defined as a random walk of monotonically increasing distance from a reference vertex such that all walk vertices have the same evaluation. On the hypercube, there can be at most n increasing steps, each with a neighborhood size that is $O(n)$ in the worst case. Thus the time to perform a neutral walk is bounded above by $n\sum_{i=1}^{n} i = O(n^3)$.

The probability mass function of neutral walk length $\mathcal{L}$ was derived by Reidys and Stadler [9]. We adopt a specialization for the hypercube. Let p be the probability that a solution belongs to the same level set as the origin of the walk. A vertex at distance r from the walk's origin has $n-r$ neighbors at distance $r+1$. Thus the probability that a walk can be extended to distance r is $\prod_{i=1}^{r}\left[1-(1-p)^{n-(i-1)}\right]$. The probability that the vertex at distance r terminates the walk is $(1-p)^{n-r}$. Hence, given concentration p, the probability that a neutral walk is of length r can be written as

$$\Pr\{\mathcal{L}=r\} = (1-p)^{n-r}\prod_{i=1}^{r}\left[1-(1-p)^{n-(i-1)}\right]$$

We use this result to compute the expected neutral walk length as follows.

$$\mathbb{E}_p[\mathcal{L}] = \sum_{r=1}^{n} r\Pr\{\mathcal{L}=r\} \tag{7}$$

To estimate p for a level set L, we compute the empirical mean neutral walk length $\mathcal{L}_\mu$ by performing a number of neutral walks from sampled points on L. If we assume $\mathcal{L}_\mu$ accurately estimates $\mathbb{E}_p[\mathcal{L}]$, then an estimate of the concentration is simply the root of the monotonic function

$$\mathbb{E}_p[\mathcal{L}] - \mathcal{L}_\mu$$

using Equation (7) parameterized by p. We use a numerical root finding algorithm to solve for p, giving us the estimate.

3 Computational Experiments

We have proved that, given our assumptions, our models provide upper and lower bounds. However, we do not know how well the models perform on actual problems where the concentration is not known. We evaluate the accuracy of our prediction bounds by exhaustively enumerating plateaus on a number of different search landscapes and comparing the actual value with the predictions given by Equations (5) and (6). To assess the accuracy and trends of the prediction we first use synthetic landscapes on which concentration is known, and then both random and structured MAXSAT landscapes on which we predict the concentration using the neutral walk method. Because we need to fully enumerate the plateaus for accuracy, we are limited to small problems in this analysis.

3.1 Concentration-Controlled Random Landscapes

To test the size prediction bounds given known concentrations, we evaluate predictions for random hypercube landscapes on which we explicitly control concentration. In particular, given a hypercube landscape X, we assign each solution an objective function value of 1 with probability p and a value of 0 with probability $1 - p$. We sample solutions at random on the landscape. If the solution is of value 1, we expand its plateau using breadth-first search. We also compute its Hamming path set. We compare the actual cardinalities with the prediction equation and the Bethe lattice approximation for concentrations that lie in the subcritical region.

We generate 100 random landscapes controlling for concentration from $p = 0.01$ to $p = 0.4$. On each landscape we calculate the Hamming path set lower bound and the Bethe upper bound using the known value of p. We sample 100 random points from the level set at value 1 and perform breadth-first search to exhaustively enumerate the plateaus. We also perform a depth-first search from each plateau vertex back to the root to enumerate the Hamming path set. We compare the average plateau and Hamming path set sizes with the prediction bounds.

We report our prediction data in the form of correlation plots. There are three types of data points. "Plateau/HP" is actual plateau size vs. Hamming path prediction. "HP/HP" denotes actual Hamming path set size vs. Hamming path prediction. "Plateau/Bethe" denotes actual plateau size vs. Bethe prediction. A perfect prediction would lie on the diagonal line included in the plots. Data for a 20 dimensional random landscape are plotted in Figure 2. The low number of "plateau/Bethe" points are because the higher concentrations exceed the critical value for the Bethe lattice.

