Proceedings of the Third Annual Symposium on Combinatorial Search (SOCS-10)

Directed Plateau Search for MAX-k-SAT

Andrew M. Sutton and **Adele E. Howe** and **L. Darrell Whitley**
Department of Computer Science*
Colorado State University
Fort Collins, CO 80523
{sutton,howe,whitley}@cs.colostate.edu

Abstract

Local search algorithms for MAX-k-SAT must often explore large regions of mutually connected equal moves, or plateaus, typically by taking random walks through the region. In this paper, we develop a surrogate plateau "gradient" function using a Walsh transform of the objective function. This function gives the mean value of the objective function over localized volumes of the search space. This information can be used to direct search through plateaus more quickly. The focus of this paper is on demonstrating that formal analysis of search space structure can direct existing algorithms in a more principled manner than random walks. We show that embedding the gradient computation into a hill-climbing local search for MAX-k-SAT improves its convergence profile.

Introduction

Local search algorithms have classically been characterized by iteratively accepting only neighbors with a *strictly* improving objective function evaluation. However, in the case of many combinatorial problems, it can be beneficial to also accept neighbors with *equal* evaluation in the interest of eventually discovering improving states. For example, for maximum k-satisfiability problems (MAX-k-SAT), Selman, Levesque, and Mitchell (1992) first discovered that accepting equal moves empirically improved the convergence time of local search. This was later studied in detail by Gent and Walsh (1993). From a theoretical perspective, Mastrolilli and Gambardella (2005) show that on unweighted MAX-k-SAT allowing local search to take equal moves results in an approximation ratio of $2/3$ which is superior to basic local search's approximation ratio of $1/2$.

The success of local search largely depends on how quickly it can follow a discrete "gradient" to move to better states. When equal moves are allowed, search must contend with *plateaus*: connected regions of the search space that are equivalent under the objective function. Plateaus pose a challenge to local search since they provide no gradient information to guide search.

Plateaus are a prominent search space feature in MAX-k-SAT problems (Smyth 2004). In this paper, we develop a "surrogate gradient" heuristic for MAX-k-SAT (on which clause length is bounded by a constant k, e.g., MAX-3-SAT) to help local search navigate plateaus. We apply a Walsh transform of the MAX-k-SAT objective function to perform a fast computation of the mean value of states over large volumes of arbitrary radius around plateau states. We then use this statistic to direct search across plateaus.

Our goal is to demonstrate that formal analysis of search space structure can direct existing algorithms in a more principled manner than random walks. Our contribution is twofold. First, we exploit theoretical analysis (exact computation of mean objective values) to develop a heuristic that guides local search through plateaus. Second, we empirically assess the utility of the heuristic in a study of its application to MAX-k-SAT problems. We first test the hypothesis that the guidance offered by the surrogate gradient improves the time to escape a particular plateau. We then apply the heuristic to the plateau phase of a hill-climbing local search algorithm in order to determine whether this ultimately translates to faster convergence. We find the surrogate gradient to be advantageous in directing search through plateaus to near-optimal levels.

Background

Formally, a combinatorial optimization problem is defined by a set of discrete states $\mathcal{X}$ and an objective function $f : \mathcal{X} \to \mathbb{R}$. Assuming minimization (as we shall hereafter), a solution to a particular combinatorial optimization problem is a state

$$x^* = \operatorname*{argmin}_{x \in \mathcal{X}} f(x).$$

Local search algorithms impose a neighborhood operator $N : \mathcal{X} \to 2^{\mathcal{X}}$. A plateau is a maximal set $P \subseteq \mathcal{X}$ such that, for all $x, y \in P$, there is a path $(x = x_1, x_2, \ldots, x_p = y)$ where, if $p > 1$, $x_{i+1} \in N(x_i)$ and $f(x_{i+1}) = f(x_i)$ for $i = 1, 2, \ldots, p$. The *level* of P is the objective function evaluation of its elements.

Since this definition allows for degenerate plateaus of cardinality one, every state belongs to a unique plateau.

*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.

Broadly speaking, hill-climbing local search algorithms progressively attempt to escape plateaus of constantly improving level. If no state on the plateau has a neighbor on a lower level plateau, the plateau is *closed*. On the other hand, a plateau that has one or more states with improving neighbors is an *open* plateau and can eventually be escaped. Determining whether a plateau P is open or closed takes time proportional to $|P|$ in the worst case.

A hill-climbing local search algorithm can attempt to escape a plateau region by moving across it using a random walk or by performing systematic search. Either approach can be prohibitive since the number of states in a particular plateau can be exponential in the problem size. The distribution of exit states across a plateau further impacts how quickly it is escaped.

Hampson and Kibler (1993) empirically studied plateau characteristics in MAX-k-SAT focusing on plateaus at near-optimal levels. They found that the number of plateaus in this region grows linearly with n (where n is the number of variables). They also found that the size of plateaus at better values grows exponentially with n, while the density of escapes decreases with n, producing an $O(n)$ increase in waiting time to escape plateaus. Thus the linear growth in waiting time on plateaus along with linear growth in the number of plateaus should produce a growth rate for a single hill-climbing episode that is roughly quadratic.

