Maximum Satisfiability: Anatomy of the Fitness Landscape for a Hard Combinatorial Optimisation Problem

Adam Prügel-Bennett and Mohammad-H. Tayarani-N.
School of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK.

***Abstract*—The fitness landscape of MAX-3-SAT is investigated for random instances above the satisfiability phase transition. This study includes a scaling analysis of the time to reach a local optimum, the number of local optima, the expected probability of reaching a local optimum as a function of its fitness, the expected fitness found by local search and the best fitness, the probability of reaching a global optimum, the size and relative positions of the global optima, the mean distance between the local and global optima, the expected fitness as a function of the Hamming distance from an optimum and their basins of attraction. These analyses show why the problem becomes hard for local search algorithms as the system size increases. The paper also shows how a recently proposed algorithm can exploit long-range correlations in the fitness landscape to improve on the state-of-the-art heuristic algorithms.**

***Index Terms*—MAXSAT, fitness landscape, scaling analysis, long-range correlation.**

I. Introduction

DESIGNING successful heuristic algorithms for hard combinatorial problems requires an understanding of the fitness landscape structure. This will be problem and even instance dependent. Nevertheless, some statistical properties are common across many instances and even across different problem classes. In this paper, we empirically study the fitness landscape of random instances of MAX-3-SAT. There are a large number of features that are influential in determining how heuristic search techniques perform. This paper studies these properties by examining different random instances with up to a few hundred variables. For this size of problem it is possible to find all the high-fitness local optima by performing multiple hill-climbs. Through studying the scaling behaviour for different problem sizes it is possible to extrapolate many properties to large instances. Thus, although it is easy to find all the global optima for small instance sizes, it also becomes clear why large instances are very challenging.

One notable property we examine is the expected fitness at a fixed Hamming distance from a local optimum. This reveals the existence of long-range correlations in expectation. In a recent paper, a new class of search algorithms, *landscape-guided hopping* was introduced which exploits this long-range correlation and out-performs the state-of-the-art algorithms on large random instances of MAX-3-SAT [1]. This current paper is a follow up to that article providing considerably more details of the landscape properties and a fresh analysis of the algorithm.

The landscape properties of random MAX-SAT has received considerable attention from a number of different research communities [2], [3], [4], [5], [6], [7], [8], [9]. Much of this research has focused on the satisfiability phase transition where, as the number of clauses per variables increases, there is a step change in behaviour (at least, in the limit of large instance size) from almost all random instances being satisfiable to almost all instances being unsatisfiable. It is believed that around this phase transition many properties averaged over the entire ensemble of random problems are analytically tractable, however, away form the phase transition their properties are no longer solvable because of complex clustering of local optima [5], [6], [9], [10]. The phase transition has attracted considerable attention because, empirically, it is found that the time taken for exact methods to prove whether an instance is satisfiable or not appears to grow exponentially in the size of the system around the phase transition. Thus, this region corresponds to instances that are deemed to be “hard” for the satisfiability decision problem. However, the focus of this paper is on understanding MAX-SAT as an exemplar of a hard optimisation problem for heuristic search algorithms. For this, the vicinity of the phase transition is peculiar because it is possible to develop specialised algorithms which take account of the fact that a satisfiable solution cannot have any non-satisfying clauses. Instead, we focus on instances with a higher clause-to-variable ratio which are typically unsatisfiable. We consider this regime more representative of the type of optimisation problems faced by developers of heuristic optimisation algorithms. Although, empirically determining that these instances cannot be satisfied is relatively fast, finding a configuration that satisfies as many clauses as possible remains difficult—the nature of this difficulty is explored in this paper.

Analysing the fitness landscape of optimisation problems has been a vigorous area of research over the last twenty years. Often the goal has been to try to identify features of the landscape which are indicative of the problem difficulty for search algorithms. Examples of measures that have attempted to capture the local ruggedness of a problem include the auto-correlation [11], and fitness distance correlations [12], [13]. However, it was soon realised that it was easy to build problems where these measures were not correlated with problem difficulty (see, for example, [14]). Another active programme of research has been to look at algebraic properties of landscapes and particularly elementary landscapes [15], [16], which are shared by many well known NP-hard problems. Unfortunately, these properties do not correlate with problem difficulty either. MAX-k-SAT does not have an elementary landscape, although it can be viewed as a superposition of k-elementary landscapes [17]. One of the themes of this paper is that to gain a deep insight into a problem requires looking at a wide range of landscape properties. The measures discussed above concentrate on local properties of the landscape. This paper investigates long-range correlations, which we show can be exploited to find better quality solutions. These long-range properties are akin to the "big valley" structure of problems, first observed in TSP [18]. However, the correlations in the MAX-SAT landscape are more complicated than a simple big valley. Although, we consider many properties, we have been selective in the data we present. For example, almost all the data shown is at a ratio of clauses-to-variables of 8. There is nothing special about this choice—we have found the same qualitative behaviour at many different ratios of clauses-to-variable substantially above the phase transition. In our opinion, a more comprehensive set of data would tend to obfuscate rather than elucidate, by overwhelming the reader (not to mention the writer).

This paper makes an "honest attempt" to extrapolate from small to large instances. Of course, such extrapolations rely on an unproven assumption that "nothing funny happens" as we extrapolate to large systems. We have not seen any evidence to suggest that this is the case. Nevertheless, all statements based on extrapolation from small systems come with a caveat that the scaling behaviour we observe in small systems is valid for very large systems. Equally, we make statements about statistical samples assuming that our samples are representative. Care has been taken to collect enough data so that the probability of a significant error is small. There are times when we show the behaviour of a single instance, since averaging over many instances would lose information. In such cases, we have not attempted to select an instance we believe is typical, but rather we have drawn an instance at random. Since large random instances often have similar statistical properties, these randomly generated instances are often "typical" (i.e. representative of a large proportion of randomly drawn instances). However, as we will see there are properties, such as the number of configurations in a global optimum, which can vary tremendously between instances.

Although, the detailed behaviour discussed in this paper is peculiar to randomly drawn instances of MAX-3-SAT, many of the qualitative features, such as the proliferation of local optima and the large-scale correlations, are observed in many other combinatorial optimisation problems. Thus, we believe that much of the qualitative behaviour discussed is likely to be shared by many other hard optimisation problems.

The rest of the paper is organised as follows. In the next section, we introduce the MAX-SAT problem and describe one of the most commonly used heuristic solvers, GSAT. Section III describes properties of global and local optima. In section IV, we examine the expected fitness of configuration in Hamming spheres of different radii from a local maximum. We also consider the probability of returning to a local optimum starting from a randomly chosen configuration in the Hamming sphere. In section V, we discuss other MAX-SAT solvers and in particular a new class of algorithms, Landscape-Guided Hopping, developed from an understanding of the long-range correlations. We draw conclusions in section VI. A few details of the analysis technique are left to the appendices.

II. MAX-SAT

In this section, we describe the MAX-SAT problem and then specify the set of instances that we shall consider. We finish the section with a discussion of GSAT, the classic local-search algorithm for MAX-SAT.

A. Problem Definition

The MAX-SAT problem is closely related to the satisfiability decision problem colloquially known as SAT. This problem involves a set of Boolean variables $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ and a set of disjunctive clauses consisting of a subset of literals (a literal is either a variable or its negation). For example, a clause might be $X_1 \vee \neg X_5 \vee X_{10}$. Each clause can be considered an additional constraint that must be satisfied. In SAT the question is: "does there exist an assignment of the variables which satisfies all the clauses?". Stephen Cook famously showed that any non-deterministic Turing machine can be reduced to a SAT instance whose size is a polynomial of the tape length, thus establishing that SAT is NP-complete [19]. A special variant of SAT is k-SAT which consists of clauses containing exactly k literals. There is a staightforward polynomial reduction of any SAT instance to a 3-SAT instance. Thus, 3-SAT is NP-complete while 2-SAT can be solved in polynomial time.

MAX-SAT is the generalisation of SAT to problems which are not fully satisfiable. It asks the question whether there exists an assignment of the variables which satisfies all but T clauses. MAX-k-SAT is NP-hard for $k \geq 2$ (thus MAX-2-SAT is NP-hard even though 2-SAT is not [20]). We will treat MAX-SAT as an optimisation problem and in particular we consider MAX-3-SAT. We use as the objective function the number of satisfied clauses, which we seek to maximise. Assuming there are m clauses and denoting the clauses by $g_i(\boldsymbol{X})$, then the fitness is given by

$$f(\boldsymbol{X}) = \sum_{i=1}^{m} [\![g_i(\boldsymbol{X}) \text{ is satisfied}]\!]$$

where $[\![g_i(\boldsymbol{X}) \text{ is satisfied}]\!]$ is an indicator function equal to 1 if clause i is satisfied and 0 otherwise (i.e. all the literals in clause i are false).

Our focus will be on randomly generated instances, where each clause consists of k randomly chosen variables which are negated with probability of a half. We require each variable in a clause to be different and all clauses to be unique. We denote the number of variables by n, the number of clauses by m, and the ratio of clauses to variables by α. As mentioned in the introduction, the ensemble of problem instances undergoes a phase transition for large n, such that almost all instances are satisfiable below a critical ratio of clauses to variables of α_c and unsatisfiable above this critical value. The value of α_c depends on k; for $k = 3$ the critical value is $\alpha_c \approx 4.3$, while for $k = 4$ the critical value is $\alpha_c \approx 9.8$. One reason for concentrating on $k = 3$ rather than $k > 3$ is that, for larger k, we must use considerably larger ratios of clauses-to-variable to find hard instances than is necessary for MAX-3-SAT. This would slows down the search algorithm preventing us from collecting as much data as we could for $k = 3$.

B. GSAT

This paper focuses on heuristic search algorithms for MAX-SAT. As a baseline algorithm, we consider the classic GSAT local-search algorithm [21]. GSAT is a hill-climbing algorithm which at each step chooses to change the variable that gives the best fitness improvement. When there are multiple alternatives it chooses to change one of them selected uniformly at random. Superficially, GSAT may appear inefficient as it considers all variables at each step, however, through judicious book-keeping GSAT can be made so that a single step requires $O(\alpha\, K^2)$ computations. It is therefore extremely fast (notice that it does not scale with n) and, in consequence, is very difficult to beat. Appendix A describes the implementation details of GSAT including a rather subtle set data structure which sped up our implementation over any other implementation that we are aware of.

As well as using GSAT as a baseline algorithm we also used it to find local optima as part of our empirical investigation of the fitness landscape. This, however, requires a method to determine whether a local optimum has been reached. A local optimum consists of a set of connected configurations all with the same number of satisfied clauses (fitness), but none with a neighbour with more satisfied clauses (figure 1 illustrates our definition of global and local optima). To determine whether GSAT reaches a local optimum we switch to an exhaustive search algorithm which maintains a set of configurations at the current fitness, and a stack of configurations at the current fitness whose neighbours have not been visited. Initially the set and stack consist of the initial configuration. The configuration on the top of the stack is popped and its Hamming neighbours are searched. If a neighbour has a higher fitness then we know that we are not at a local optimum. If the neighbour is at the current fitness, but has not been previously seen (i.e. it is not in the set), it is added to the stack and set. We continue popping configurations from the stack until either a fitter configuration is found or the stack is empty (this is very similar to the best-first search algorithm for testing connectivity of a graph). For this exhaustive search all fitness computations can be done efficiently using the same book-keeping used to implement GSAT.

One of the limiting constraints in carrying out our analysis is the memory requirements of exhaustive search. This becomes prohibitive for large n (above a few hundred at $\alpha = 8$) and close to the phase transition when the constant-fitness plateaus become very large.

III. Landscape Analysis

In this section, we look at some of the properties of the fitness landscape for MAX-SAT and in particular we concentrate on the scaling behaviour of different quantities as the system size grows. This analysis reveals why problem instances become difficult for local-search algorithms as they become large.

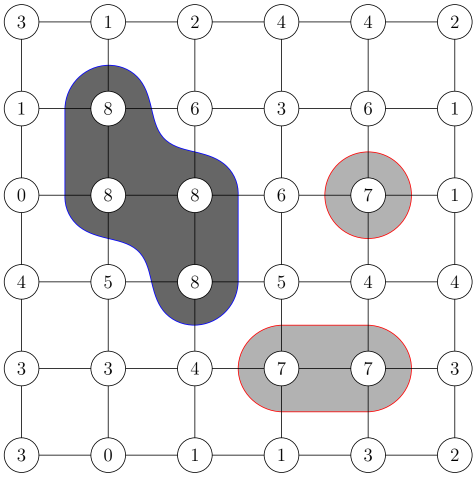


Fig. 1. Schematic illustration of our definition of global and local maxima. Assuming neighbours in the diagram are connected by edges, then there is one global maximum at fitness 8 consisting of four connected configurations (shown by the dark shaded region) and two local maxima of fitness 7 consisting of one and two configurations (shown by the lighter shaded region). Of course, the Hamming neighbourhood of MAX-SAT has a very different topology to the one shown here.

A. Density of States

We start by considering the number of configurations at each fitness level. The mean fitness for any MAX-k-SAT problem is $f_{av} = (1 - 2^{-k})\alpha\, n$. We can compute the spread of fitnesses around the mean through random sampling (this will miss rarely occurring fitness values). Figure 2 shows the logarithm of the histogram of fitnesses around the mean fitness scaled by $\sqrt{\alpha\, n}$ for single instances of size 100, 1000 and 10 000. The results are almost identical for all randomly drawn instances (data not shown). The curves in figure 2 are approximately quadratic indicating that the distributions are approximately normally distributed around their mean. The variance is empirically found to be around $0.1\alpha\, n$. This picture remains true for different values of α as shown in figure 3.

We can understand the behaviour of the density of states by assuming that the clauses are independent of each other. In this case, the fitness is just the sum of $\alpha\, n$ independent Boolean random variables, with a probability of $1 - 2^{-k}$ of being 1 (i.e. the clause is satisfied). By the central limit theorem, we would expect the distribution of fitnesses to be approximately normally distributed with mean $(1 - 2^{-k})\alpha\, n$ and variance $2^{-k}(1 - 2^{-k})\alpha\, n$. This is close to the observed behaviour around the mean fitness. For any particular instance, some of the clauses will share the same variables, and so their truth values are correlated. As a consequence, the distribution of fitnesses will deviate from a normal distribution, particularly close to its tails. Approximately normal behaviour of the density of states is observed in other models where the objective function consists of a number of approximately independent components. For example, this is true in a large number of

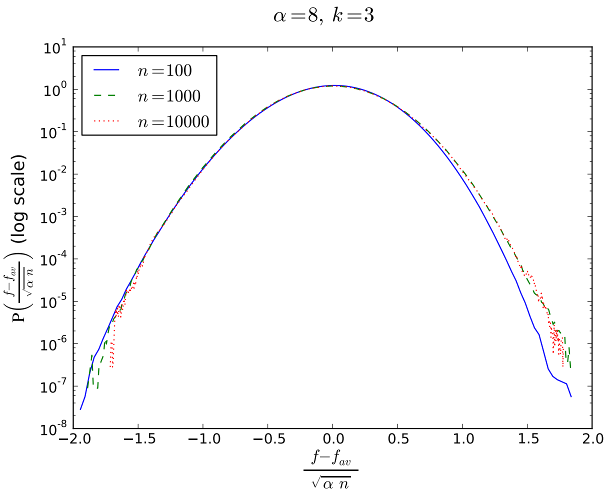


Fig. 2. Logarithm of histogram of fitnesses computed by sampling 10^9 random configurations for single instances with $\alpha = 8$ at $n = 100$, 1000 and 10 000.