To determine the accuracy of our concentration estimate, we run the above experiments again and estimate p using the neutral walk method. Instead of using

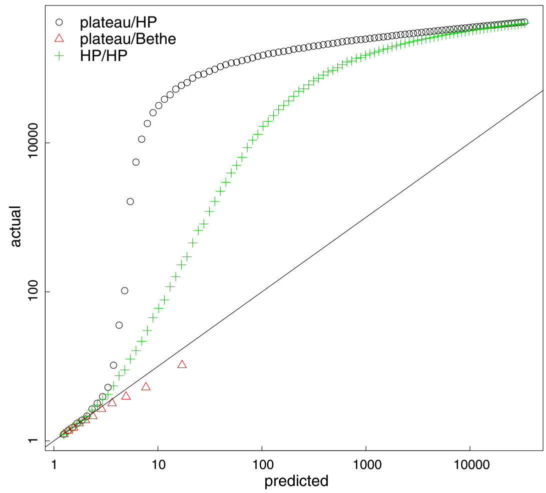

Fig. 2. Log-log plot of predictions on 20 dimensional random landscape

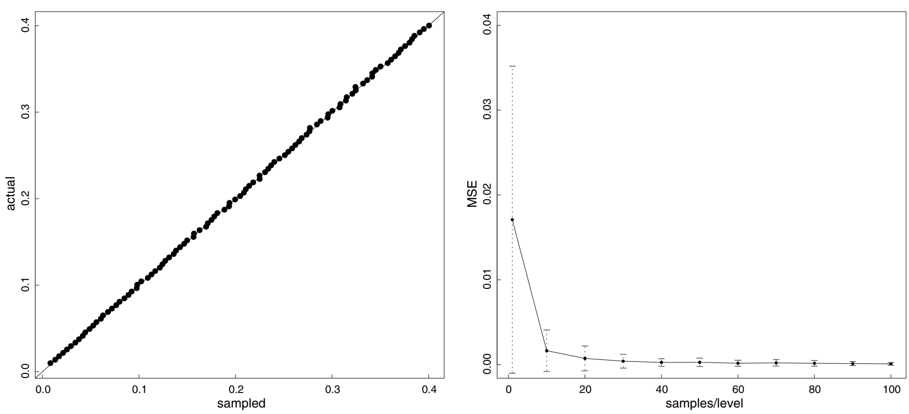

Fig. 3. Actual p vs. estimated p (left). Mean squared error between actual and estimated p vs samples/level set (right).

the known p value for the prediction bounds, we take 10 neutral walks from each of the 100 sampled points and predict the concentration with the resulting walk lengths. We compare the actual p values used to generate the landscape with the values estimated by the neutral walk method. These data are plotted on the left in Figure 3. We find a tight correlation between the predicted and actual concentrations. To determine how much effort needs to be expended to estimate p, we plot the mean squared error between known concentration and estimated concentration with respect to samples per level set on the right in Figure 3. Both plots were generated using data from the 20 dimensional random hypercube.

The high accuracy of the p estimation with low sample size is encouraging because the time to predict the size of the plateau for a *single* solution (including neutral walk sampling) is on the order of 200-5000 microseconds whereas measuring the actual plateau can take several minutes or longer on the relatively small problems we investigated.

3.2 MAXSAT Landscapes

To test how well the bounds transfer to actual problems, we perform experiments on random and structured MAXSAT problems. On MAXSAT the objective function is the number of formula clauses satisfied. On uniform random problems, most solutions belong to a small number of objective function values. This typically results in solutions of average value belonging to vast plateaus. Hampson and Kibler [5] found that, due to their relatively high exit density, plateaus of average value are easy for local search to escape, and thus local search is most affected by plateaus of higher value. Therefore we follow the technique used by Frank et al. [4] and Smyth [8] employing a stochastic local search algorithm (WalkSat [13]) to sample the highest value plateaus in the search space.

Plateau measurement time depends on the number of vertices on the plateau. Thus large plateaus quickly become intractable to enumerate as they grow with depth and problem size. Some level sets can have a small number of extremely large plateaus which cannot be enumerated in a reasonable amount of time. Rather than omitting these data points (which would bias the results to make a lower bound appear tighter than it actually is) we only report the top three level sets for two benchmark sets.

