Real-like MAX-SAT Instances and the Landscape Structure Across the Phase Transition

Francisco Chicano
ITIS Software, Universidad de Málaga
Málaga, Spain
chicano@lcc.uma.es

Gabriela Ochoa
Computing Science and Mathematics,
University of Stirling
Stirling, Scotland, UK
gabriela.ochoa@stir.ac.uk

Marco Tomassini
Information Systems Department,
University of Lausanne
Lausanne, Switzerland
marco.tomassini@unil.ch

ABSTRACT

In contrast with random uniform instances, industrial SAT instances of large size are solvable today by state-of-the-art algorithms. It is believed that this is the consequence of the non-random structure of the distribution of variables into clauses. In order to produce benchmark instances resembling those of real-world formulas with a given structure, generative models have been proposed. In this paper we study the MAX-3SAT problem with model-generated instances having a power-law distribution. Specifically, we target the regions in which computational difficulty undergoes an easy/hard phase transition as a function of clause density and of the power-law exponent. Our approach makes use of a sampling technique to build a graph model (a local optima network) in which nodes are local optima and directed edges are transitions between optima basins. The objective is to relate the structure of the instance fitness landscape with problem difficulty through the transition. We succeed in associating the transition with straightforward network metrics, thus providing a novel and original fitness landscape view of the computational features of the power-law model and its phase transition.

CCS CONCEPTS

• **Theory of computation** → *Mathematical optimization*; **Random search heuristics**.

KEYWORDS

Local optima networks, real-like MAX-SAT, phase transition, problem difficulty

ACM Reference Format:
Francisco Chicano, Gabriela Ochoa, and Marco Tomassini. 2021. Real-like MAX-SAT Instances and the Landscape Structure Across the Phase Transition. In *2021 Genetic and Evolutionary Computation Conference (GECCO '21), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3449639.3459288

1 INTRODUCTION

The Boolean satisfiability problem (SAT) is the prototypical NP-complete problem and has been used for decades to prove NP-completeness of a host of other combinatorial decision problems [16]. But SAT, in spite of its worst-case exponential complexity, is very important in applications too. There are several fields in which SAT can be applied such as formal verification, in both hardware and software, planning, cryptography and several others. This goes through first encoding specific problems into SAT instances and then solving them with a suitable SAT solver. For theoretical work, random uniform SAT instances are often preferred because they lend themselves better to mathematical treatment. However, it has been found that smaller uniform instances are actually much harder to solve and even intractable by the same state-of-the-art SAT algorithms that are extremely successful in solving large real-world instances quickly and efficiently [5, 11]. In fact, industrial SAT instances tend to have nonrandom clauses, i.e., they possess a structure and it is this structure that can be exploited by solvers.

A few decades ago it has been discovered that k-SAT instances in which n literals (variables and their negations) are distributed in a random uniform manner between m clauses of size k literals show what is called a *computational phase transition* for particular values of the critical parameter $\alpha = m/n$ [8, 17, 18]. For $k = 3$, i.e. 3-SAT, $\alpha \approx 4.26$. Before the critical value almost all instances are satisfiable and the cost of finding a solution remains low. But in the vicinity of the critical value the chance of generating satisfiable instances tends to zero and the cost of finding a solution grows exponentially with the number of variables n. This interesting behavior bears a formal resemblance to critical phenomena observed in the theory of phase transitions in statistical physics and has been intensively studied with the methods of this discipline [9].