Frank et al. (1997) found that escape density on open plateaus tends to decrease as plateau level approaches the optimal objective function value. This implies that plateaus nearer to optimal solutions become increasingly difficult to escape. They also found that escape density for near-optimal plateaus *increases* with constrainedness as measured by number of clauses. Smyth (2004) performed an empirical analysis of plateau structure on uniform random and structured problems. He found that closed plateaus tend to be smaller than open plateaus and that structural characteristics of plateaus correlate with instance hardness for stochastic local search algorithms.

A common strategy for escaping plateaus is to introduce a small amount of noise to the search process. For satisfiability problems, noise is added to the local search process in the form of a biased random walk (Selman, Kautz, and Cohen 1994) which gives rise to the high performance WALKSAT algorithm (Selman, Kautz, and Cohen 1996). Algorithms from the WALKSAT family avoid plateaus by inverting variables that belong to unsatisfied clauses even if the move results in a *disimproving* value of the objective function. Within a clause, however, it may be necessary to break ties among a collection of variables if they have equal score. To address this, tie-breaking heuristics are based on the *dynamics* of the algorithm, such as how recently the variables have been flipped, e.g., WALKSAT-TABU, Novelty, and R-Novelty (McAllester, Selman, and Kautz 1997). Adding stochasticity to the latter two results in the probabilistically approximately complete variants: Novelty+ and R-Novelty+ (Hoos 1999). Further refinements include diversification (Novelty++), deterministic greedy moves (G^2WSAT) (Li and Huang 2005), adaptive noise and combinations of these strategies (Li, Wei, and Zhang 2007). In contrast to these tie-breaking heuristics, our surrogate gradient is based solely on features of the search space.

Computing a surrogate gradient

The state space for an n variable, m clause MAX-k-SAT instance is the set of all complete variable assignments which is isomorphic to $\{0,1\}^n$: the set of all binary strings of length n. Let

$$f_j(x) = \begin{cases} 1 & \text{if clause } j \text{ is unsatisfied under } x; \\ 0 & \text{otherwise.} \end{cases}$$

The MAX-k-SAT objective function can be written as

$$f(x) = \sum_{j=1}^{m} f_j(x).$$

The problem of maximizing the number of satisfied clauses thus reduces to minimizing f.

Local search applied to MAX-k-SAT typically employs the "flip" operator; the neighborhood consists of states that can be reached by inverting exactly one variable in the assignment. Thus the search space can be characterized as a metric space on $\{0,1\}^n$ using Hamming distance.

In the following, we show how to efficiently calculate the mean value of the objective function over volumes of the search space and use it to distinguish among states of equal value. To do so, we first introduce a Walsh transform of the MAX-k-SAT objective function.

Walsh transform

For $0 \le i < 2^n$, the i^{th} Walsh function is defined as

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

where the inner product is taken over the length-n binary string x and the length-n binary string representation of the index i. The *order* of the i^{th} Walsh function is $\langle i,i \rangle$, that is, the number of ones in the length-n binary string representation of i.

Any function $h : \{0,1\}^n \to \mathbb{R}$ can be represented in the orthogonal Walsh basis:

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

where w_i is a real-valued constant called a Walsh coefficient.

Functions of this type can be seen as elements of a vector space $\mathbb{R}^{2^n}$. The MAX-k-SAT objective function f described above is one such example. The Walsh transform can be applied to f in the following way.

$$w = \mathbf{W} f, \qquad f = \mathbf{W}^{-1} w,$$

where $\mathbf{W}$ is a $2^n \times 2^n$ matrix such that

$$\mathbf{W}_{ij} = \frac{1}{2^n} \psi_i(j).$$

Since $\mathbf{W}$ is a linear transform, the coefficient vector can be computed as

$$w = \mathbf{W} f = \sum_{j=1}^{m} \mathbf{W} f_j.$$

Rana et al. (1998) have shown that if k_j is the number of literals in the j^{th} clause then the order of nonzero Walsh coefficients of the corresponding subfunction f_j is bounded by k_j. This means that each subfunction f_j contributes at most 2^{k_j} nonzero Walsh coefficients to the Walsh transform of f and w is sparse. Hence the objective function for MAX-k-SAT with at most k literals per clause can be written as

$$f(x) = \sum_i w_i \psi_i(x), \tag{1}$$

and the number of nonzero terms is at most $m2^k$. Since k is typically taken to be constant, the Walsh transform can be computed in $O(m)$ time.

The mean of f over search space volumes

Let $\mathcal{D}(x, y)$ denote the Hamming distance between states x and y. The sphere of radius r around a state x is the set of all states lying at Hamming distance r from x defined as

$$S^{(r)}(x) = \{y : \mathcal{D}(x, y) = r\}.$$

The $2^n \times 2^n$ sphere matrix of radius r is defined as

$$\mathbf{S}^{(r)}_{xy} = \begin{cases} 1 & \text{if } \mathcal{D}(x,y) = r; \\ 0 & \text{otherwise.} \end{cases}$$

We use the fact that Walsh functions are eigenfunctions of sphere matrices to compute the mean of f over spheres of arbitrary radius in polynomial time. In general, we can compute

$$\mathbf{S}^{(r)} \psi_i(x) = \gamma_i^{(r)} \psi_i(x) \tag{2}$$

where $\gamma_i^{(r)}$ is a recursively-defined eigenvalue of $S^{(r)}$:

$$\gamma_i^{(r)} = \frac{1}{r}\left((n - 2\langle i, i\rangle)\gamma_i^{(r-1)} - (n - r + 2)\gamma_i^{(r-2)}\right) \tag{3}$$

with $\gamma_i^{(1)} = (n - 2\langle i, i\rangle)$ and $\gamma_i^{(0)} = 1$.