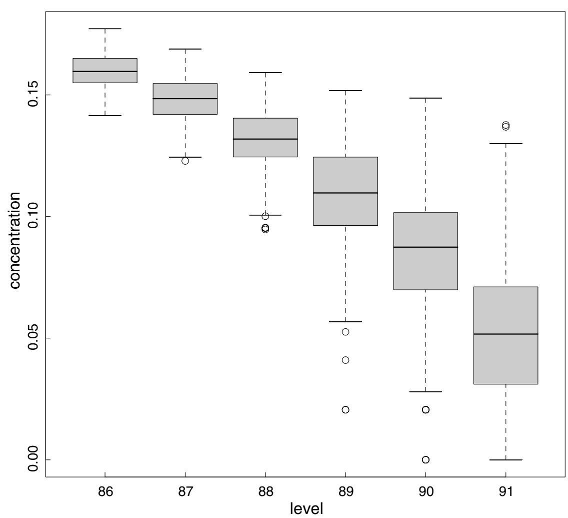

Fig. 4. Estimated concentration with respect to level set on `uf20-91` problems

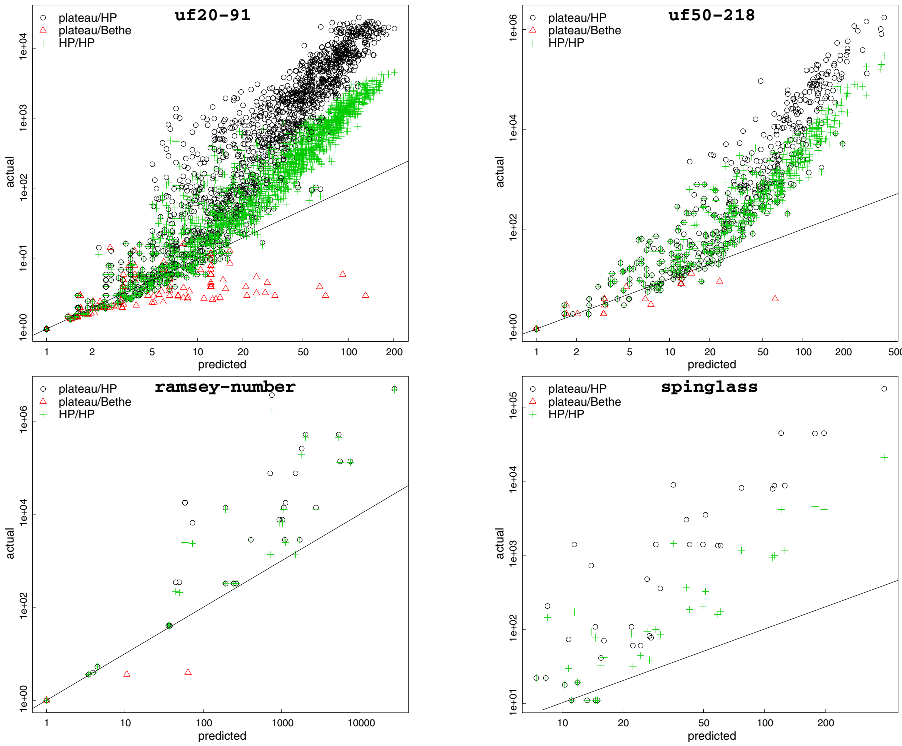

Fig. 5. Predictions for MAXSAT problems. Results on random uniform sets: 20 variables and 91 clauses `uf20-91`, and 50 variables and 218 clauses `uf50-218` appear on the top; results on structured problem instances are plotted on the bottom.

We use two uniform random benchmark distributions from SATLIB: `uf20-91`, a 20 variable 91 clause set, and `uf50-218`, a 50 variable 218 clause set. We select 100 random instances from each set and perform WalkSat to generate 100 solutions each on the top six levels (for `uf20-91`) and the top three levels (for `uf50-218`). Note that all instances in these sets are satisfiable.

We estimate p for each level set by using the neutral walk method, taking 10 walks from each sampled solution. From each solution we exhaustively enumerate its plateau and its Hamming path set and compare the actual sizes to the bounds in Equations (5) and (6). The results from `uf20-91` are plotted in the top left of Figure 5. The results from `uf50-218` are plotted in the top right of Figure 5. Note the trends in accuracy when compared to each other and to the random landscape (see Figure 2).