MAX-kSAT is the optimization version of k-SAT, where the goal is to find an assignment of variables to maximize the number of satisfied clauses. Starting from a different viewpoint, in a previous study an attempt was made to understand what fitness landscape features change when one goes from easy underconstrained instances to hard overconstrained ones using uniform MAX-3SAT and the local optima networks (LONs) model [12]. The authors found that there is a single optimal plateau before the phase transition and many of them after the transition, qualitatively explaining the search difficulty. Ansótegui et al. analyzed a number of industrial SAT instances and found that they possess structures that sets them apart from random instances [1, 2]. In particular, they found that heterogeneity, locality, modularity and self-similarity are among the features that characterize real-world SAT formulas. Focusing on heterogeneity, i.e., the fact that few variables appear in many clauses and most variables appear in few clauses, Ansótegui et al. proposed models that can generate heterogeneous non-uniform instances according to various distributions, and in particular the scale-free model, in which the degree distribution of a variable follows a power-law [1]. Ansótegui et al. showed empirically that these model instances undergo a computational phase transition that depends on both the ratio α and the exponent of the power law β. In follow-up work, Bläsius et al. [4] studied the transition in more

978-1-4503-8350-9/21/07. https://doi.org/10.1145/3449639.3459288

detail, in an attempt to explain why SAT solvers that work well on industrial instances show poor performance on uniform random instances. Starting from Bläsius et al. analysis and results, in the present work we study the fitness landscapes of the corresponding MAX-3SAT instances using the LONs model. Our results show that the network metrics that can be computed from the LONs change before the easy/hard phase transition, anticipating the critical point.

The article has the following structure. We first describe and illustrate the difference between uniform and power-law distributed k-SAT instances in Section 2. After that, a brief introduction to the nature and terminology of local optima networks is provided in Section 3 to make the study self-contained. This is followed by a description of the methodology used in Section 4, including the selection of problem, the sampling method and the performance evaluation. Section 5 reports and discusses the results in terms of network metrics and through suitable visualizations and, finally, we conclude and suggest ideas for future work in Section 6.

2 RANDOM UNIFORM VS POWER-LAW DISTRIBUTED INSTANCES

To generate a uniform random k-SAT formula, k variables are randomly chosen in each clause and negated with probability $1/2$. In contrast, to build a scale-free distribution SAT formula one still samples each clause randomly but the probabilities of choosing a variable x_i to enter a clause depend on weights that are determined by the expression $Pr(X = x_i) \propto i^{1/(1-\beta)}$, where β determines the shape of the distribution. Finally, the variables are negated with probability $1/2$ [2, 4]. Using these weights, the probability distribution of the variable degree in the clause-variable interaction graph of the resulting k-SAT formula approximately follows a power-law distribution, that is, $Pr(degree(x) = d) \propto d^{-\beta}$. A graph representation of SAT instances is a useful tool to understand how variables are distributed among the formula clauses. There are various ways for doing this [19]; to illustrate the issue, here we use the *variable interaction graph* G. Each node v of G is a variable and there is an edge $\{v_1, v_2\}$ between two variables v_1 and v_2 when they appear together in at least one clause regardless of the variable's sign. Figure 1 provides a graphical illustration for two instances with $n = 100$ and $m = 300$. The left picture corresponds to a 3-SAT formula drawn uniformly at random, while the right picture corresponds to a 3-SAT formula drawn according to a power-law with $\beta = 2.5$. Although the graphs are too small to allow for accurate network statistics, the different structure is apparent. In the random uniform instance, variables have degrees that are close to the average value and G approaches a random graph. In the power-law case the degrees are more heterogeneous with many variables of low degree and a few with high degree, giving a fatter tail. To generate SAT formulas with a long tail variable distribution we used the scale-free model suggested by Ansótegui et al. [2] in the version published by Bläsius et al. in [4]. We direct the reader to [4] for details on how to generate scale-free formulas by using a power-law distribution in the choice of variables.

3 LOCAL OPTIMA NETWORKS

The local optima network (LON) model [13] is a tool to capture the structure of fitness landscape as seen by an algorithm that includes a local search heuristic. We use a variant of LONs, the compressed LONs (CLONs) [14], which allows modeling the global structure of landscapes with neutrality (i.e. existence of plateaus of local optima with equal fitness). We describe below the LON model, before introducing the compressed model (CLON).