Equations (2) and (3) are proved in Sutton et al. (2010) for the general case of k-bounded pseudo-Boolean functions (which includes MAX-k-SAT).

Let the function $g^{(r)}(x)$ denote the mean value of f evaluated over all states contained in a volume of Hamming space within radius r of x. The total number of states lying within a Hamming ball of radius r is

$$t_r = \sum_{u=0}^{r} \binom{n}{u}$$

and the function $g^{(r)}$ can be computed as follows

$$\begin{aligned} g^{(r)}(x) &= \frac{1}{t_r} \sum_{u=0}^{r} \sum_{y \in S^{(u)}(x)} f(y) \\ &= \frac{1}{t_r} \sum_{u=0}^{r} \mathbf{S}^{(u)} f(x) \\ &= \frac{1}{t_r} \sum_{u=0}^{r} \sum_i \gamma_i^{(u)} w_i \psi_i(x) \qquad \text{by (1) and (2).} \end{aligned}$$

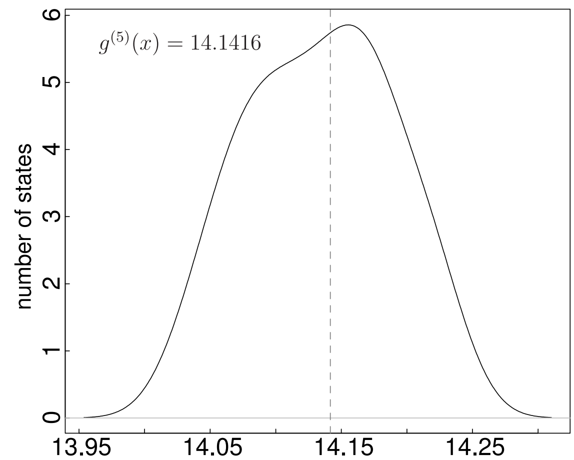

Figure 1: Empirical density function of $g^{(5)}$ evaluated over equal neighbors of a plateau state x sampled from SATLIB instance `uf250-1065-01`.

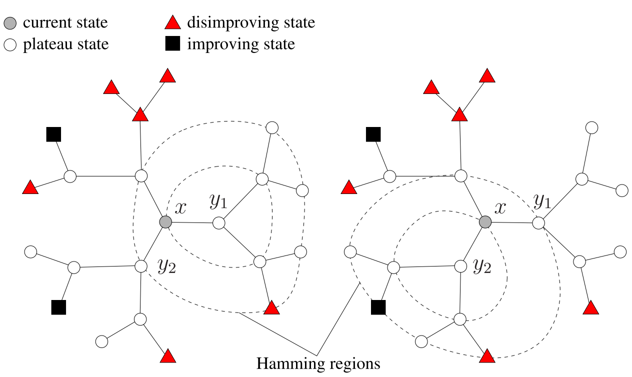

Figure 2: Schematic of surrogate gradient heuristic. Hamming ball of radius two (denoted by closed splines) around neighbors y_1 and y_2 of state x. Due to an improving state near y_2, it is likely that $g^{(2)}(y_2) < g^{(2)}(y_1)$.

which is again computable in time polynomial in the number of variables of the instance. Note that this bound holds even if the cardinality of the region of space is exponential in n.

An important observation is that $f(x) = f(y)$ *does not necessarily imply* $g^{(r)}(x) = g^{(r)}(y)$ and so $g^{(r)}$ might be used to delineate among a set of states with equal evaluation. For example, in Figure 1 we plot the empirical density function of $g^{(5)}$ evaluated over equal neighbors of a particular state sampled from SATLIB instance `uf250-1065-01`.

All other things being equal, a plateau state with escapes within radius r would be expected to have a lower average $g^{(r)}$ value than a state with no escapes within radius r. Thus $g^{(r)}$ could function as a heuristic for choosing more promising states among a set with equal evaluation. In other words, $g^{(r)}$ might be used as a surrogate "gradient" function to guide search across plateaus (see Figure 2).

DPS$(x, f, g^{(r)})$

$g_{best} \leftarrow g^{(r)}(x)$
while not done
  do $N \leftarrow \{y : \mathcal{D}(x, y) = 1\}$
    if $\exists\, y \in N : f(y) < f(x)$
      then return
    $E \leftarrow \{y \in N : f(y) = f(x)\}$
    $x \leftarrow \text{argmin}_{z \in E}\, g^{(r)}(z)$
    if $g^{(r)}(x) \geq g_{best}$
      then $x \leftarrow$ random element of E

Figure 3: Directed plateau search process.

Escaping plateaus

In this section we test the hypothesis that a surrogate gradient function that computes the mean value of the objective in a volume of search space of a given radius can direct a search algorithm to escape a plateau more quickly than random search alone.

Directed plateau search

On non-degenerate plateaus, the objective function is no longer useful for directing search to more promising states. Typically, hill-climbing search algorithms resort to a stochastic process by iteratively selecting equal neighbors at random until the plateau is escaped. We will instead use $g^{(r)}$ to direct search across plateau states by choosing plateau neighbors that lie in regions with lower average f. Given a state x, we perform local search minimizing $g^{(r)}$ using only neighbors with equivalent f values. The directed plateau search process (DPS) is given in Figure 3.