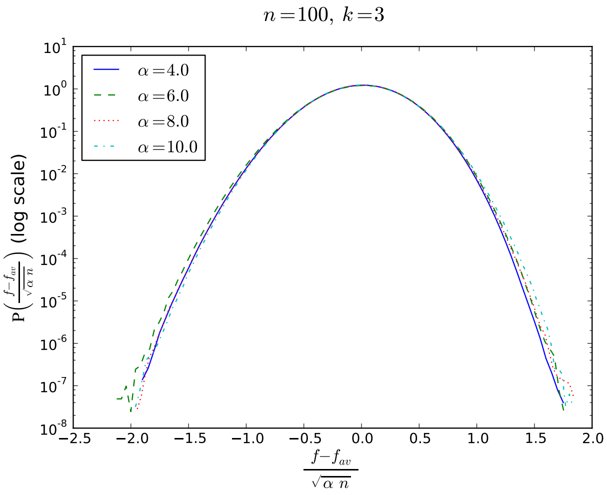


Fig. 3. Logarithm of histogram of fitnesses computed by sampling 10^9 random configurations for single instances with $n = 100$ at $\alpha = 4$, 6, 8 and 10.

constraint satisfaction problems where the objective function counts the number of violated constraints. This includes both easy problems such as onesmax and hard problems such as graph-colouring.

B. Auto-correlation

We can measure the auto-correlation of a random walk through the search space, which is often used as a measure of the ruggedness of the fitness landscape [11]. To compute this, we consider a random walk starting at an arbitrarily chosen initial configuration and moving to a Hamming neighbour at each step. Let $f(t)$ be the fitness at step t, then the auto-correlation is given by

$$\mathcal{R}(\tau) = \frac{1}{\sigma^2}\mathbb{E}\big((f(t+\tau) - f_{av})(f(t) - f_{av})\big)\,,$$

where σ^2 is the variance in the fitness for random configurations. We have computed the auto-correlation for the same three instances used in figure 2. These are shown in figure 4. We see that, under this scaling, the auto-correlations are remarkably similar. For large τ the auto-correlation function appears to drop off approximately exponentially as

$$\mathcal{R}(\tau) \sim \mathrm{e}^{-\tau/l},$$

where l is known as the correlation length [22]. Empirically $l \approx 0.4 \times n$. The correlation length is taken to be a measure of the landscape ruggedness—the smaller l the more rugged the landscape. We observe that, as n increases, the ruggedness decreases. That is, if we consider two instances of size n and n' we get roughly the same level of ruggedness if we make n'/n random steps on the instance of size n' as we would for a single step on the instance of size n.

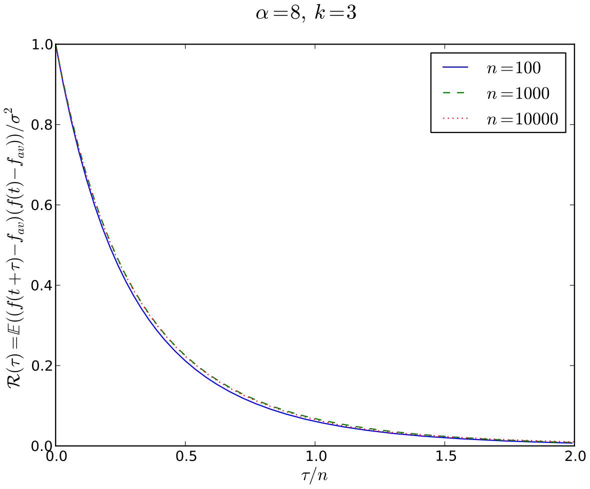


Fig. 4. Auto-correlation for three instances of size $n = 100$, 1000, 10 000 with $\alpha = 8$ plotted against the time difference τ/n.

Interestingly the correlation length does not alter significantly with α. This is shown in figure 5 (to emphases the approximate exponential fall off we have plotted the logarithm of $\mathcal{R}(t)$). The autocorrelation length suggests that the landscape is relatively smooth with long-range correlations. We will see the origin of the long-range correlation and evaluate one of its properties analytically in section IV.

C. Time to Local Optimum

In much of the analysis of MAX-3-SAT, we will study properties of the local and global optima. We begin this analysis by considering the time taken by GSAT to reach a local optimum. To check if a local optimum has been reached we use the exhaustive search algorithm described above, however, if we find that we have not reached a local optimum we carry on GSAT from where we left off. The time taken to reach a local optimum depends critically on α. For $\alpha \ll \alpha_c$ the time to reach a local (and usually the global) optimum is relatively short. However, for $\alpha \approx \alpha_c$ the time to reach a local optimum increases rapidly. This is illustrated in figure 6, where we show the mean and median times for GSAT to reach a local optimum plotted against α for instances of size $n = 50$. The large increase around $\alpha = 4$ is indicative of

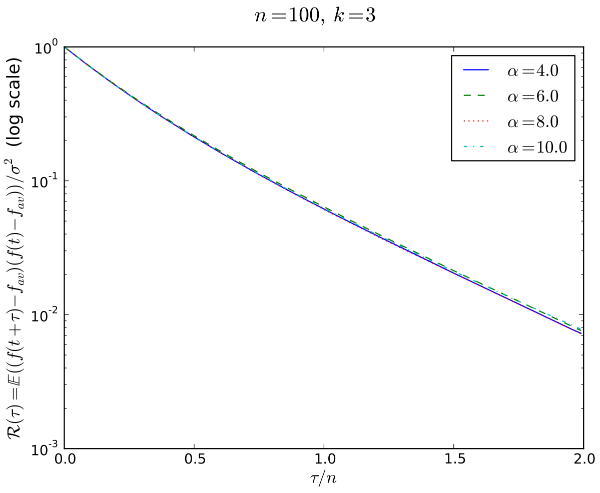


Fig. 5. Logarithm of the auto-correlation for four instances of size $n = 100$ and $\alpha = 4$, 6, 8, 10 plotted against the time difference τ/n.

large plateau regions in the solution space. We could compute this only for small instances as the computation time, and the memory requirement to check if we have reached a local optimum, becomes prohibitive around the phase transition (i.e. for $\alpha \approx \alpha_c = 4.3$). Interestingly, the big jump in the mean time taken to reach a local maximum around the phase transition is not reflected in the autocorrelation function, showing that the autocorrelation function is a comparatively poor indicator of the performance of search algorithms.

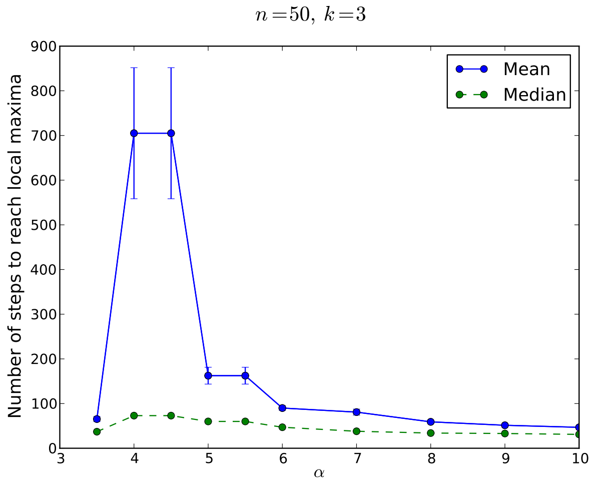


Fig. 6. Mean time to reach a local optimum versus the ratio of clauses to variables, α. Each data point represents the mean over 10 instances and 100 hill-climbs per instance.

This paper will concentrate on the regime $\alpha \gg \alpha_c$. For the sake of consistency we give results for $\alpha = 8$, although the same qualitative behaviour is observed at other values of α in this regime. As observed in figure 6, the mean is considerably higher than the median, indicating that the distribution of times to reach a local maximum has a long tail. Even away from the phase transition, the number of steps to reach a local optimum can vary considerably. Figure 7 shows this distribution plotted on a semi-log scale to emphasise the rare events. This data was gathered on a single randomly-chosen problem instance. We notice that, on rare hill-climbs, it can take a exceedingly long time to hit a local optimum even at $\alpha = 8$.

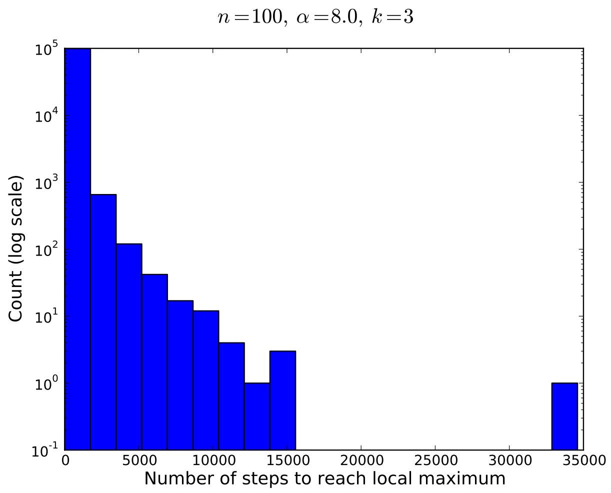


Fig. 7. Count of occurrences that number of steps to reach a local maximum occurs in each bin plotted on a logarithmic scale. The data is for a single instance with $n = 100$ and $\alpha = 8$ collected on 100 000 hill-climbs.

Figure 8 shows the mean time to hit a local optimum versus the problem size, n. The graph shows that the time increases super-linearly, but still polynomially. The super-linear increase is due to the growth in the size of the plateau regions. Plotting the same data on a log-log scale, figure 9, indicates that the time to reach a local optimum appears to increase sub-quadratically. The graph shows the best straight line fit to the data. Using this extrapolation, the mean time to reach a maximum for a problem of size 10 000 would be 660 000 steps, while the median time is 320 000. Although some caution is needed in extrapolating from small instances to large instance, nevertheless, for $\alpha = 8$ it is fairly clear that reaching a local optimum is not hugely demanding.

D. *Number of Local Optima*

What makes large MAX-SAT instances difficult in this regime (i.e. $\alpha > \alpha_c$) is the large number of local optima. To investigate this, we ran GSAT until there was no improvement in 100 steps. To check whether we reached a local optimum, the exhaustive search algorithm was run. If a fitter configuration is found then the exhaustive search is re-initialised from this fitter solution. This is repeated until a local optimum is reached. The full search is repeated from a large number of randomly chosen starting configurations. Each local maximum is recorded together with the number of times that it is hit. Of course, we have no guarantee that we have reached every local maximum, particularly if a local maximum has a small basin of attraction. Typically there are some local maxima which have very small probabilities of being visited. For example, in one typical instance with $n = 50$, after 10^7 hill-climbs, there were 375 local optima found of which 5 were discovered less than 10 times. For another randomly chosen instance this time

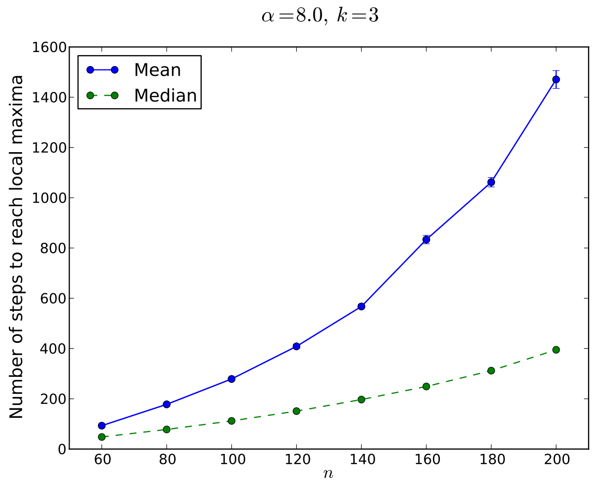


Fig. 8. Mean time for GSAT to reach a local optimum versus the problem size n for fixed $\alpha = 8$. Each data point represents the mean over 100 instances and 1000 hill-climbs per instance.

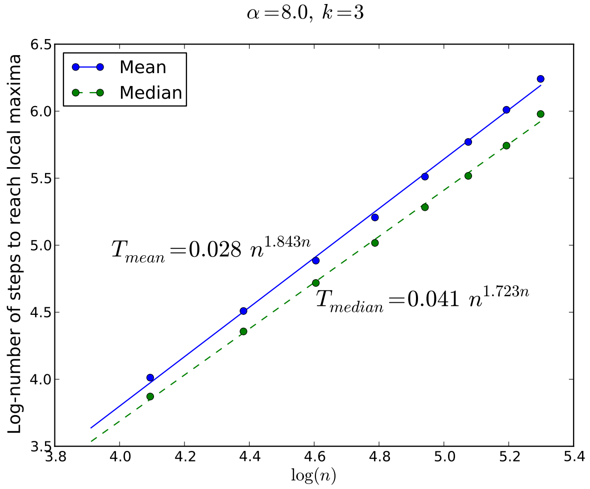


Fig. 9. Natural logarithm of the mean time to reach a local optimum versus the natural logarithm of the problem size. This shows the same data as in figure 8, but using a log-log axis. In addition, the best straight line fit to the data is shown.

with $n = 100$, again after 10^7 hill-climbs, there were 25 126 local optima found of which 13 764 where found less than 10 times and 5150 local optima which were visited only once. However, those local optima that have been observed only once are all low fitness local optima (fitness less than or equal to 774, where the maximum fitness is 782).

Figure 10 shows the mean and minimum number of times each local optimum was found in 10^7 attempts. Of course, there are likely to be local optima that we have not found. However, note that, for fitnesses greater than 776, we have visited each local optimum at least 100 times. Thus with high probability any local optima with a fitness greater than 776, that we have not found, would have a basin of attraction 100 times smaller than those that we have found. We call high fitness optima with abnormally small basins of attraction *elves*.