3.1 LON Model

A fitness landscape [20] is a triplet (S, N, f) where S is a set of admissible solutions i.e., a search space, $N : S \longrightarrow 2^S$, is a neighborhood structure, a function that assigns a set of neighbors $N(s)$ to every solution $s \in S$, and $f : S \longrightarrow \mathbb{R}$ is a fitness function that can be pictured as the *height* of the corresponding solutions.

In our study, the search space is $\{0, 1\}^n$, i.e., the space of binary strings of length n, so its size is 2^n. As neighborhood, we consider the standard Hamming distance 1 neighborhood, that is, the set of all solutions at a maximum Hamming distance of 1 from the current solution.

Local optima. A local optimum, which in MAX-SAT is a maximum, is a solution l such that $\forall s \in N(l)$, $f(l) \geq f(s)$. Notice that the inequality is not strict, in order to allow the treatment of neutrality (local optima of equal fitness), which is known to widely occur on MAX-SAT. The set of local optima, which corresponds to the set of nodes in the network model, is denoted by L.

Perturbation edges. Edges are directed and based on the perturbation operator (p bit flips). There is an edge from local optimum l_1 to local optimum l_2, if l_2 can be obtained after applying a random perturbation (p bit flips) to l_1 followed by local search. Edges are weighted with estimated frequencies of transition. We determined the edge weights in a sampling process. The weight is the number of times a transition between two local optima occurred. The set of edges is denoted by E.

LON. Is the directed and weighted graph LON = (L, E), where nodes are the local optima L, and edges E are the perturbation edges.

3.2 Compressed LON Model

When the number of local optima is high in a LON it is difficult to analyze and visualize the structure of the landscape. A natural way of reducing the model size in landscapes with high-levels of neutrality is to redefine the nodes of the model. The compressed LON model (CLON) [15] joins the local optima that are connected components and have the same fitness value.

Compressed local optima. A compressed local optimum (also called a local optima plateau) is a set of connected local optima nodes (a connected component) in the LON with the same fitness value. This set of compressed optima, denoted by CL, corresponds to the set of nodes in the Compressed LON model.

Compressed Perturbation edges. The set of perturbation edges is defined as above for the LON model. However, after compression, there are no edges between nodes with the same fitness, as connected components with the same fitness become a single node. The set of edges among compressed nodes are also aggregated and their weights summed. We call this set compressed edges, CE.

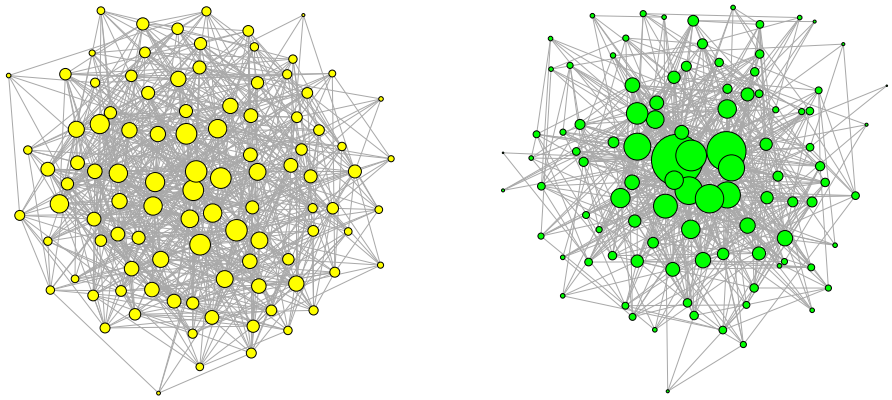


Figure 1: Left image: variable interaction graph of a random uniform SAT formula with $n = 100$ and $m = 300$. Right image: variable interaction graph for a power-law distributed formula with the same number of variables and clauses. Node size is proportional to the node degree.

Compressed LON. Is the directed graph CLON = (CL, CE), where nodes are the compressed local optima CL and edges the compressed perturbation edge set CE.