Until a plateau exit is found, plateau moves are chosen to minimize the surrogate gradient $g^{(r)}$. However, it is possible to reach a local optimum with respect to $g^{(r)}$. In this case, the search reverts to a random plateau walk by taking random plateau moves without regard to $g^{(r)}$ until a plateau move with an improving $g^{(r)}$ value is found.

Empirical results

Our hypothesis is that the information provided by $g^{(r)}$ allows DPS to escape plateaus more quickly on average than a random plateau walk (RPW). Starting from an initial state x, each plateau escape process generates a sequence of, not necessarily unique, states $(x = x_1, x_2, \ldots, x_t)$, called a *trace*, until a plateau exit is found, or the number of states exceeds some bound. We define $L_{dps}(c)$ to be the *trace length* of DPS on level c: the length of a trace beginning at a state x with $f(x) = c$ until stopping criteria are met and define $L_{rpw}(c)$ to be the length of a trace generated by a random plateau walk with initial state on level c.

If we choose states uniformly at random from a particular level c, we can characterize $L_{dps}(c)$ and $L_{rpw}(x)$ as random variables. So to test our hypothesis we must show that $L_{dps}(x)$ stochastically dominates $L_{rpw}(c)$. Sampling these random variables amounts to performing both a DPS and a random plateau walk from states on a level and measuring their trace lengths. To do so, we sampled 100 states each at levels 5,4,3,2, and 1 on all 1000 instances in the SATLIB benchmark set `uf100-430`. For each sampled state, we measured the trace length of both DPS and a random plateau walk. For the radius parameter we used a number of different values: $r = \{1, 2, 5, 10, 20\}$.

r	1	2	5	10	20
mean	30.38	29.75	28.88	29.55	29.87
sd	109.33	111.71	104.36	110.16	111.71

Table 1: Mean and standard deviation of DPS trace lengths for different radius values on levels 4 and 5 of `uf100-430`

Note that if the initial state has an improving neighbor, the plateau can be immediately escaped by both processes (trace of length 1) so such data points are useless to our experiment and are removed from the data. Furthermore, the maximum allowed trace length was set to 2000 states. We say the process *fails* if it does not escape the plateau within the allotted trace length. To remove states that may lie on closed plateaus, we remove from consideration states on which both the random walk and DPS process fail.

To test whether r has an effect on escape time, we measured the mean and standard deviation of the trace lengths on levels 4 and 5 of the `uf100-430` distribution. The results are shown in Table 1. The mean trace length for DPS appears not to significantly depend directly on radius. For the experiments in this paper, we will use $r = 5$.

The results for $r = 5$ on the `uf100-430` distribution are shown in Figure 4. The data come from a population of 10^5 states (100 states/instance). The distribution of each random variable is heavy-tailed, and follows an overdispersed Poisson distribution. Such a distribution can be modeled by a negative binomial distribution with parameters $\alpha = \beta = \sigma^{-2}$ where σ^2 is the variance. To test for stochastic dominance we perform the (nonparametric) sign test that $L_{rpw}(c) - L_{dps}(c)$ is, on average, greater than zero. For each level and each radius, we compute a p-value of less than 0.0001 when comparing to random plateau walk. We can thus conclude there is a statistically significant effect, and that DPS escapes plateaus more quickly on average than a simple random plateau walk. Statistics for the escape experiments are shown in Table 2.

Timing and efficiency. Clearly, the Walsh coefficients and the sphere eigenvalues can be computed off-line. If the value of $g^{(r)}(x)$ is stored when DPS is called, for any $z \in N(x)$ the value of $g^{(r)}(z)$ can be obtained using a difference equation. This equation can be evaluated in time proportional to the number of literals containing the variable which must be negated to transform x into z. Hence, there is a small overhead associated with calculating the surrogate gradient in each plateau search step. We would expect, however, that shorter trace lengths ultimately translate to faster escape time. To investigate this further, let $T_{dps}(c)$ denote the processing time needed by DPS to escape level c and $T_{rpw}(c)$ be the processing time needed by a random plateau

	level 5		level 4		level 3		level 2		level 1	
	random	dps	random	dps	random	dps	random	dps	random	dps
median trace	8	5	11	6	20	11	57	43	248	230
mean trace	23.19	17.89	44.23	34.78	115.58	103.21	283.34	271.97	574.92	561.42
std dev	76.71	76.21	131.89	116.27	291.16	280.45	504.00	503.98	679.23	678.45
% failures	0.03	0.05	0.14	0.07	0.96	0.87	4.16	4.15	10.94	10.69
mean % directed	-	61.02	-	54.72	-	44.09	-	29.93	-	14.18
% both failed	0.003		0.072		0.974		9.991		60.572	

Table 2: Results for plateau escape experiments: trace length statistics, percentage of runs each method failed (i.e., reached the cutoff), and (for DPS) mean percentage of steps that utilized the surrogate gradient heuristic. We remove runs in which both methods failed.