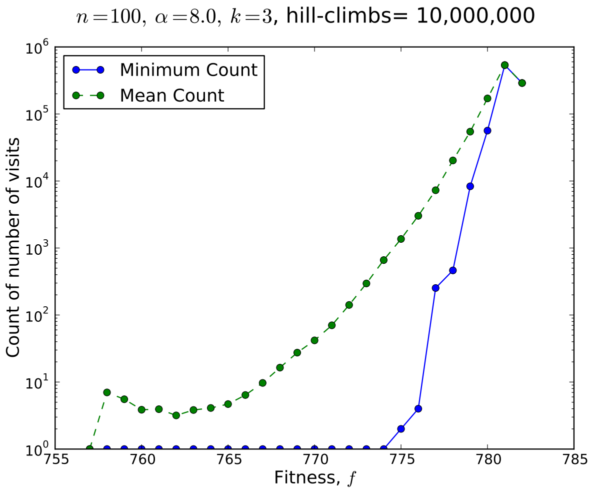


Fig. 10. Number of times local optima are hit in an experiment with 10^7 hill-climbs for a particular randomly drawn instance with $n = 100$ and $\alpha = 8$. This particular instance has a global optimum with an unusually small basin of attraction—hence the kink at high fitness. Also note, at fitness 774 we have visited each local optimum on average around 660 times, but there are 5 local optima at this fitness that have only been visited once.

Although we cannot rule out the existence of elves empirically, we strongly doubt their existence as we have not found any example of elves in a very large number of trials. Furthermore, as we will see later on, we have strong theoretical reasons to believe that high fitness optima will have relatively large basins of attraction (as is also found empirically), so it is unlikely that there can exist an elf. As a consequence, we believe that we have found all the global optima for the small problem instances we report. Of course, as the problem size increases the number of times we find the fittest observed optima decreases until we can no longer have confidence that a global optimum has not been missed.

Although, we can be confident for small instances of reaching all fit local optima, we are almost surely missing local optima with fitness $f < 775$. There exists a number of imputation techniques for estimating the true number of local optima at a given fitness level (see, for example, [23], [24]). A couple of papers have used these methods for estimating the number of local optima for MAX-SAT, but these have concentrated on instances close to the phase transition [25], [26]. We have tried applying a simple probabilistic model to our data. We assume that the probability of reaching a local optimum, i, is a random variable, Z_i, where all probabilities of the same fitness, f, come from the same gamma distribution

$$Z_i \sim \gamma(z|a_f, b_f) = \frac{b_f^a\, z^{a_f - 1}\, \mathrm{e}^{-b_f\, z}}{\Gamma(a_f)}.$$

The probability that local optimum, i, is visited n_i times in N trials is assumed to be Poisson distributed

$$\mathbb{P}(n_i|Z_i) = \frac{(N\, Z_i)^{n_i}\, \mathrm{e}^{-N\, Z_i}}{n_i!}.$$

The assumption of a Poisson distribution will be reasonably accurate given that each run is independent, and the probability of finishing in any given optimum is small. The choice of a gamma distribution is somewhat arbitrary, although, it is a common choice for modelling distributions of positive random variables, which frequently fits empirical data quite well.

Given a list of visiting frequencies for local maxima with fitness f, $(n_i|f_i = f)$ the likelihood of the data is given by

$$\mathbb{P}\big((n_i|f_i = f)|a_f, b_f\big) = \prod_{i\in\{i|f_i=f\}} \int_0^\infty \mathbb{P}\big(n_i|z_i\big)\,\gamma(z_i|a_f, b_f)\,\mathrm{d}z_i.$$

We could choose a_f and b_f to maximise this likelihood, however, this fails to take into account those optima that have not been visited. To correct for this, we include n_f^u unobserved local maxima at fitness f in our likelihood estimation, where n_f^u is equal to the expected number of unobserved local optima given a_f and b_f. Details of this calculation are given in appendix B. This inference technique appears to be quite accurate provided the expected number of times the local maxima at fitness f is visited is greater than 1 (this observation is based on running this inference procedure while holding out some of the data—results not shown). When the basins of attraction become smaller than this, the maximum-likelihood estimation breaks down. This is not too surprising as the only data we have, to determine the shape of the gamma distribution, comes from its tail.

Figure 11 shows the mean probability of visiting a local optimum of fitness f. The dashed lines show the estimation based on observed local optima (this is just the mean count of the number of visits as shown in figure 10 divided by N). The solid curve shows the estimated values using the maximum-likelihood estimator of a gamma distribution described above. This breaks down when the estimated probability goes below N^{-1} (where in our case $N = 10^7$). We have not shown the maximum-likelihood estimation below this point ($f < 768$) as the algorithm is numerically unstable. Obtaining a reliable estimation for the number of local optima at low fitness is extremely difficult and we have made no further attempt to do so in this paper.

In figure 12, we plot the logarithm of the number of observed local maxima divided by n, versus the fitness divided by $\alpha\, n$, for three particular instances of size 50, 75 and 100. We performed 10^7 hill-climbs to find the maxima. For $n = 75$ and $n = 100$ we significantly underestimate the true number of local optima at lower fitnesses. The plot shows that these curves are roughly similar for different n. It seems plausible that these curves could converge to a universal curve for sufficiently large n. This behaviour is consistent with the hypothesis that the number of local maxima grows exponentially with the system size.

The number of local optima also grows with α, although apparently not exponentially. This is shown in figure 13 where we have counted the mean number of local optima found in 10^6 hill-climbs plotted against α. With this number of hill-climbs, some fraction of the local optima will not be found, nevertheless, it is clear that the number of local optima grows, apparently linearly, with α.

As is evident from figure 11, the expected probability of visiting a local optimum grows as its fitness increases. The ratio of the mean probabilities of visiting a local optimum

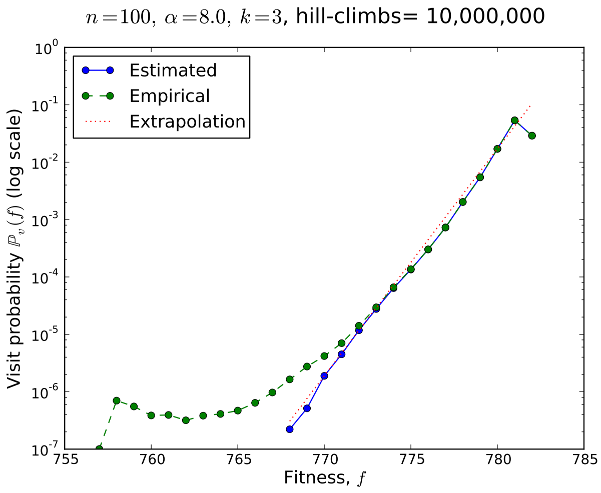


Fig. 11. Mean probability of visiting a local optimum of fitness f using the empirical data (assuming that all local optima have been found) and using the maximum-likelihood estimation described in the text. The extrapolation is just a straight line fit, $\mathbb{P}_v(f) \propto 2.49^f$

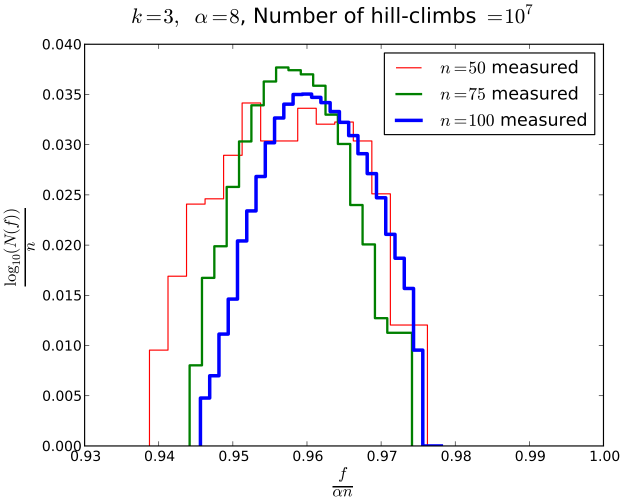


Fig. 12. Plot of the logarithm of the number of local maxima divided by n versus the fitness divided by $\alpha\, n$.

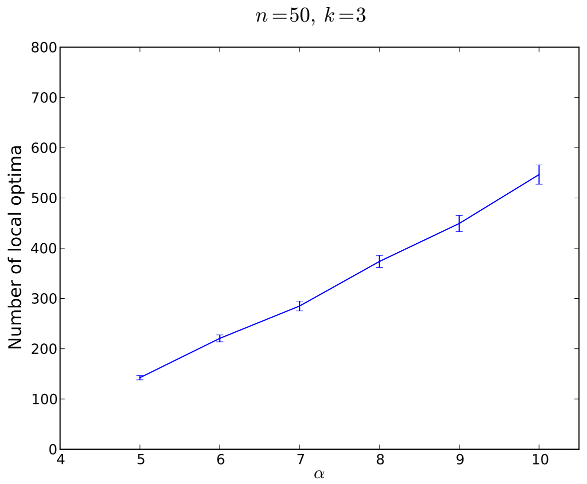


Fig. 13. Mean number of local optima found in 10^6 hill-climbs averaged over 100 instances versus the clause-to-variable ratio, α.

of fitness $f+1$, to visiting a local optimum of fitness f, is equal to around 2.1 for $n=50$, 2.47 for $n=75$, and 2.49 for $n=100$. A consequence of this is that there is clearly a strong bias towards high fitness local maxima. Figure 14 shows the proportion of local optima at each fitness value, and the probability of finding a local optimum using GSAT. Note that figure 14 shows the empirically measured proportion of local optima at each fitness level and thus underestimates the true proportion of local optima at low fitness.

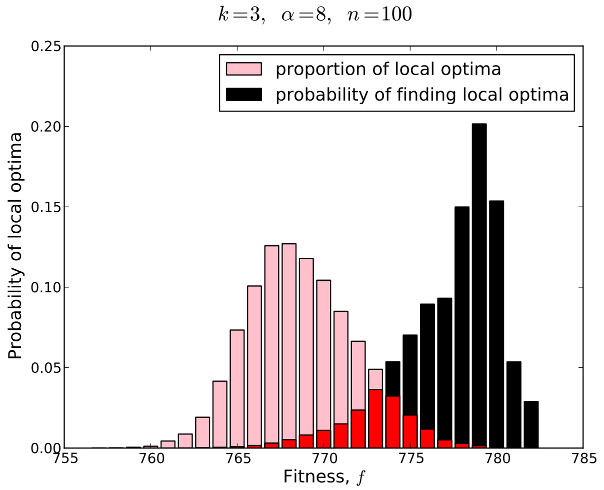


Fig. 14. Histograms showing the proportion of local optima at each fitness value and the probability of finding a local optimum of each fitness, for one instance.

E. Reaching the Global Optima

From the perspective of finding fit solutions, it is clearly a desirable property of MAX-SAT that the fitter local optima have, in expectation, larger basins of attraction. However, the gap between the expected fitness found and the fitness of the globally optimal solutions grows with the problem size, n. This is illustrated in figure 15, where we show the maximum fitness and the mean fitness of the local maxima found by GSAT averaged over an ensemble of randomly drawn instances—note that the fitnesses have been scaled by $1/(\alpha\, n)$ so the gap between the expected fitness found by GSAT and the maximum fitness appears constant for large n. For a single instance, the fitness of the maxima found by GSAT will fluctuate on every run. Figure 16 shows the variance in the fitnesses found by different runs of GSAT divided by $\alpha\, n$ plotted against the reciprocal of the problem size (we used exhaustive search to ensure that we reached a local maximum on each run). The plot is consistent with the variance in fitness growing linearly with the system size n. That is the size of the fluctuations (given by the standard deviation) scales as $\sqrt{n}$. In figure 15, we have also plotted one standard deviation around the average, estimated by extrapolating the empirically measured variance to large n. We see that instances become hard as n becomes large since the gap between the expected fitness of a local optimum found by GSAT and the globally optimal fitness grows linearly with n, while the standard deviation grows only as $\sqrt{n}$. Thus, the chance of finding a global optimum decreases as n increases.

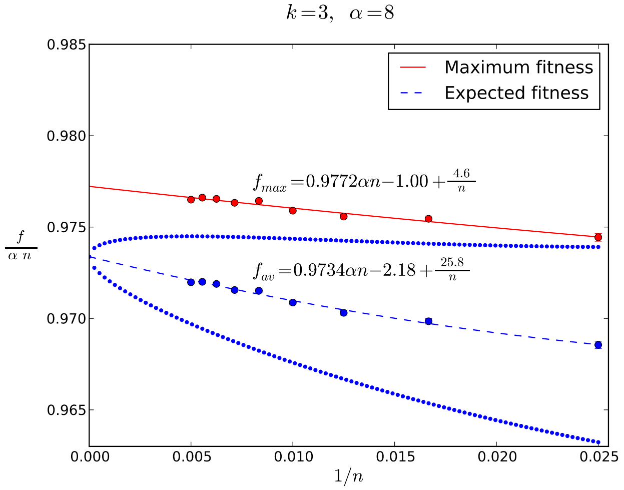


Fig. 15. Plot of the expected maximum fitness and the average fitness found by GSAT versus $1/n$. This is calculated by performing 10^5 hill-climbs on 400 randomly generated instances of the problem. We have computed the standard deviation in the fitness found by GSAT and show the mean plus and minus 1 standard deviation extrapolated for all n by the dotted lines.

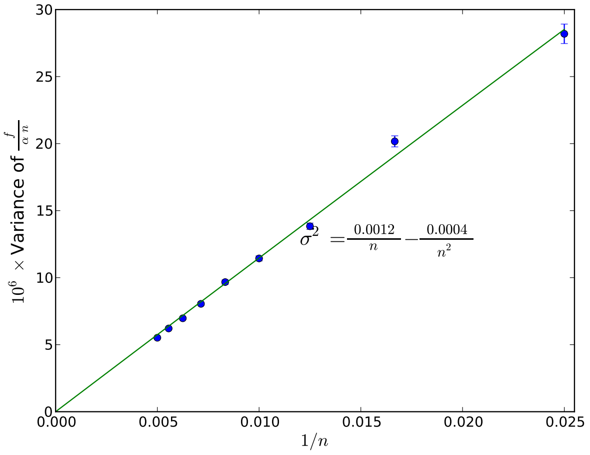


Fig. 16. Plot of the variance of $f/(\alpha\, n)$ found by GSAT averaged over 400 instances against $1/n$.

In figure 17, the logarithm of the probability of finding a globally optimal solution is plotted against n. The straight-line fit is consistent with the premise that the probability of finding a global maximum decreases exponentially with the size of the system. Using the straight-line fit in figure 17 to extrapolate to large n, we find the probability of GSAT finding a global optimum for an instance of size $10\,000$ would be around 10^{-76}. Although the value obtained from such an extrapolation is likely to be inaccurate, nevertheless, it provides a strong indication that the strategy of using multiple runs of GSAT takes exponential time (in the instance size).