We report the estimated p values found by the neutral walk method on the 20 variable uniform random SAT problem in Figure 4. The estimated concentration on each problem set of a particular size appears to decrease as a function of evaluation. This reflects the empirical decrease in plateau size with respect to level found originally by Hampson and Kibler [5] and later by Frank et al. [4]

and Smyth [8]. We also see a marked increase in variance as level increases which suggests plateau size becomes less uniform in better regions of the search space.

The random uniform problem instances show similar trends in accuracy. This could be an artifact of the inherent statistical regularity of random problems. To address this, we tested our predictions on structured problem instances. We performed the above experiments on the top three levels of a set of six Ramsey number problems from the MAXSAT 2007 problem competition. This problem set is comprised of several different instances with differing numbers of variables and clauses. The results are shown in Figure 5 on the bottom left. We also performed the above experiments on the top 10 levels of a 27 variable spin glass problem. This problem is unsatisfiable and the best solution by WalkSat was found on level set 145 (out of 162). These results are shown on the bottom right in Figure 5. The sparsity of the bottom plots is due to the smaller cardinality of the structured problem sets. The Ramsey numbers problems tended to have the largest concentration values: their nonzero concentration ranged from 0.09 to 0.68. The other instances had nonzero concentration values ranging from 0.01 to 0.2 or less. The concentrations were also higher relative to the critical value of $\frac{1}{n-1}$ on the structured instances, hence the paucity of data points from the Bethe model on the corresponding plots.

We see similar trends in accuracy with size across the random and structured problems. Furthermore, the trend is again similar to what we found on the hypercube graph model reported in Figure 2.

4 Impact on Algorithm Design

Accepting equal search moves can be beneficial or detrimental to a search algorithm depending on the immediate properties of a plateau region. Small, easy to escape plateaus offer little impediment to search while large, hard to escape plateaus are vast regions that lack "gradient" information and may result in search stagnation. A stochastic local search algorithm exhibits *plateau behavior* when a significant number of consecutive steps all have the same evaluation. This behavior signals that a plateau has been reached by the algorithm and certain measures may need to be taken to either exploit or react to the encountered plateau region.

Plateau moves can be *beneficial* to stochastic local search [1,3] because they provide neutral moves that may eventually lead to improving states. Plateau behavior is thus not always problematic. Frank et al. [4] point out that stochastic local search algorithms typically respond to plateau behavior by one or more of the following strategies 1) doing nothing, 2) detecting plateaus, 3) performing a short random walk, or 4) randomly restarting. The viability of each of these tactics depends on the size of the plateau in question, along with its exit density. Hence, knowledge of the expectation of plateau size can be beneficial in determining how an algorithm should react to plateau behavior.

For example, plateaus that are relatively small might easily be enumerated with breadth-first search whereas moderately sized plateaus (depending on exit density) might be escaped by taking a small "jump," e.g., flipping a number of variables at random [14]. On the other hand, a search that has reached a vast plateau region may obtain better results, depending on the evaluation level, by simply restarting. Even roughly identifying the size of moderate to large plateau regions can be difficult.

On large plateaus, several researchers [4,5] have discussed the inherent trade-off between continuing plateau search and changing the search strategy. In these studies, empirical measurements of plateau size are gathered off-line for a representative sample of a particular problem class and plateau characteristics are generalized to the entire class. The bound estimates presented in this paper allow rough plateau size approximations to be performed without much computational effort on-line. These estimates could be used in stochastic local search algorithms for strategic adaptation during execution: potentially providing information that allows search to quickly determine which of the above options (or other strategies) may be the best way to respond to plateau behavior.

We might also generalize their use to the prediction of *algorithm* performance. According to the well-known *no free lunch theorem* [15], no single algorithm performs consistently well across a set of problem instances. Indeed, specific algorithm performance often depends strongly on salient problem instance features. *Portfolio-based* approaches use different features to select among a set of algorithms to be applied [16]. For instance, Xu et al. [17] have recently introduced the SATzilla portfolio for satisfiability problems. The approach requires learning the relationship between a problem feature set and the likelihood of a particular algorithm's success in solving an instance with a given set of measurements. We conjecture that plateau size can provide additional problem space information. Hence the concentration and percolation estimates for MAXSAT presented in this paper may be beneficial as a computationally cheap, if rough estimate of a problem instance feature that may aid algorithm selection.