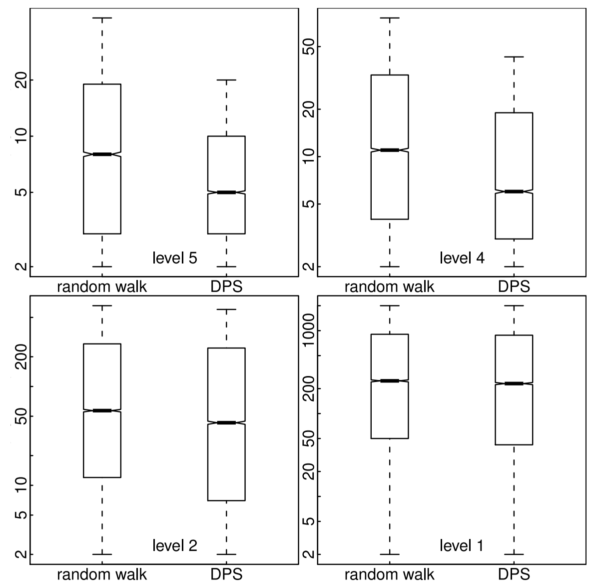

Figure 4: Plateau escape experiments (at radius 5) for levels 5,4,2, and 1 of uf100−430 distribution. A sign test confirms statistical significance for each with $p < 0.0001$.

walk. We measure the *relative speed-up* at level c as $\frac{T_{rpw}(c)}{T_{dps}(c)}$. We report the median relative speed-up for levels 10 to 1 on the uf100−430 distribution in Figure 5. On higher levels, when there is an improvement in trace length, we see this corresponds to an improvement in processing time. However, as the lowest levels are approached, we see that the advantage of DPS is diminished toward lower levels (c.f. Figure 4), and the overhead for computing the surrogate gradient translates to a slow down in processing time.

Hill-climbing search

An episode of hill-climbing local search can be seen as a process that escapes plateaus at progressively improving levels. Suppose a hill-climbing process is started from an arbitrary state x_0 and eventually reaches some state x^* with $f(x^*) < f(x_0)$ (recall we are minimizing). The waiting time (in terms of number of evaluations) between these two boundary states can be modeled as a sum over random variables which gives the time spent at each level between $f(x_0)$ and $f(x^*)$. Letting τ_c denote the number of evaluations performed at level c, we have the total waiting time

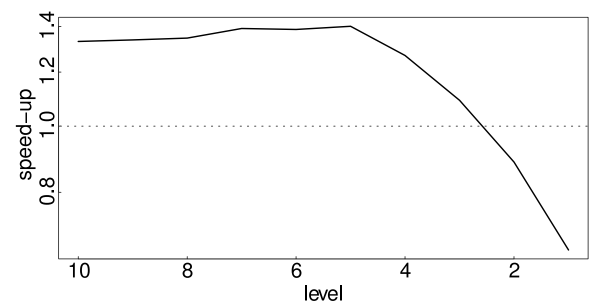

Figure 5: Median relative CPU time speed-up (DPS) for escaping best 10 levels of uf100−430 distribution.

$$\Lambda = \sum_{c=f(x^*)+1}^{f(x_0)} \tau_c \tag{4}$$

Since not all levels are visited during an episode of hill-climbing, let χ_c be the indicator random variable where

$$\chi_c = \begin{cases} 1 & \text{if search skips level } c; \\ 0 & \text{otherwise.} \end{cases}$$

If the hill-climbing local search randomly selects among equal neighbors, then we have

$$\tau_c = L_{rpw}(c)(1 - \chi_c)$$

By linearity of expectation the expected waiting time is

$$\mathbb{E}[\Lambda] = \sum_{c=f(x^*)+1}^{f(x_0)} \mathbb{E}[L_{rpw}(c)(1 - \chi_c)] \tag{5}$$

Instead of a random walk, if DPS is implemented at each level to direct search across plateaus, we can substitute $L_{rpw}(c)$ with $L_{dps}(c)$ in Equation (5). If we assume that the probability of skipping a level is invariant under plateau search dynamics (and we have no reason to believe otherwise), then given the results presented in the previous section, we would expect statistically shorter convergence times

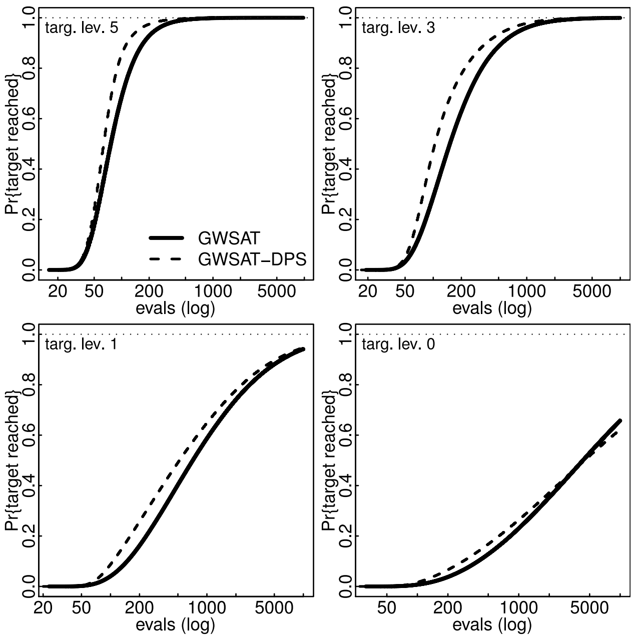

Figure 6: Empirical run length distributions for 1000 runs each on 1000 instance `uf100-430` distribution for four different target levels.