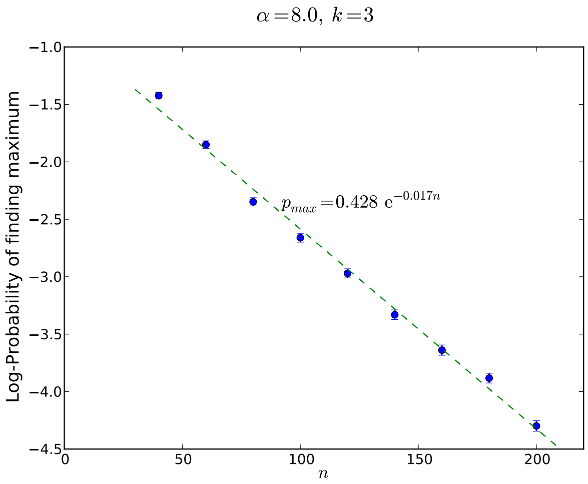


Fig. 17. Natural log-probability of finding the maximum fitness optima versus n averaged over 400 randomly generated instances.

F. Number of Global Optima

Figure 18 shows a histogram of the number of global optima for 10 000 random MAX-3-SAT instances with $n = 100$ and $\alpha = 8$. We observe that there is quite a wide spread in the number of the global optima. The expected number of global optima raises from 2.7 for $n = 40$ to around 4.0 for $n = 200$.

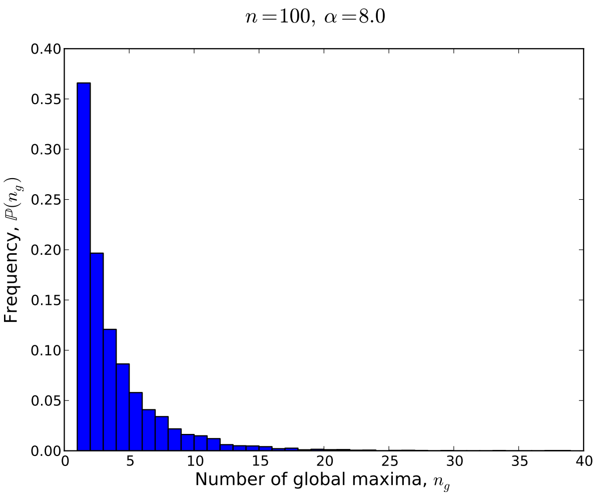


Fig. 18. Histogram of the number of global optima for 10 000 random MAX-3-SAT instances with $n = 100$ and $\alpha = 8$.

Note that we define each optimum to be a connected set of configurations at the same fitness with no fitter neighbours (see figure 1). Figure 19 shows a histogram of the number of configurations in each global optimum measured in 10 000 randomly generated problems. The tail of the distribution has been truncated in figure 19; there are 468 global optima with more than 200 configurations. The largest global optimum in these 10 000 instances has 2136 configurations.

In figure 20 the same data used in figure 19 is plotted on log-log axes. We see, for large n_f, that $\mathbb{P}(n_f)$ falls off close to $1/n_f$. This is an extremely fat-tailed distribution for which there is no mean. It is interesting to observe that although these instances are in many ways statistically similar, at least, in regard to the size of the global maxima the instances can have dramatically different properties.

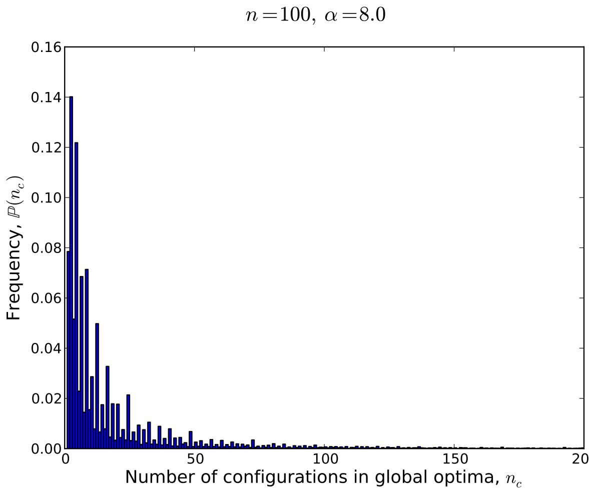


Fig. 19. Histogram of the number of configurations in each global optimum for 10 000 randomly generated problems with $n = 100$ and $\alpha = 8$. The histogram has been truncated with 468 global optima having more than 200 configurations and where the maximum is 2136.

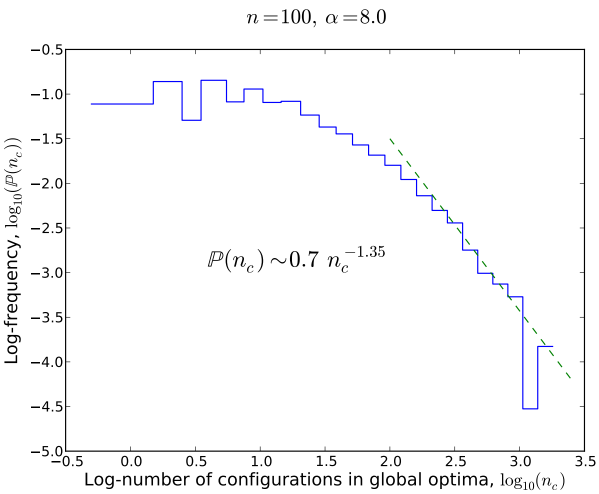


Fig. 20. Distribution of the size of global optima plot on a log-log axis. To plot this, the data in figure 19 has been put into bins of increasing size. The dashed line is a straight line fit corresponding to a distribution with a tail falling off as $0.7n_c^{-1.35}$.

The distance between global optima also varies from instance to instance. Figure 21 shows schematically the sizes of the global optima and the mean and minimum Hamming distances between the optima for one particular instance with 3 global optima. Figure 22 shows a histogram of Hamming distances between all configurations that make up the global optima for this particular instance.

In figure 23 we show a histogram of the Hamming distances between configurations at the global optimal fitness, averaged over 1000 instances, with $n = 100$ and $\alpha = 8$. The peak at low values is produced predominantly by the configurations

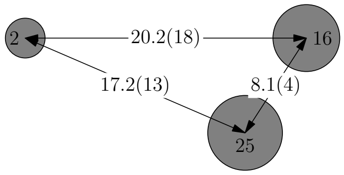


Fig. 21. Pictorial representation of the distance between the global optima in a particular instance of a problem with $n = 100$ and $\alpha = 8$. The number of configurations in the three global optima are 2, 16 and 25 while the average Hamming distance between global optima are 20.2, 17.2 and 8.1 with the minimum Hamming distance being 18, 13 and 4.

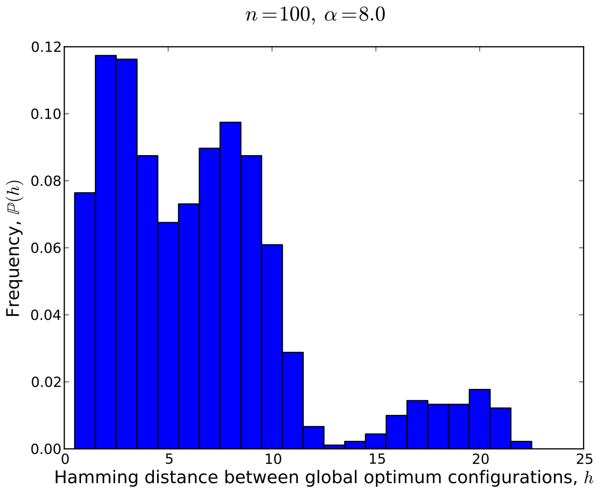


Fig. 22. Histogram of Hamming distances between configurations at the global optimal fitness for the same instance as shown in figure 21.

in the same global optimum, although there are also global optima with a minimum Hamming separation of 2. Notice that there is a wide spread of Hamming distance with some global optima having a Hamming distance greater than $n/2$. This may appear counter-intuitive as it shows that there are instances where there exist entirely unrelated ways of optimally solving these problems. However, there are $2^k - 1 = 7$ different ways in which each clause can be satisfied. Since the instances are constructed from random clauses, each variable occurs in roughly the same number of clauses, with the variables being negated roughly half the time. In a fit configuration, the truth value of the variables are chosen with respect to each other so that a large number of clauses are satisfied, but this careful balance can be achieved in many different ways. Thus, it is not too difficult to understand why there can be global optima which are a considerable Hamming distance apart. For more structured MAX-SAT problems, the balance of variables in each clause may be decidedly different so that good solutions may be more correlated. An early observation about SAT and MAX-SAT was that most instances were found to be easy to solve. Later it was found that random instances seem particularly hard [2]. A consequence of the structure of random instances of MAX-SAT is that a slight change to the set of clauses, although only resulting in a small change in the fitness for each configuration, can totally change the ranking of the local optima, and so may swap the position of the global optima to a quite different part of the search space. This sensitivity makes it very difficult to construct a heuristic for finding a global optimum.

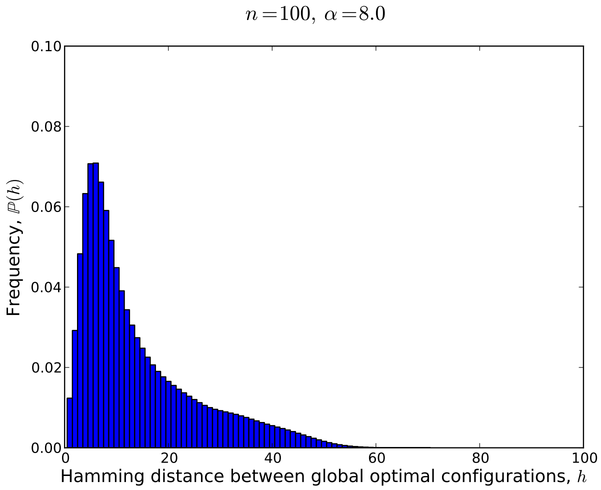


Fig. 23. Histogram of Hamming distances between configurations at the global optimal fitness for 1000 instances of MAX-3-SAT with $n = 100$ and $\alpha = 8$.

G. Distance between Optima

We have seen that, for large instances, GSAT will, with overwhelming probability, finish in a local optimum with a fitness considerably lower than the maximum fitness. This might not prevent an algorithm from finding a globally optimal solution, provided the global optimum was close to most good local optima. If this was the case, it might be possible to develop an algorithm which quickly moved from one local optimum to a fitter one. However, we show here that this hope is fruitless. Figure 24 shows the mean Hamming distance from a configuration in a local optimum to the *nearest* global optimum. These results are averaged over all local optima found in 100 instances.

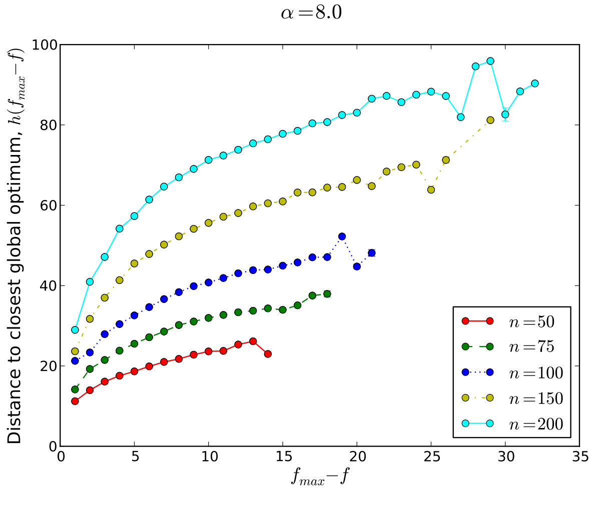


Fig. 24. Measure of the mean Hamming distance to the closest global optimum from a local optimum of fitness c.

We can collapse the curves in figure 24 onto a universal curve by rescaling the axes and fitting a single correction to scaling parameter. This is shown in figure 25 and demonstrates that the distance from a local optimum to a global optimum scales linearly with the problem size. The Hamming distance between an optimum of fitness $f_{max}-1$ to f_{max} clearly grows with increased n, although it is difficult to be sure how this Hamming gap grows. The answer to this question depends on extrapolating the curve in figure 25 to the y-axis. A plausible extrapolation would be that this gap was around $0.1n$, although it is not inconceivable that it extrapolates to zero indicating that the Hamming distance would be $o(n)$ and $\omega(1)$. As the number of configurations in a Hamming-ball of radius h grows as $\binom{n}{h} = \mathrm{e}^{\Theta(h)}$ (for $h > n/2$), the expected time to move from a sub-optimal local maximum to an optimal local maximum by searching the neighbourhood would (extrapolating from this data) appear to increase super-polynomially.

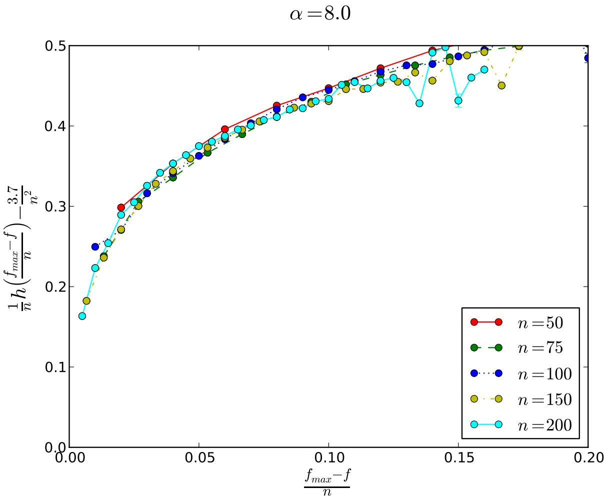


Fig. 25. Rescaled version of figure 24 showing that the curves collapse on to each other. To get a decent collapse we included a "correction to scaling" term ($-3.7/n^2$), where the parameter was chosen to fit the data.

IV. Landscape Correlations

The evidence presented thus far provides compelling reasons for believing that the landscape is difficult for a hill-climber to navigate. However, as we saw in the auto-correlation there exist significant long-range correlations in the landscape. This is a consequence of the structure of the objective (fitness) function. On average, each variable occurs in just $k \times \alpha$ clauses, with an equal probability of a clause depending on the variable or on its negation. The change in fitness due to flipping a variable will therefore be equal to the sum of the clauses that are satisfied only by that variable minus the sum of clauses that are unsatisfied, but are made satisfied by flipping the variable. Typically, this change in fitness is quite small. As a consequence, the configurations around a fit configuration also tend to be fit.