5 Conclusion

We have introduced methods for estimating bounds on plateau size for MAXSAT problems. These bounds may support portfolio approaches to MAXSAT by indicating problem difficulty for local search or principled adaptation for handling large plateaus.

We found that the accuracy in our estimates showed surprisingly similar trends across both random and structured problem instances. However, one inherent weakness with the approach is the large divergence in accuracy with plateau size. In the case of Bethe approximation, this is an artifact of the instability as the critical concentration is approached, thus the bound is not useful for larger values of p. For the Hamming path set, the bounds diverge as the cumulative effect from ignoring the internal path term increases.

We are continuing to refine the bounds on hypercube percolation, which would address divergence in accuracy. Furthermore, we would like to assess the influence of the phase transition on concentration.

The second important plateau characteristic is exit density, which we have not addressed in this paper. Future work also includes estimating plateau exit density and relating exit density and plateau size to problem difficulty and stochastic local search behavior.

References

1. Gent, I.P., Walsh, T.: An empirical analysis of search in GSAT. Journal of Artificial Intelligence Research 1, 47–59 (1993)
2. Selman, B., Levesque, H., Mitchell, D.: A new method for solving hard satisfiability problems. In: Proceedings of AAAI 1992, San Jose, CA (1992)
3. Mastrolilli, M., Gambardella, L.M.: How good are tabu search and plateau moves in the worst case? European Journal of Operations Research 166, 63–76 (2005)
4. Frank, J., Cheeseman, P., Stutz, J.: When gravity fails: Local search topology. Journal of Artificial Intelligence Research 7, 249–281 (1997)
5. Hampson, S., Kibler, D.: Plateaus and plateau search in boolean satisfiability problems: When to give up searching and start again. DIMACS Series in Discrete Math and Theoretical Computer Science 26, 437–453 (1993)
6. Yokoo, M.: Why adding more constraints makes a problem easier for hill-climbing algorithms: Analyzing landscapes of CSPs. In: Smolka, G. (ed.) CP 1997. LNCS, vol. 1330, pp. 356–370. Springer, Heidelberg (1997)
7. Hoos, H.H., Stützle, T.: Stochastic Local Search: Foundations and Applications. Morgan Kaufmann, San Francisco (2004)
8. Smyth, K.R.G.: Understanding stochastic local search algorithms: An empirical analysis of the relationship between search space structure and algorithm behaviour. Master's thesis, University of British Columbia (2004)
9. Reidys, C.M., Stadler, P.F.: Neutrality in fitness landscapes. Applied Mathematics and Computation 117, 321–350 (2001)
10. Reidys, C., Stadler, P., Schuster, P.: Generic properties of combinatory maps and neutral networks of RNA secondary structures. Bull. Math. Biol. 59, 339–397 (1997)
11. Stauffer, D., Aharony, A.: Introduction to Percolation Theory. Routledge, New York (1991)
12. Schuster, P., Fontana, W., Stadler, P.F., Hofacker, I.L.: From sequences to shapes and back: a case study in RNA secondary structures. In: Proceedings of the Royal Society London B, vol. 255, pp. 279–284 (1994)
13. Selman, B., Kautz, H., Cohen, B.: Local search strategies for satisfiability testing. In: Johnson, D.S., Trick, M.A. (eds.) DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 26. AMS, Providence (1996)
14. Gent, I., Walsh, T.: Unsatisfied variables in local search. In: Hallam, J. (ed.) Hybrid Problems, Hybrid Solutions, pp. 73–85. IOS Press, Amsterdam (1995)
15. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)
16. Leyton-Brown, K., Nudelman, E., Andrew, G., McFadden, J., Shoham, Y.: A portfolio approach to algorithm selection. In: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI 2003 (2003)
17. Xu, L., Hutter, F., Hoos, H., Leyton-Brown, K.: SATzilla: Portfolio-based algorithm selection for SAT. Journal of Artificial Intelligence Research 32, 565–606 (2008)