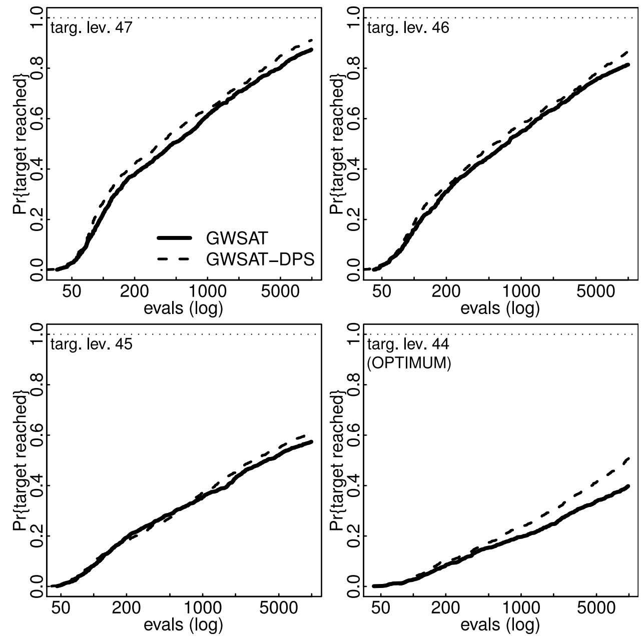

Figure 7: Empirical run length distributions for 1000 runs each on overconstrained instance `s3v80c1000-1` for four different target levels (44 is optimal).

for these hill-climbing episodes when compared to the standard random walk. In other words, using DPS to escape plateaus should result in faster plateau escape times which ultimately translate into faster convergence.

Empirical results

To test the hypothesis that directed plateau search can speed up periods of hill-climbing local search, we incorporated the DPS process into the plateau phase of GWSAT: a variant of the randomized greedy hill-climbing local search algorithm in which an unbiased random walk is performed with probability p in each step (Selman and Kautz 1993). Our reasoning for using GWSAT is that the random walk element is necessary for probabilistically departing closed plateaus, which still pose a problem to DPS.

Since we assess the effect of DPS on single episodes of hill-climbing, we set the MAX-TRIES parameter to 1. We set the MAX-FLIPS parameter of GWSAT to 10000 and used a walk probability of $p = 0.3$. To observe the effect of DPS on GWSAT's convergence to specific levels of the objective function, we performed both GWSAT and GWSAT with DPS (GWSAT-DPS) targeting different levels.

Run length distributions for levels 5, 3, 1, and 0 (level 0 being the optimal value) are plotted in Figure 6. To generate these run length distributions, we performed both searches 1000 times for each instance in `uf100-430`. This set contains 1000 instances so each empirical cumulative distribution function is generated from 10^6 data points. For these target levels, GWSAT-DPS dominates. In target level 0, there is a slight crossover around 2000 evaluations.

The SATLIB benchmark set we used contains filtered random uniform problems generated at the phase transition region (clause to variable ratio ≈ 4.3). To investigate the impact of adding DPS to a hill-climbing algorithm deep into the overconstrained phase, we generated empirical run length distributions for benchmark instances from the Fourth MAX-SAT Evaluation in 2009.[1] Again, we performed both searches 1000 times on each instance in the set `s3v80c1000` (clause to variable ratio 12.5). We use the same settings as above except for the radius argument for DPS which we set to 10. Since each instance has a different optimal value, we plot the empirical run length distribution only for 1000 runs on a single instance (`s3v80c1000-1`) in Figure 7. For this instance, the optimal value (found by a complete solver) is 44. Again we see dominance for the variant with DPS, but no crossover at the optimal level.

Implications for search

The empirical results suggest that GWSAT augmented with DPS tends to dominate until the lowest level plateaus. A closer look at the number of evaluations necessary for convergence to levels 5 and 0 is given in Figure 8. The effect is far less dramatic when converging to level 0.

The observed traces for DPS at each level suggest that on near-optimal plateaus, the surrogate gradient is not followed as often. For instance, in Table 2, at level 1 an average of approximately 14% of the steps are directed whereas at level 5 just over 62% of the steps are directed. This is likely due to a loss in resolution of the surrogate gradient at lower levels. As the extremal value of f is approached, the

[1] `http://www.maxsat.udl.cat/09/`

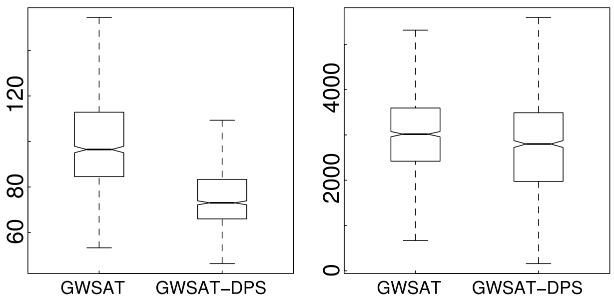

Figure 8: Comparison of number of evaluations (log scale) to find level 5 (left) and level 0 (right) on `uf100-430`.