A. Expected Fitness in Hamming Sphere

We can analytically compute the mean fitness in a Hamming sphere around any configuration from knowledge of the number of satisfied literals in each clause. Let us denote the variables by $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ and the clauses by $g_i(\boldsymbol{X})$ where $i = 1, 2, \ldots, m = \alpha\, n$. We partition the clauses (for a given configuration $\boldsymbol{X}$) into equivalence classes, $\mathcal{S}_l$, depending on the number of satisfied literals in the clause. Thus, we write $g_i(\boldsymbol{X}) \in \mathcal{S}_l$ if clause i has l satisfied literals. Clearly l can take values 0 to k (the number of literals in each clause). We denote the indicator function using square brackets $[\![predicate]\!]$, which is equal to 1 if the *predicate* is true and 0 otherwise. We denote the size of the equivalence classes (i.e. the number of clauses with l satisfied literals) by

$$s_l(\boldsymbol{X}) = \sum_{i=1}^{m} [\![g_i(\boldsymbol{X}) \in \mathcal{S}_l]\!] .$$

Since the fitness, $f(\boldsymbol{X})$, of a configuration is equal to the number of satisfied clauses we have

$$\begin{aligned} f(\boldsymbol{X}) &= \sum_{i=1}^{m}\sum_{l=1}^{k} [\![g_i(\boldsymbol{X}) \in \mathcal{S}_l]\!] = m - \sum_{i=1}^{m} [\![g_i(\boldsymbol{X}) \in \mathcal{S}_0]\!] \\ &= m - s_0(\boldsymbol{X}). \end{aligned}$$

To compute the expected fitness in a Hamming sphere of radius h we average over all configurations $\boldsymbol{X}'$ with Hamming distance $d(\boldsymbol{X}', \boldsymbol{X}) = h$ from the configuration of interest, $\boldsymbol{X}$. We denote the set of configurations in this Hamming sphere by

$$\mathcal{X}_h(\boldsymbol{X}) = \left\{\boldsymbol{X}' \in \{T, F\}^n | d(\boldsymbol{X}', \boldsymbol{X}) = h\right\}.$$

We note that $|\mathcal{X}_h(\boldsymbol{X})| = \binom{n}{h}$. The expected fitness in the Hamming sphere is

$$f_h(\boldsymbol{X}) = \mathbb{E}\big(f(\boldsymbol{X}') | d(\boldsymbol{X}', \boldsymbol{X}) = h\big) = m - c_h(\boldsymbol{X}),$$

where $c_h(\boldsymbol{X})$ is the expected number of unsatisfied clauses (or cost) at a Hamming distance h from configuration $\boldsymbol{X}$,

$$\begin{aligned} c_h(\boldsymbol{X}) &= \mathbb{E}\big(s_0(\boldsymbol{X}') | d(\boldsymbol{X}', \boldsymbol{X}) = h\big) \\ &= \frac{1}{\binom{n}{h}} \sum_{\boldsymbol{X}' \in \mathcal{X}_h(\boldsymbol{X})} \sum_{i=1}^{m} [\![g_i(\boldsymbol{X}') \in \mathcal{S}_0]\!] . \end{aligned}$$

The number of unsatisfied literals in any clause must be in the set $\{0, 1 \ldots, k\}$ so that

$$\sum_{l=0}^{k} [\![g_i(\boldsymbol{X}) \in \mathcal{S}_l]\!] = 1.$$

Putting this into the equation for $c_h(\boldsymbol{X})$ we obtain

$$c_h(\boldsymbol{X}) = \frac{1}{\binom{n}{h}} \sum_{l=0}^{k}\sum_{i=1}^{m} \sum_{\boldsymbol{X}' \in \mathcal{X}_h(\boldsymbol{X})} [\![g_i(\boldsymbol{X}') \in \mathcal{S}_0]\!]\, [\![g_i(\boldsymbol{X}) \in \mathcal{S}_l]\!] ,$$

where we have reordered the summations (which we are clearly allowed to do as they are all over finite ranges). Now we observe that

$$\sum_{\boldsymbol{X}' \in \mathcal{X}_h(\boldsymbol{X})} [\![g_i(\boldsymbol{X}') \in \mathcal{S}_0]\!]\, [\![g_i(\boldsymbol{X}) \in \mathcal{S}_l]\!] = \binom{n-k}{h-l} [\![g_i(\boldsymbol{X}) \in \mathcal{S}_l]\!] ,$$

since, for the indicator functions to be true, we have to flip the l satisfied variables in clause i and leave the unsatisfied variables unchanged. This means we have to flip $h-l$ variables that are not in clause i (there are $n-k$ such variables). Thus, there are $n-k$ choose $h-l$ ways to flip these variables. Substituting this result into our expression for $c_h(\boldsymbol{X})$ we find

$$\begin{aligned} f_h(\boldsymbol{X}) &= m - \frac{1}{\binom{n}{h}} \sum_{l=0}^{k} \sum_{i=1}^{m} \binom{n-k}{h-l} \left[\!\left[g_i(\boldsymbol{X}) \in \mathcal{S}_l \right]\!\right] \\ &= m - \frac{1}{\binom{n}{h}} \sum_{l=0}^{k} \binom{n-k}{h-l} s_l(\boldsymbol{X}). \end{aligned}$$

This is easy to compute given $s_l(\boldsymbol{X})$ for $l = 0, 1, \ldots, k$. Obtaining the expected variance in the same Hamming sphere is much more complicated than the mean fitness as it depends on the similarity between clauses. We can, however, obtain a simple approximation for the variance—details are given in appendix C.

In passing, we note that Grover [15] defined a difference operator for a neighbourhood $\mathcal{N}(\boldsymbol{X})$ as

$$\nabla^2 f(\boldsymbol{X}) = \frac{1}{|\mathcal{N}(\boldsymbol{X})|} \sum_{\boldsymbol{X}' \in \mathcal{N}(\boldsymbol{X})} \big(f(\boldsymbol{X}') - f(\boldsymbol{X}) \big),$$

and defined a landscape to be *elementary* if it satisfies the wave-like equation

$$\nabla^2 f(\boldsymbol{X}) = \lambda f(\boldsymbol{X}),$$

for some λ. For a Hamming neighbourhood $\nabla^2 f(\boldsymbol{X}) = f_1(\boldsymbol{X}) - f(\boldsymbol{X})$, where for MAX-SAT, $f_1(\boldsymbol{X})$ is equal to

$$\begin{aligned} f_1(\boldsymbol{X}) &= m - \frac{1}{n} \big((n-k) s_0(\boldsymbol{X}) + s_1(\boldsymbol{X}) \big) \\ &= \frac{k\,m}{n} + \frac{(n-k)}{n} f(\boldsymbol{X}) - \frac{s_1(\boldsymbol{X})}{n}, \end{aligned}$$

so that

$$\nabla^2 f(\boldsymbol{X}) = \frac{k}{n} (m - f(\boldsymbol{X})) - \frac{s_1(\boldsymbol{X})}{n}.$$

The dependence of this on $s_1(\boldsymbol{X})$, which differs from configuration to configuration, prevents MAX-SAT from being an elementary landscape. However, MAX-k-SAT has been shown to be a superposition of k-elementary landscapes, so it has some intriguing algebraic properties [17].

Figure 26 shows a bar chart of the number of clauses with l satisfied literals ($l = 0$, 1, 2 and 3), for a configuration in the global optimum with the largest basin of attraction for a particular instance. Similar qualitative features are observed for all fit configurations.

In figure 27, we show the expected fitness of a configuration in a Hamming sphere of radius h from the configuration, $\boldsymbol{X}$, with $s_l(\boldsymbol{X})$ given in figure 26. The behaviour of the curve at large Hamming distances is predominantly determined by the number of configurations where all the literals are satisfied (i.e. $s_3(\boldsymbol{X})$)—this number is not terribly consistent between optima of the same fitness.

The qualitative shape of the curve is similar for all n, although the standard deviation in the fitness is of order $\sqrt{n}$. In figure 28 we show the expected fitness for an instance with $n = 10\,000$ and $\alpha = 8$ around a good solution found using landscape guided hopping described in section V-B. The

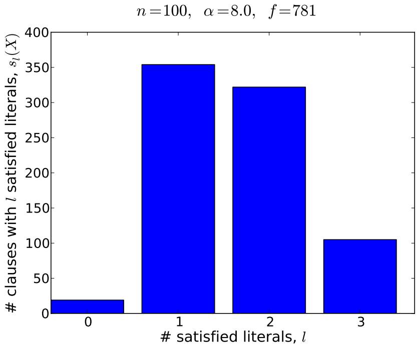


Fig. 26. Number of clauses with l satisfied literals is shown for one of the optimal configurations of a MAX-3-SAT problem with $n = 100$ and $\alpha = 8$.

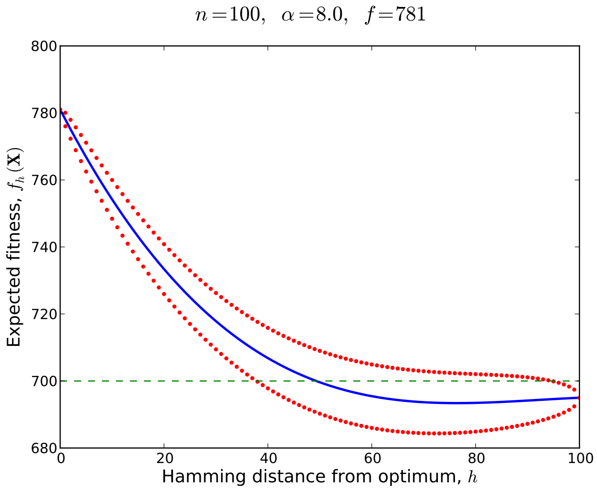


Fig. 27. Expected fitness of configurations in a Hamming sphere of radius h around the same configuration shown in figure 26. The dotted curves show one standard deviation around the mean. Note that the average fitness (shown by the horizontal dashed line) is at $(1-2^{-k})\alpha\,n = 700$.

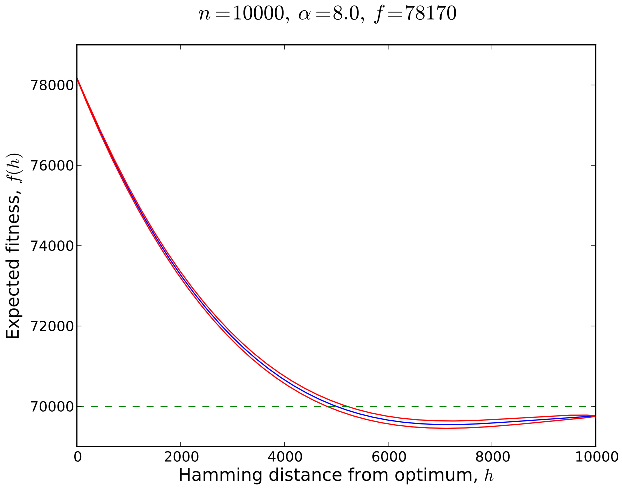


Fig. 28. Expected fitness of configurations in a Hamming sphere of radius h around a high-fitness configuration in a problem with $n = 10\,000$ and $\alpha = 8$. The number of clauses with l satisfied literals are $s_0(\boldsymbol{X}) = 1\,830$, $s_1(\boldsymbol{X}) = 36\,842$, $s_2(\boldsymbol{X}) = 31\,086$ and $s_3(\boldsymbol{X}) = 10\,242$.

solution is extremely unlikely to be a global optimum and may not be a local optimum.

If we scale the fitness by the system size we observe that the line for the average fitness is very similar to that of figure 27, although, the relative variance is clearly much smaller. Figure 29 shows the expected fitness for each Hamming sphere for the most frequently visited optima at each fitness where an optimum was found. At each fitness level, we chose the most frequently visited optimum. We note the same qualitative behaviour in all the curves, although there is some slight variation. A few of the curves cross each other, and differ markedly for large Hamming distances.

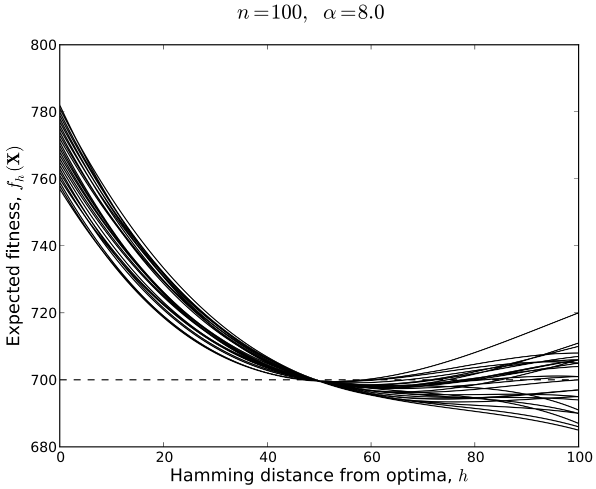


Fig. 29. Expected fitness in a Hamming sphere for the most frequently visited optima at each fitness where an optimum was found. This is for the same instance as that shown in figure 27.

These curves show the existence of large-scale correlations in the fitness landscape. That is, the presence of a local optimum changes the expected fitness of configurations away from the mean fitness at all Hamming distances. Despite this large-scale structure, local-search algorithms still fail to reliably find a global optimum. The reason for this is that the landscape is sufficiently rugged that it is not possible to exploit the long-range correlation using local fitness information alone. Figure 30 shows a density plot of the local optima as a function of their fitness and their Hamming distance from the most frequently visited global optimum. There is a small correlation between the quality of the local optima and the closeness to this global optimum, however, the vast majority of local optima are around $n/2$.

A problem instance would satisfies the big-valley hypothesis if the closer a local optimum is to the global optimum the fitter the optimum [18]. In MAX-3-SAT we often have multiple global optima, which, as shown in figure 23, can be a considerable distance apart. Furthermore, although there is a slight tendency for fit local optima to be close to a global optimum, there are plenty of fit local optima at a considerable distance from a global optimum, and unfit local optima close to a global optimum. Thus, these instances do not have a classic big-valley structure.

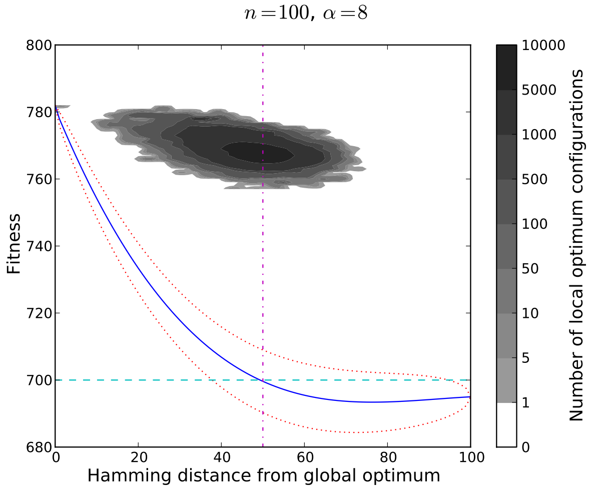


Fig. 30. Density plot of local optimum configurations as a function of their fitness and Hamming distance from the most frequently visited global optimum. We also plot the mean fitness in a Hamming sphere from the same global optimum. The shading of the point shows the number of configurations at that fitness and Hamming distance. Note the exponential scale on the density. We only show those local optimum configurations found using 10^7 hill-climbs, so we are likely to have missed many low-fitness optima.

B. Basins of Attraction

We can empirically measure the "basin of attraction" of a local optimum by repeatedly flipping a fixed number of variables and running GSAT until it reaches a local optimum. We then see if this is the same local optimum from which we started. We can thus measure the return probability starting from a configuration in a given Hamming sphere. Note that since GSAT is stochastic the basin of attraction is a probabilistic concept even at the level of individual configurations. Figure 31 shows the return probability for the most frequently visited global optimum. We observe that we have above a 50% chance of reaching the global maximum if we start within a Hamming distance of around 30. Figure 32 shows the return probability for one of the local optima with the smallest found fitness for the same instance as that shown in figure 31. This local optimum was found only once in 10^7 hill-climbs.

In principle, given the return probability, $p_r(h)$, it is straightforward to compute the probability of finding the local optimum starting from a random initial position. This is given by

$$\mathbb{P}\big(\text{finding local optimum}\big) = \frac{1}{2^n} \sum_{h=0}^{n} \binom{n}{h} p_r(h),$$

since the probability of starting in a Hamming sphere of radius h is $2^{-n}\binom{n}{h}$. Unfortunately, for return probabilities that fall off rapidly, this sum is dominated by the tail of the distribution which is effectively truncated when we measure $p_r(h)$ empirically. Using the empirically measured values of $p_r(h)$ thus severely underestimates the probability of reaching an optimum for optima with small basins of attraction. We can obtain a better estimate of the probability for finding a local maximum by approximating the tail of $p_r(h)$ by the best-fit exponential. Using this estimate we find the probability of finding the local maximum shown in figure 32 is approximately

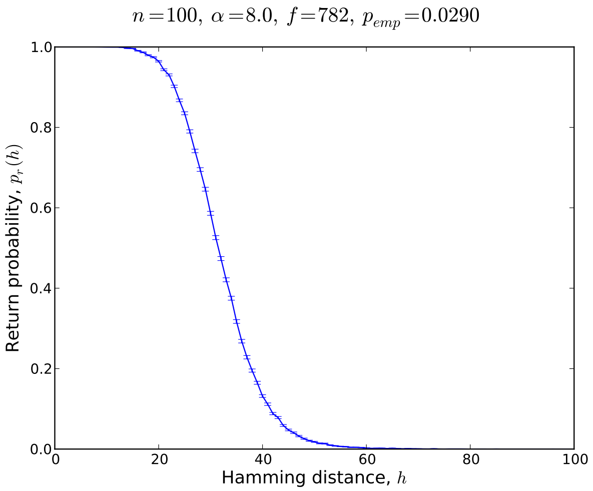


Fig. 31. Return probability starting from a Hamming sphere of radius h versus h. This is for the most probable global optimum solution of and instance with $n = 100$ and $\alpha = 8$. It has a probability of being visited of 0.029. The empirical probability is computed by running 10 000 hill-climbs starting at randomly chosen configurations in each Hamming sphere.

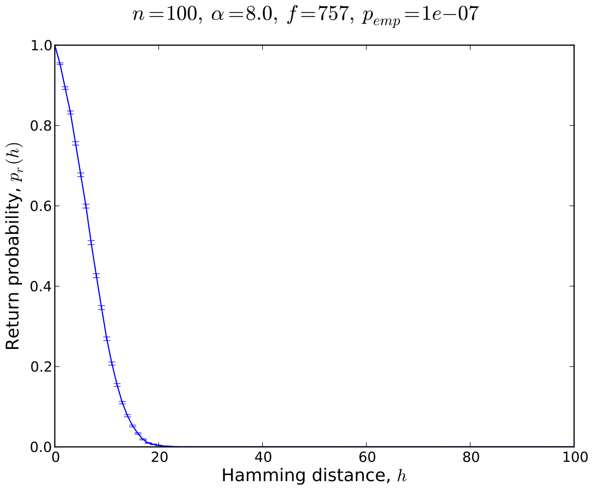


Fig. 32. Return probability starting from a Hamming sphere of radius h versus h. This is for a local optimum with the smallest fitness that we found. The local optimum was found once in 10^{-7} hill-climbs.

1.3×10^{-9}. Given that we only sampled 10^7 times it would appear that we were lucky to find a maximum with such a small basin of attraction. However, there are presumably so many such maxima that we are exceedingly likely to find at least one.

V. MAX-SAT Solvers

We have so far considered GSAT. In this section, we discuss an improved algorithm, WALKSAT, and a new class of heuristic search, *Landscape Guided Hopping* (LGH), inspired by the large-scale structure of the landscape.

A. WALKSAT

WALKSAT is a modification of GSAT which incorporates a stochastic step [27]. With a probability p either a GSAT move is made or else a walk move is made. In the walk move, an unsatisfied clause is chosen uniformly at random and one of the variables, again chosen uniformly at random, is flipped. Empirical studies show that the optimal probability of making a walk move, p, is around 0.5 throughout the run. In the results shown we have used $p = 0.5$.

WALKSAT is surprisingly effective. Over 20 local search algorithms have been implemented in the MAX-SAT solver framework UBCSAT [28]. On large random problem instances, with large α, we found GWSAT (a fast implementation of WALKSAT) consistently outperformed all other local search algorithms. Running for a short time, IROTS [29]—a Tabu search-based algorithm—obtained high-quality solutions unusually quickly, however, it runs out of steam and was outperformed by WALKSAT given sufficient time. Furthermore, WALKSAT is reported to perform at least as well, if not better, than GAs [30], PSO [31], ACO [32], and EDAs [33]. Note that WALKSAT will not get trapped in a local optimum due to its walk steps. As a consequence, we found GSAT more useful for investigating properties of the fitness landscape, since local maxima are well defined using this algorithm. We compare the performance of WALKSAT and GSAT in the next section.

B. Landscape-Guided Hopping

Here we analyse a new approach to solving hard optimisation problems we call *landscape-guided hopping* (LGH). This approach was first proposed in [1]. Landscape-guided hopping uses a population of independent hill-climbers to find fit solutions. These good solutions are then used to hop to a new part of the fitness landscape. This involves a very large move in the fitness landscape in terms of Hamming distance. The motivation for making this move is the existence of long-range correlation in the mean fitness as discussed in section IV.

We consider the simplest form of landscape-guided hopping where we choose the configuration which has the closest average distance to the set of solutions found by neighbour search. We call this algorithm *average landscape-guided hopping* (ALGH). Denoting the set of solutions found by neighbourhood search as $\mathcal{A}$ then we hop to a configuration

$$\boldsymbol{X}' = \underset{\boldsymbol{X}\in\{F,T\}^n}{\operatorname{argmin}} \frac{1}{|\mathcal{A}|} \sum_{\boldsymbol{X}\in\mathcal{A}} d(\boldsymbol{X}, \boldsymbol{X}'),$$

where $d(\boldsymbol{X}, \boldsymbol{X}')$ is the Hamming distance between $\boldsymbol{X}$, $\boldsymbol{X}'$. Since the configurations are binary vectors, $\boldsymbol{X}'$ can be found simply by choosing the most commonly occurring value for each variable in the strings (that is, if X_i is true in the set $\mathcal{A}$ more often than false then we set X_i' to be true; otherwise we set X_i' to false).

A more sophisticated version of landscape-guided hopping is to cluster the solutions and hop to the centroids of the clusters—we call this *clustered landscape-guided hopping* (CLGH). This algorithm was used heavily in [1]. We can get a slight advantage using CLGH if we run the algorithm multiple times as we can reuse the initial search of the population. The clustering provides some diversity in the starting position for WALKSAT. However, the performance gain is marginal, so we concentrate in this paper on the simpler ALGH algorithm.

In both forms of landscape-guided hopping, the fitness of $\boldsymbol{X}'$ is typically considerably lower than that of any of the solutions found by neighbourhood search. However, after applying local search the fitness rapidly improves and overtakes local search. In the results shown below we have used GSAT on a population of 100 individuals run for 10 000 iterations before applying landscape-guided hopping. Afterwards we apply WALKSAT. In the initial phase, the progress is 100 times slower for the population than a single GSAT solver since we have to perform local search independently on every member.

Figure 33 shows GSAT, WALKSAT and ALGH run on a randomly generated instance with $n = 10\,000$ and $\alpha = 8$. For WALKSAT and ALGH after hopping (where we use WALKSAT), we show the best fitness found so far rather than the current fitness (although, we have shown the drop in fitness caused by hopping). The fitness rapidly improves from the fitness of random solutions at $(1 - 2^{-k})\alpha n = 70\,000$ to a fitness around 78000. GSAT reaches a fitness of 78043 after 1776779 iterations and remains at that fitness until iteration 9907640 where it reaches 78044. WALKSAT is initially slower than GSAT, but continues to improve. For landscape-guided hopping, we show the average fitness of the population for the first 10 000 steps (since this is a population of 100 the total number of functions evaluations is 10^6). Averaging causes a loss in the fitness, but there is a remarkably rapid recovery where landscape-guided search overtakes GSAT and WALKSAT. This improvement is maintained.

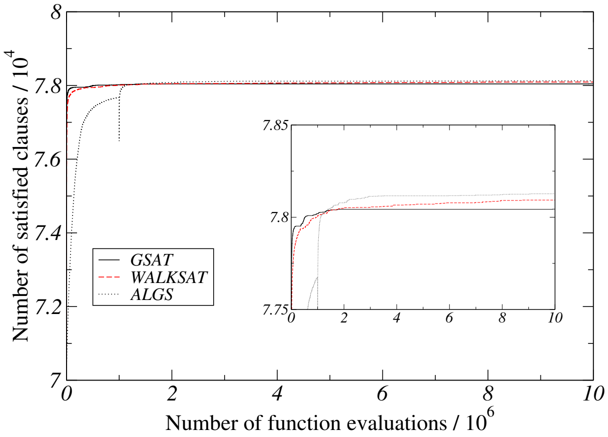


Fig. 33. Example of running GSAT, WALKSAT, ALGH and CLFH on a randomly drawn instance with $n = 10\,000$ and $\alpha = 8$. The insert shows the same figure but with the ordinate (y-axis) rescaled to show more clearly the region of high fitness.

In the initial phase of landscape-guided search, the population of hill-climbers slowly correlates as the fitter local optima tend to be correlated as illustrated in figure 30. Figure 34 shows the mean Hamming distance between independent configurations found by GSAT as a function of the number of iterations. The landscape-guided hopping algorithms exploit this correlation to hop to a part of the fitness landscape with higher-quality solutions.

To properly evaluate optimisation algorithms, we must look at the performance over a number of runs. In figure 35 we show a histogram of the fitnesses found by GSAT, WALKSAT and ALGH run for 1.25, 2.5, 5 and 10×10^6 function evaluations. ALGH used an initial population of 100 GSAT solvers run for 10 000 iterations—these values were chosen through initial experimentation as they seem to give satisfactory results. The remainder of the run used WALKSAT. The run times were not significantly different for any of the algorithms. Each algorithm was run 4000 times and a histogram of the best results found is plotted. GSAT is initially better than WALKSAT, although WALKSAT begins to catch up after around 1 000 000 steps. ALGH, after averaging, rapidly beats both GSAT and WALKSAT. This advantage is maintained as we continue to run the algorithms.

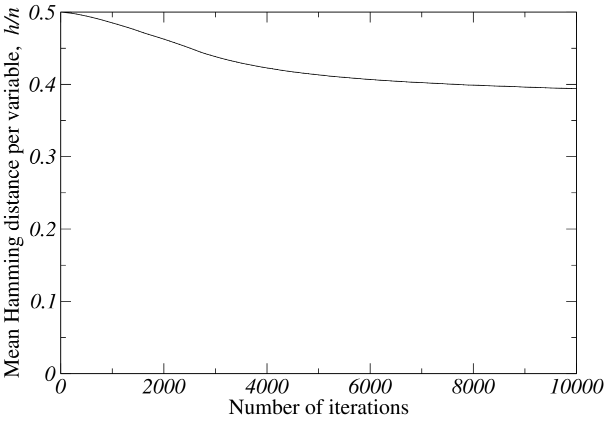


Fig. 34. The mean Hamming distance divided by n for a population of independent GSAT solvers for the same problem as that used in figure 33. The error bars are small than the thickness of the line and are not shown.

To get some idea of how close the results might be to the global optima we note that using the extrapolation in figure 15 the expected mean fitness found by GSAT is 77868. For this instance, GSAT reaches a mean fitness value of 78010 after 5×10^6 steps, and a fitness of 78013 after 10^7 iterations. The discrepancy could be due to an extrapolation error or simply the variation between instances. The expected maximum fitness predicted by extrapolation from figure 15 is 78177. The maximum fitness found by ALGH is 78166. In a much longer run we were able to find a solution with a fitness value of 78200; thus, the best solution we have found after 10^7 steps is still some way off the optimum for this instance. What is clear, however, is that ALGH substantially out-performs the state-of-the-art, WALKSAT. The mean fitness of the solutions found by ALGH in the runs shown in figure 35 was 78123. We run WALKSAT ten times longer (that is for 10^8 steps) and obtained on a sample of 20 such runs a mean fitness of 78112. Although, we carried out this analysis on a single instance, for instances this size, the beavioiur observed in figure 35 is typical. On all random instances we tried, of this size and greater, we found ALGH gave better performance that WALKSAT run for ten times as long.