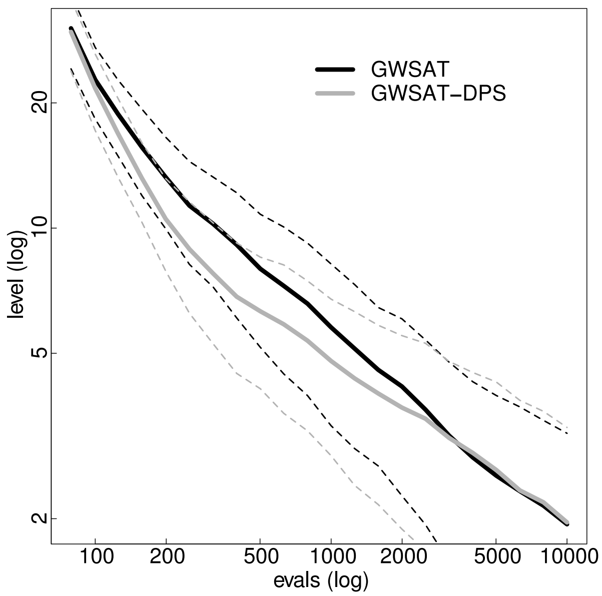

Figure 9: Mean convergence plot for GWSAT and GWSAT-DPS over `uf250-1065` distribution. Dashed lines indicate standard deviation from mean convergence.

local volume will be composed mainly of states on disimproving levels. Furthermore, many plateaus at this level are closed; GWSAT-DPS will have to rely on GWSAT's walk probability rather than plateau search. These characteristics affect a heavy tailed distribution in waiting time on plateaus at near optimal levels. These factors statistically obscure the gains in convergence obtained by DPS on other levels.

In order to observe the convergence profile more clearly, we plot the mean convergence for GWSAT and GWSAT-DPS over the distribution `uf250-1065` in Figure 9. The gains begin occurring around level 20 and vanish near level 3 as long random walks on low level plateaus begin to dominate.

To investigate this phenomenon on the overconstrained set, we report the percentage of runs (out of 1000) that reached certain levels near and at the optimal level. The optimal level is found by the complete solver MINIMAXSAT (Heras, Larrosa, and Oliveras 2008). Let opt+ℓ be the percentage of runs that reached the ℓ^{th} level above the optimal

inst.	alg.	opt+5	opt+4	opt+3	opt+2	opt+1	opt+0
1	GWSAT	96.9	93.2	88.2	83.8	59.0	43.1
	GWSAT-DPS	**97.6**	**94.4**	**90.3**	**87.5**	**61.3**	**51.6**
2	GWSAT	98.3	95.8	**86.3**	71.3	**47.0**	**35.2**
	GWSAT-DPS	**99.2**	**96.8**	85.7	**72.8**	46.8	32.3
3	GWSAT	88.9	70.1	61.3	52.1	**29.5**	**29.4**
	GWSAT-DPS	**90.5**	**70.4**	**64.7**	**53.4**	29.2	28.8
4	GWSAT	92.4	89.1	82.0	78.1	**42.0**	**18.9**
	GWSAT-DPS	**95.0**	**91.4**	**85.2**	**80.8**	41.3	17.1
5	GWSAT	95.4	93.1	86.5	75.5	**42.5**	28.9
	GWSAT-DPS	**97.0**	**96.1**	**89.5**	**77.0**	40.4	**30.7**
6	GWSAT	88.7	73.0	72.7	67.6	58.4	43.6
	GWSAT-DPS	**92.5**	**73.8**	**73.3**	**70.2**	**60.5**	**49.5**
7	GWSAT	98.7	98.3	95.4	89.7	74.6	**73.8**
	GWSAT-DPS	**99.7**	**99.5**	**96.3**	**93.0**	**74.7**	64.0
8	GWSAT	96.3	89.4	68.5	63.8	39.3	**18.7**
	GWSAT-DPS	**98.2**	**92.8**	**75.1**	**71.1**	**47.1**	18.2
9	GWSAT	97.1	95.6	95.0	90.0	86.4	40.0
	GWSAT-DPS	**97.8**	**97.4**	**97.1**	**94.0**	**93.4**	**53.5**
10	GWSAT	92.8	88.8	81.1	80.0	62.1	25.7
	GWSAT-DPS	**95.8**	**94.3**	**88.9**	**87.4**	**69.5**	**32.7**

Table 3: Percentage of runs that reached best six levels each of the 10 instances from the MAX-SAT 2009 `s3v80c1000` benchmark set. opt+ℓ is percentage of runs (out of 1000) that reached level ℓ. Higher percentages are in boldface.

(opt+0 is the percentage of runs solved). We report these data in Table 3. Again we see that the heavy-tail escape behavior obscures the advantage of DPS at the optimal level.

The results demonstrate that search space structure can be used to influence the trajectory of plateau search in such a way that certain plateaus may be escaped more quickly. DPS may also be used to quickly move to better regions of the objective function. This could be useful for faster approximations on large problems, or hybridized in the manner of Kroc et al. (2009) to switch to a complete solver when the surrogate gradient is no longer helpful.

Directing other search algorithms

In this paper our aim was to show that knowledge of search space properties can direct a hill-climbing local search such as GWSAT to escape plateaus more quickly. However, GWSAT performs inferior to other local search paradigms such as WALKSAT (Selman, Kautz, and Cohen 1996) in the phase transition region and G^2WSAT (Li and Huang 2005) in the overconstrained phase. Incorporating the DPS method to the plateau phase of GWSAT does not offer enough improvement (even early on) to perform competitively with these other algorithms.