VI. Conclusion

The picture of how MAX-3-SAT becomes difficult is known to be qualitatively similar to a large number of other hard optimisation problems. The analysis presented in this paper

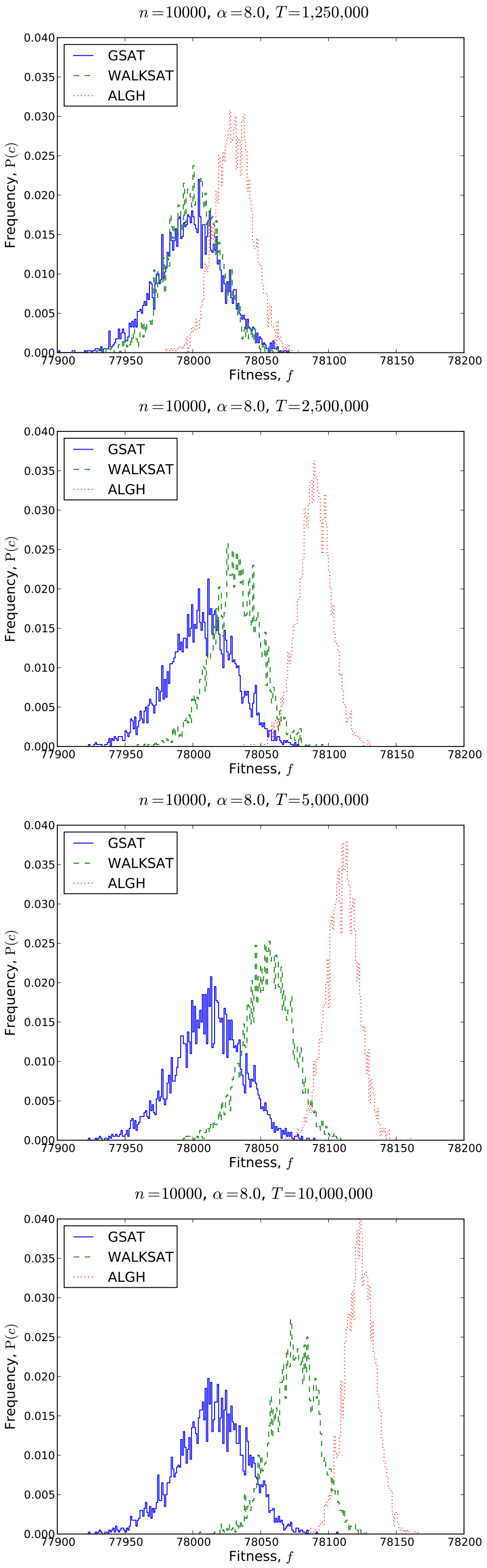


Fig. 35. Histograms of the fitness found by GSAT, WALKSAT, ALGH on a single randomly drawn instance with $n = 10\,000$ and $\alpha = 8$. The four diagrams show different number of function evaluations.

fills in the details. We see that there is an exponential growth in the number of local optima. Although, fitter maxima tend to have larger basins of attraction, this bias is not sufficient to out-weigh the raise in the number of local optima. The fitness gap between a typical local optimum found by GSAT and the global optima increases with n. The fluctuations between runs grow as $\sqrt{n}$ so the chances of reaching a global optimum decrease. The Hamming distance between local and global optima also seem to grow with n, so finding a fit local optimum does not provide significant information about the location of a global optimum.

WALKSAT improves on GSAT by making the search more stochastic. This seems to allow WALKSAT to discover fitter local optima with larger basins of attraction. Finally, we can take advantage of the correlation between fit configurations. This correlation follows from the fact that the average fitness around any configuration will change slowly with the distance from that configuration. We can use this to make large, beneficial hops across the search space by moving to the closest configuration to a set of fit solutions. This averaging is a rather crude way to hop over the search space, and there may well be better ways to exploit knowledge of the average large-scale structure.

As we previously observed, changing a single clause will only cause a small change in the fitness values, but can change the ordering of the optima, thus switching the position of the global optima a long distance in the search space. A heuristic, such as landscape guided hopping that is fairly robust is unlikely to be sensitive to a small change in the fitness values. Thus, although it clearly helps the search to move to a part of the search space with high-quality solutions, it seems highly unlikely that it systematically moves close to a global optimum. Of course, if $P \neq NP$ and random instances of MAX-SAT are truly hard instances as is commonly assumed, then we cannot expect any efficient algorithm to guide the search towards a global optimum.

Although, the qualitative behaviour we observe in MAX-3-SAT is likely to be generic to a large number of optimisation problems, there will also be differences that are key to developing a successful search strategy. For example, we studied MAX-3-SAT away from the phase transition. Close to the phase transition the landscape contains large plateau regions making it much harder to find local optima. Also, MAX-SAT is peculiar in that local neighbourhood search algorithms such as GSAT are so fast. That is, the time taken to make a single step does not grow with the instance size. This is likely to be one of the reasons that GSAT and WALKSAT are so hard to beat. For many problems it is not unusual that computing the difference in fitness between two neighbouring configurations is substantially faster than computing the fitness of a random configuration. This is one reason why neighbourhood search and hybridised search algorithms are often so effective. Nevertheless, MAX-SAT is at the extreme end of the spectrum in terms of the low cost of neighbourhood search. Another aspect of MAX-SAT, that makes it easy to work with, but may not be so common, is that the configurations are binary strings with all strings being viable solutions. For the random MAX-SAT instances, all configurations are equal likely to be a global optimum. This makes MAX-SAT easy to work with. Many operations such as finding the closest configuration to a set of configurations (necessary to implement landscape guided hopping) are particularly easy to perform for MAX-SAT. Yet another feature of MAX-3-SAT is that the rate of growth of the number of local optima is rather small, allowing us to reliably find all the global optima for instances with several hundred variables. All these features made MAX-3-SAT an easy problem for collecting the data presented in this paper.

To summarise the thesis of this paper: combinatorial optimisation problems are hugely diverse at the level of variables and their interactions. Despite this, there are surprising similarities of the landscapes on larger scales (see for example [34], [35]. For example, many problems become hard through the proliferation of local optima. This similarity in the behaviour over large Hamming distances is reminiscent of what physicists term *universality* where many long-range properties seem to be independent of the detailed short-range interactions. It is this commonality which provides the hope that search strategies are likely to have applicability to a wide variety of problems. However, there are important ways in which the long-range landscape structures of combinatorial optimisation problems can differ. It is our belief that the dimensions in which these differences occur are small enough to be catalogued. They include the ruggedness of the landscape, the speed of local search, the number of local optima, size of plateau regions, correlation between local optima, and the symmetries of the search space. Capturing these features requires a holistic study of the fitness landscape that looks at a number of different measurements. This paper attempts to provide such a study as a reference point. The next step to mapping out the difference that can affect the design and effectiveness of heuristic search strategies is to perform similar studies on other problems. We leave that task for future papers.

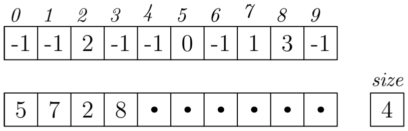
Appendix

A. Implementing GSAT

We can implement GSAT so that each step takes on average $\Theta(k^2\alpha)$ updates. To do so, we must maintain a count of the number of satisfied variables in each clause and the change in fitness caused by flipping each variable. A variable occurs on average in $k\,\alpha$ clauses. When a variable is flipped we have to change the count of the number of satisfied variables in each clause. We also have to change the cost for flipping variables for all the variables that occur in the clauses that have been changed by the variable flip. Since there are $k-1$ variables in a clause, other than the variable being flipped, we have potentially to change the flip cost of, on average, $(k-1)k\,\alpha$ variables. It is a tedious but straightforward exercise to perform the book-keeping described above.

A less trivial part of the implementation is to be able to choose which variable to flip given a list of flip costs. Since there are n variables, searching this list would dominate the run time. There are two types of variables that need to be maintained, improving moves and neutral moves. Typically, the total number of improving moves throughout a run will be less than $2^{-k}\alpha\,n$. Since the run time to reach a local optimum grows super-linearly, the vast majority of moves will not involve a fitness improvement. Therefore, the improving moves do not have to be implemented efficiently. The more challenging issue is to maintain a set of fitness-neutral moves. In particular we want to be able to add elements, delete elements and choose a random element in constant time. Binary trees or hash tables do not allow this; however, there is a simple, although not well known, data structure that does this. We call this a bounded set.

The bounded set implementation works provided that 1) the number of objects that could be put into the set are both known when the set is created, and 2) they can be stored in memory. In our case, our list of neutral moves can only involve the n variables—thus this easily fits in memory. We implement the bounded set using two arrays. The first, *index*, array is a fixed-size array of size n which is used to point to elements in the second array. The second, *member*, array contains the elements in the set. We also keep a counter of the number of elements in the set. If the elements are not in the set, we assign a value of -1 to those elements in the index array. In figure 36 we show a set of up to 10 possible elements containing elements 5, 7, 2 and 8.



0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
-1	-1	2	-1	-1	0	-1	1	3	-1

										size
5	7	2	8	•	•	•	•	•	•	4

Fig. 36. Illustration of the two arrays used to implement a bounded set containing elements 5, 7, 2 and 8. The top, index, array holds the indices of elements in the bottom, member, array.

To add an element, say 3, to the set we put it into the next available position (position 4) in the member array and update the element at position 3 in the index array to 4. We finally increment the set size. To remove an element, 5 say (assuming the array is as shown in figure 36), we check its position using the index array and find it is in position 0. We then move the last element, 8, in the member array into this position (updating the position of element 8 in the index array), decrement the element count, and set the 5^{th} element in the index array to -1. To find a random element we just select a random number from 0 to the size of the set and choose the element in that position in the member array.

WALKSAT can be run as efficiently as GSAT. The only modification needed is to keep a bounded set of the unsatisfied clauses.

B. Estimating the Number of Optima

In estimating the number of local optima, we treat each set of optima at a given fitness separately. To simplify the notation we drop the subscripts indicating the fitness, for example, we denote the parameters of the Gamma distribution by a and b rather than a_f and b_f. Recall from section III-D that we shall assume that the probability of visiting an optimum, i, is gamma distributed, and the likelihood of visiting the optimum in N trails is Poisson distributed. To compute the likelihood of visiting a local optimum, at fitness f, n_i times we integrate over the unknown probability, Z_i, of visiting the local optimum

$$\begin{aligned}\mathbb{P}\big(n_i|a,b\big) &= \int\limits_0^\infty \mathbb{P}\big(n_i|z_i\big)\,\gamma(z_i|a,b)\,\mathrm{d}z_i \\ &= \frac{b^a}{(b+N)^{a+n_i}}\,\frac{N^{n_i}}{n_i!}\,\frac{\Gamma(a+n_i)}{\Gamma(a)}.\end{aligned}$$

Given a list of the number of times each local optimum at fitness f is visited $(n_i|f_i = f)$, the log-likelihood is given by

$$\begin{aligned}\mathcal{L} &= \log\left(\prod_{n_i|f_i=f} \mathbb{P}\big((n_i)|a,b\big)\right) \\ &= n^t\big(\bar{n}\log(N) + a\log(b) - (a+\bar{n})\log(b+N) \\ &\quad - \log(\Gamma(a))\big) + \sum_i \left(\Gamma(a+n_i) - \log(n_i!)\right)\end{aligned}$$

where n^t is the total number of local optima at fitness f, and $\bar{n} = \sum_i n_i/n^t$ is the mean number of times a local optimum at this fitness is visited. To maximise the likelihood we set the derivative of $\mathcal{L}$ with respect to a and b to zero. After rearranging we find

$$a = \frac{\bar{n}}{\mathrm{e}^{\psi(a) - \frac{1}{n^t}\sum_i \psi(a+n_i)} - 1}, \qquad b = \frac{Na}{\bar{n}}$$

where $\psi(x)$ is the digamma function which is defined as $\psi(x) = \mathrm{d}\log(\Gamma(x))/\,\mathrm{d}x$. Notice that we have a self-consistency equation for a, which, can be easily solved by bisection. The function

$$f(a) = \frac{\bar{n}}{\mathrm{e}^{\psi(a) - \frac{1}{n^t}\sum_i \psi(a+n_i)} - 1}$$

has the property $f(0) = 0$ and $f'(0) = 0$, while for large a

$$f(a) \sim a + \frac{\sigma^2 - \bar{n}}{2\bar{n}} \pm O\left(\frac{1}{a}\right)$$

where $\sigma = \frac{1}{n_t}\sum_i n_i^2 - \bar{n}^2$. Thus, there must be a solution for finite $a > 0$ provided $\sigma^2 > \bar{n}$ since the function $f(a)$ starts out smaller than a for some sufficiently small a, but finishes larger than a. For $\sigma^2 \le \bar{n}$, the maximum-likelihood solution occurs when $a \to \infty$ which corresponds to $\gamma(x_i|a,b)$ being a delta function at $x_i = \bar{n}$. This agrees with our intuition that, in this case, we can explain all the fluctuations in the number of times an optimum is visited through the Poisson process, and thus the maximum-likelihood distribution occurs when all the optima have the same probabilities of being found.

To obtain an accurate model of the distribution of probabilities of finding an optimum, we should also include the optima that we have not observed. Of course, we do not know this. However, we can estimate this number for a given set of parameters, a, b and n^t The expectation of not finding a local optimum is given by $\left(\frac{b}{b+N}\right)^a$. The expected number, n^u, of local optima that are not observed are thus given by

$$n^u = n^t\left(\frac{b}{b+N}\right)^a = n^t\left(\frac{a}{a+\bar{n}}\right)^a$$

We have $n^t = n^u + n^o$ where n^o are the number of observed local optima. We can find n^u by starting from $n^t = n^o$ and computing the expectation of n^u. This is then used to update n^t. Eventually this converges although the convergence is rather slow and benefits from using Aitken's δ^2-process [36, Section 5.3]. Empirically it is found that if the probability of being visited in N trials falls below 1 then the likelihood is optimised by $a = 0$ and $n^u = \infty$. This is clearly a quirk of the probabilistic model we are using and reflects the fact that there is not enough data to make a reasonable fit. If we just extrapolate the expected probability of visiting a local maximum, we see from figure 11 that to visit a local maximum at a fitness of 757 at least once would require around 10^{11} samples which is computationally infeasible. Thus, obtaining a reliable estimate of the number of local optima at low fitness is extremely challenging.

C. *Fitness Variance in Hamming Sphere*

Computing the variance in the fitness of the configurations in a Hamming sphere depends on the structure of the clauses, and, in particular, which pairs of clauses share common variables. An exact computation of the variance, though possible becomes quite complicated. However, for randomly drawn instances where each clause is independent, we can obtain a reasonable approximation by assuming the probability of two clauses being satisfied is equal to the product of either being satisfied. This is true only in expectation for random configurations; however, it provides a reasonable estimate to the variance found in typical instances.

The variance is given by

$$\sigma^2 = \mathbb{E}_h\big(s_0^2(\boldsymbol{X}')\big) - \mathbb{E}_h\big(s_0(\boldsymbol{X}')\big)^2$$

where we have used the shorthand

$$\begin{aligned}\mathbb{E}_h\big(q(\boldsymbol{X}')\big) &= \mathbb{E}\big(q(\boldsymbol{X}')|d(\boldsymbol{X}',\boldsymbol{X}) = h\big) \\ &= \frac{1}{\binom{n}{h}} \sum_{\boldsymbol{X}'\in\mathcal{X}_h(\boldsymbol{X})} q(\boldsymbol{X}')\end{aligned}$$

for an arbitrary function q. Now

$$\begin{aligned}s_0^2(\boldsymbol{X}') &= \left(\sum_{i=1}^m [\![g_i(\boldsymbol{X}) \in \mathcal{S}_0]\!]\right)^2 \\ &= \sum_{i=1}^m [\![g_i(\boldsymbol{X}) \in \mathcal{S}_0]\!] \\ &\qquad + \sum_{i\neq j=1}^m [\![g_i(\boldsymbol{X}) \in \mathcal{S}_0]\!]\,[\![g_j(\boldsymbol{X}) \in \mathcal{S}_0]\!]\end{aligned}$$

where we use the fact that since an indicator function equals either 0 or 1, the square of the indicator function is equal to itself. Thus $\sigma^2 = \sigma_1^2 + \sigma_2^2$, where

$$\begin{aligned}\sigma_1^2 &= \sum_{i=1}^m \Big(\mathbb{E}_h\big([\![g_i(\boldsymbol{X}') \in \mathcal{S}_0]\!]\big) - \mathbb{E}_h\big([\![g_i(\boldsymbol{X}') \in \mathcal{S}_0]\!]\big)^2\Big) \\ \sigma_2^2 &= \sum_{i\neq j=1}^m \Big(\mathbb{E}\big([\![g_i(\boldsymbol{X}') \in \mathcal{S}_0]\!]\,[\![g_j(\boldsymbol{X}') \in \mathcal{S}_0]\!]\big) \\ &\qquad - \mathbb{E}\big([\![g_i(\boldsymbol{X}') \in \mathcal{S}_0]\!]\big)\,\mathbb{E}_h\big([\![g_j(\boldsymbol{X}') \in \mathcal{S}_0]\!]\big)\Big).\end{aligned}$$

(Note σ_1^2 and σ_2^2 are not themselves variances—σ_2^2 can and often is negative). As we showed in section IV-A

$$\mathbb{E}_h\big(\llbracket g_i(\boldsymbol{X}') \in \mathcal{S}_0 \rrbracket\big) = \sum_{l=0}^{k} p_l \llbracket g_i(\boldsymbol{X}) \in \mathcal{S}_l \rrbracket$$

where $p_l = \binom{n-k}{h-l}/\binom{n}{h}$. Since any clause is in just one equivalence class $\mathcal{S}_l$ then

$$\mathbb{E}_h\big(\llbracket g_i(\boldsymbol{X}') \in \mathcal{S}_0 \rrbracket\big)^2 = \sum_{l=0}^{k} p_l^2 \llbracket g_i(\boldsymbol{X}) \in \mathcal{S}_l \rrbracket .$$

Thus σ_1^2 is equal to

$$\sigma_1^2 = \sum_{l=0}^{k} (p_l - p_l^2) s_l(\boldsymbol{X}).$$

Since clauses are drawn independently, we would expect, for a random configuration, $\boldsymbol{X}$, that the probability that two clauses, i and j, are both satisfied is in expectation

$$\mathbb{E}_h\big(\llbracket g_i(\boldsymbol{X}) \in \mathcal{S}_0 \rrbracket \llbracket g_j(\boldsymbol{X}) \in \mathcal{S}_0 \rrbracket\big) = \\ \mathbb{E}_h\big(\llbracket g_i(\boldsymbol{X}) \in \mathcal{S}_0 \rrbracket\big)\, \mathbb{E}_h\big(\llbracket g_j(\boldsymbol{X}) \in \mathcal{S}_0 \rrbracket\big)$$

so that σ_2^2 cancel. This is only true when averaged over all possible clauses—it is not true for any particular pair of clauses. However, due to the large number of pairs of clauses it provides a acceptable approximation for typical randomly drawn instances. However, the configurations $\boldsymbol{X}$ we consider are not randomly drawn configurations, but are fit configurations so that the argument that they are independent of the clauses is flawed. This leads to a small but systematic discrepancy in the prediction of the variance. In figure 37, we compare the empirically measured variance in the fitnesses of configurations in a Hamming sphere of radius h from a global optimum configuration and the approximation assuming $\sigma_2^2 = 0$.

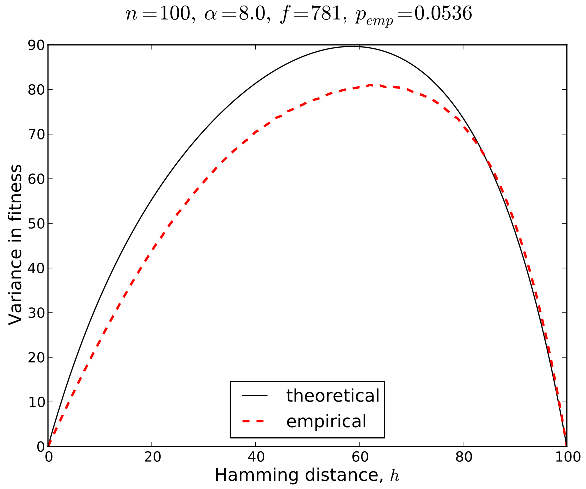


Fig. 37. Comparison of the variance in a Hamming sphere of radius h from a global optimum configuration measured empirically and using the approximation $\sigma_2^2 = 0$.

Although there is clearly a systematic error with the theoretical approximation, it translate into an almost negligible error when we consider the standard deviation. We illustrate this in figure 38, where we show both the mean fitness and the mean fitness ±1 standard deviation computed using the formula above and measured empirically. As we observe ,the approximation (i.e. assuming $\sigma_2^2 = 0$) provides a perfectly adequate estimate of the size of the fluctuations. In figure 28, where we show the fluctuations for an instance of size $n = 10\,000$, the discrepancy between the empirically measured fluctuations and the theoretical approximation are smaller than the thickness of the line.

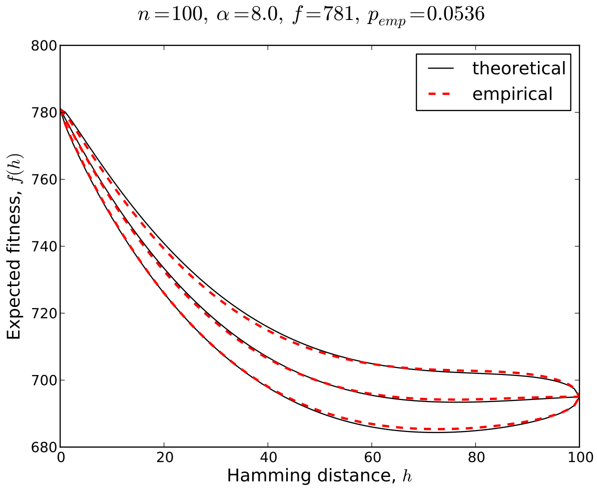


Fig. 38. Expected fitness in a Hamming-sphere of radius h. The mean fitness and the mean fitness plus and minus 1 standard deviation are shown. The empirical results are computed by sampling 10^6 configurations in each possible Hamming sphere.

As a final observation, we note that p_l is of order 1, while $s_l(\boldsymbol{X})$ is of order n, thus σ_2^2 is of order n and the fluctuations are of order $\sqrt{n}$.

References

[1] M. Qasem and A. Prügel-Bennett, "Learning the large-scale structure of the max-sat landscape using populations," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 518–529, 2010.

[2] D. Mitchell, B. Selman, and H. Levesque, "Hard and easy distributions of SAT problems," in *Proceedings of the Tenth National Conference on Artificial Intelligence*. San Jose, CA, USA: Publ by AAAI, Menlo Park, CA, USA, 1992, pp. 459–465.

[3] J. M. Crawford and L. D. Auton, "Experimental results on the crossover point in random 3-SAT," *Artificial Intelligence*, vol. 81, no. 1-2, pp. 31–57, 1996.

[4] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, "Determining computational complexity from characteristic 'phase transisition'," *Nature*, vol. 400, pp. 133–137, 1999.

[5] O. C. Martin, R. Monasson, and R. Zecchina, "Statistical mechanics methods and phase transitions in optimization problems," *Theoretical Computer Science*, vol. 265, no. 1-2, pp. 3–67, 2001.

[6] M. Mézard and R. Zecchina, "The random K-satisfiability problem: from an analytic solution to an efficient algorithm," *Phys. Rev. E*, vol. 66, p. 056126, 2002.

[7] A. Prügel-Bennett, "Symmetry breaking in population-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 1, pp. 63–79, 2004.

[8] D. Arclioptas, A. Naor, and Y. Peres, "Rigorous location of phase transitions in hard optimization problems," *Nature*, vol. 435, no. 9, pp. 759–764, 2005.

[9] M. Mézard, T. Mora, and R. Zecchina, "Clustering of Solutions in the Random Satisfiability Problem," *Phys. Rev. Letts.*, vol. 94, p. 197205, 2005.

[10] A. Montanari, G. Parisi, and F. Ricci-Tersenghi, "Instability of one-step replica-symmetry-broken phase in satisfiability problems," *Journal of Physics A: Mathematical and General*, vol. 37, no. 6, pp. 2073–2091, 2004. [Online]. Available: http://stacks.iop.org/0305-4470/37/2073

[11] E. D. Weinberger, "Correlated and uncorrelated fitness landscapes and how to tell the difference," *Biol. Cyber.*, vol. 63, pp. 325–336, 1990.

[12] T. Jones, "Evolutionary algorithms, fitness landscapes and search," Ph.D. dissertation, University of New Mexico, Albuquerque, 1995.

[13] P. Merz and B. Freisleben, "Fitness landscape analysis and memetic algorithms for the quadratic assignment problem," *Evolutionary Computation, IEEE Transactions on*, vol. 4, no. 4, pp. 337 – 352, Nov. 2000.

[14] L. Altenberg, "Fitness distance correlation analysis: An instructive counterexample," in *Proceedings of the Seventh International Conference on Genetic Algorithms*, T. Bäck, Ed. San Mateo, CA: Morgan Kaufmann, 1997, pp. 57–64.

[15] L. K. Grover, "Local search and the local structure of NP-complete problems," *Operations Research Letters*, vol. 12, pp. 235–243, 1992.

[16] P. Stadler, "Towards a theory of landscapes," *Complex Systems and Binary Networks*, pp. 78–163, 1995.

[17] A. M. Sutton, A. E. Howe, and L. D. Whitley, "A theoretical analysis of the k-satisfiability search space," in *Proceedings of the Second International Workshop on Engineering Stochastic Local Search Algorithms. Designing, Implementing and Analyzing Effective Heuristics*, ser. SLS '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 46–60.

[18] K. Boese, A. Kahng, and S. Muddu, "On the big valley and adaptive multi-start for discrete global optimizations," *Operation Research Letters*, vol. 16, no. 2, 1994.

[19] S. A. Cook, "Characterizations of pushdown machines in terms of time-bounded computers," *Journal of the ACM*, vol. 18, no. 1, pp. 4–18, Jan. 1971.

[20] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*. New York: W.H. Freeman and Company, 1979.

[21] B. Selman, H. J. Levesque, and D. G. Mitchell, "A new method for solving hard satisfiability problems," in *AAAI*, 1992, pp. 440–446.

[22] P. Stadler, "Landscapes and their correlation functions," *Journal of Mathematical Chemistry*, vol. 20, no. 1, pp. 1–45, 1996.

[23] J. Garnier and L. Kallel, "Efficiency of local search with multiple local optima," *SIAM J. Discr. Math., Vol*, vol. 15, pp. 122–141, 2000.

[24] C. R. Reeves and A. V. Eremeev, "Statistical analysis of local search landscapes," *The Journal of the Operational Research Society*, vol. 55, no. 7, pp. pp. 687–693, 2004. [Online]. Available: http://www.jstor.org/stable/4102015

[25] C. R. Reeves and M. Aupetit-Bélaidouni, "Estimating the number of solutions for sat problems," in *PPSN*, 2004, pp. 101–110.

[26] A. Albrecht, P. Lane, and K. Steinh
"ofel, "Analysis of Local Search Landscapes for k-SAT Instances," *Mathematics in Computer Science*, vol. 3, no. 4, pp. 465–488, 2010.

[27] B. Selman, H. A. Kautz, and B. Cohen, *Noise Strategies for Improving Local Search*. AAAI Press, 1994, vol. vol.1, no. July, pp. 337–343.

[28] D. Thompkins, "UBCSAT, The Satisfiability Library," 2009, May 9, [Online]. Available: http://www.satlib.org/ubcsat.

[29] K. Smyth, H. Hoos, and T. St
"utzle, "Iterated robust tabu search for MAX-SAT," *Advances in Artificial Intelligence*, pp. 995–995, 2003.

[30] S. Rana and D. Whitley, "Genetic algorithm behavior in the MAXSAT domain," in *Parallel Problem Solving from NaturePPSN V*. Springer, 1998, p. 785.

[31] A. M. Abdelbar and S. Abdelshahid, "Swarm optimization with instinct-driven particles," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, vol. 2. IEEE, 2004, pp. 777–782.

[32] M. Villagra and B. Barán, "Ant colony optimization with adaptive fitness function for satisfiability testing," in *Proceedings of the 14th international conference on Logic, language, information and computation*. Springer-Verlag, 2007, pp. 352–361.

[33] M. Pelikan and D. Goldberg, "Hierarchical BOA solves Ising spin glasses and MAXSAT," in *Genetic and Evolutionary ComputationGECCO 2003*. Springer, 2003, pp. 213–213.

[34] F. Krzakala and J. Kurchan, "Landscape analysis of constraint satisfaction problems," *Physical Review E*, vol. 76, p. 021122, 2007.

[35] A. K. Hartmann and M. Weight, *Phase Transitions in Combinatorial Optimization Problems*, 1st ed. Wiley, 2005.

[36] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computation*, 3rd ed. Cambridge, UK.: Cambridge University Press, 2007.

Mohammad-H. Tayarani-N. received a BSc and MSc degree in computer science from the Islamic Azad University of Mashhad, Iran in 2005 and 2008, respectively. He is currently studying for a PhD in the School of Electronics and Computer Science at the University of Southampton, UK. His main research interests include evolutionary algorithms, machine learning, and fractal image compression.

Adam Prügel Bennett received the B.S. degree in physics from the University of Southampton, Southampton, U.K., in 1984, and the Ph.D. degree in theoretical physics from the University of Edinburgh, Edinburgh, U.K., in 1989. He worked in research jobs in Oxford, Paris, Manchester, Copenhagen, and Dresden before finally returning to the School of Electronics and Computer Science, University of Southampton. He has been a Reader at the School of Electronics and Computer Science since 2007. His main research interests include evolutionary algorithms, machine learning, and bioinformatics.