We used GWSAT to isolate as much as possible the improvement offered by the DPS heuristic to a simple hill-climbing algorithm. One direction of future work is to use the surrogate gradient to inform more advanced local search algorithms on the relative benefit of different plateau moves. Though algorithms from the WALKSAT family are not strictly classified as hill-climbing algorithms (disimproving moves are allowed in non-extremal states (Li and Huang 2005)), the surrogate gradient could be used as a heuristic for breaking ties among moves of equal score.

To investigate this we modified the WALKSAT algorithm

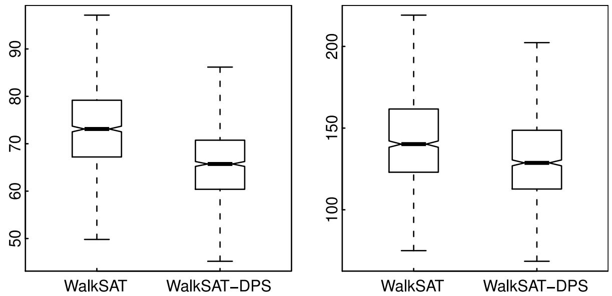

Figure 10: Comparison of number of evaluations (log scale) to find level 10 (left) and level 5 (right) on `uf100-430` using WALKSAT and WALKSAT-DPS.

by integrating the search step in DPS (Figure 3) to break ties among moves of equal score. We measured the number of evaluations each variant required to reach certain levels over the critically constrained `uf100-430` distribution. The results are given in Figure 10 for levels 10 and 5. A one-sided t-test shows the integration of the DPS step results in a significant reduction in convergence time to both levels at $p < 10^{-15}$. As with GWSAT-DPS, the effect of directed plateau search becomes negligible at the optimal level. Again, we conjecture that here the loss in resolution diminishes the power of the surrogate gradient.

In line 9 of the DPS process in Figure 3 performs a blind random selection from the equal neighbors if no surrogate gradient information is available. Another direction of future work is to resort to a tie-breaking heuristic (such as Novelty+ and its variants) in this random selection process to perhaps further improve the performance of the augmented hill-climbing local search.

Conclusion

One of the issues with local search applied to combinatorial optimization problems is that often we do not understand why and when it works. A large amount of effort is expended in ad-hoc tweaking and tuning without enough understanding of the processes that make search successful. In this paper we showed that a structural analysis of the search space can be used to guide a local search algorithm.

We developed a surrogate plateau "gradient" function based on a Walsh transform of the MAX-k-SAT objective function. This surrogate gradient gives the average objective function value over localized volumes of the search space to provide information to direct search through plateaus more quickly. We have shown that this improves the convergence time of hill-climbing local search on MAX-k-SAT problems, especially when targeting near-optimal levels.

We have thus shown that it is beneficial to use exact information about the search space structure to escape plateaus, rather than by resorting to blind random walks. We also believe that this approach will provide a rigorous foundation for future algorithmic innovations.

References

Frank, J.; Cheeseman, P.; and Stutz, J. 1997. When gravity fails: Local search topology. *JAIR* 7:249–281.

Gent, I. P., and Walsh, T. 1993. An empirical analysis of search in GSAT. *JAIR* 1:47–59.

Hampson, S., and Kibler, D. 1993. 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.

Heras, F.; Larrosa, J.; and Oliveras, A. 2008. MiniMaxSat: an efficient weighted Max-SAT solver. *JAIR* 31:1–32.

Hoos, H. 1999. On the run-time behaviour of stochastic local search algorithms for SAT. In *Proc. of AAAI'99*, 661–666.

Kroc, L.; Sabharwal, A.; Gomes, C.; and Selman, B. 2009. Integrating systematic and local search paradigms: A new strategy for maxSAT. In *Proc. of IJCAI'09*.

Li, C. M., and Huang, W. Q. 2005. Diversification and determinism in local search for satisfiability. In *Proc. SAT-05*, volume 3569 of *LNCS*, 158–172.

Li, C.; Wei, W.; and Zhang, H. 2007. Combining adaptive noise and look-ahead in local search for SAT. *LNCS* 4501:121–133.

Mastrolilli, M., and Gambardella, L. M. 2005. How good are tabu search and plateau moves in the worst case? *Eur J Oper Res* 166:63–76.

McAllester, D.; Selman, B.; and Kautz, H. 1997. Evidence for invariants in local search. In *Proc. of AAAI'97*.

Rana, S.; Heckendorn, R. B.; and Whitley, L. D. 1998. A tractable Walsh analysis of SAT and its implications for genetic algorithms. In *Proc. of AAAI'98*, 392–397.

Selman, B., and Kautz, H. A. 1993. Domain-independent extensions to GSAT : Solving large structured variables. In *Proc. of IJCAI'93*, 290–295.

Selman, B.; Kautz, H. A.; and Cohen, B. 1994. Noise strategies for improving local search. In *Proc. of AAAI'94*, 337–343.

Selman, B.; Kautz, H.; and Cohen, B. 1996. Local search strategies for satisfiability testing. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 26.

Selman, B.; Levesque, H.; and Mitchell, D. 1992. A new method for solving hard satisfiability problems. In *Proc. of AAAI'92*.

Smyth, K. R. G. 2004. 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.

Sutton, A. M.; Whitley, L. D.; and Howe, A. E. 2010. Computing the moments of k-bounded pseudo-Boolean functions over Hamming spheres of arbitrary radius in polynomial time. Technical Report CS-10-104, Colorado State University. `http://www.cs.colostate.edu/TechReports/Reports/2010/tr10-104.pdf`