4 METHODOLOGY

4.1 Benchmark Instances

We used instances with $n = 500$ variables, as Bläsius et al. did [4]. We used the same instance generator as in [4] to generate random instances following a power-law distribution for the degree of the variables in the clause-variable interaction graph[1]. The number of variables is low compared to hard industrial SAT instances, but our focus is on the degree distribution, a feature characterizing industrial SAT instances independently of their size. The parameter β of the power-law distribution was chosen between 2.5 and 8.0 in steps of 0.5 (12 different values). The parameter $\alpha = m/n$ (clause to variable ratio) was chosen between 2.0 and 5.0 in steps of 0.2 (16 different values). Bläsius et al. found that the value of α in which the easy/hard transition for 3-SAT happens depends on β and varies from 2.4 to 4.2. The range of α we use is large enough to include the critical α value for all the values of β we use. For each combination of α and β we generated 100 different instances with random seeds from 1 to 100. In total 19 200 instances were generated ($12 \cdot 16 \cdot 100$). The generator was configured to generate clauses without duplicated variables.

4.2 Global Optima Computation

In order to compute relevant (C)LONs metrics and decorate the network visualizations with information about the global optimum we tried to find it in all the instances. With 500 variables, exhaustive methods to compute a global optimum are out of question and many exact methods also fail. Thus, in many instances we were not able to identify a global optimum. In these cases we used the best found solution as an estimation of the global optimum in the (C)LONs. We used three different approaches to identify the best solution or global optimum:

(1) First, we ran an Iterated Local Search (ILS, details in Subsection 4.3) to collect the information needed to build the (C)LONs and we stored the best solution found in any run of ILS for each instance. If the maximum number of satisfied clauses is m, then the instance is SAT and at least one global optimum was found.
(2) For the instances in which the best solution found by ILS is lower than m (7 826 instances), we ran MaxHS[2], one of the two best complete solvers in the MAX-SAT Evaluation 2020 [3]. We ran MaxHS with a time limit of 1 hour. In this way, we found a certified global optimum for 1 226 instances.
(3) We ran the MiniSat solver[3] for the remaining 6 600 instances in order to check if the instance is SAT or UNSAT. There were 314 (out of 6 600) instances that were SAT and, thus, have a certified global optimum of m. In the other 6 286 instances we have no certified global optimum and we use the best value found by ILS or MaxHS as the estimated global optimum.

4.3 Sampling Method

The sampling procedure consists of aggregating the local optima and transition edges obtained by 100 runs of the Iterated Local Search (Algorithm 1). The stopping condition was to reach a maximum number of 10 000 iterations or 5 000 iterations without any improvement in the number of satisfied clauses.

The ILS perturbation flips 10% of the variables selected at random (50 bit flips for $n = 500$). We used this value because in some preliminary experiments it gave good performance results compared to other values for the perturbation. The local search operator is a first improvement local search. If a flip in one bit of the current solution increases the number of satisfied clauses, the current solution is replaced by the new one. The local search is applied until a local optimum is reached (no neighbor can increase the number of satisfied clauses). A new local optimum is only accepted in Line 7 if it improves the incumbent solution. We report only the neutral and improving edges encountered between local optima in Line 6.

[1] https://github.com/RalfRothenberger/Power-Law-Random-SAT-Generator

[2] We used version 4.0, available at https://maxsat-evaluations.github.io/2020/

[3] We used version 2.2.0, available at http://minisat.se

Algorithm 1 Iterated Local Search

```
x_0 ← generateRandomSolution();
x ← applyLocalSearch(x_0);
while not stopping condition do
    y ← perturb (x);
    z ← applyLocalSearch(y);
    reportEdge(x,z);
    if f(z) > f(x) then
        x ← z;
    end if
end while
return x;
```

Worsening edges are counted but not reported. In particular, this implies that all the non-neutral local optima (and edges) visited after finding the global optimum are removed from the LON. This helps to avoid a bias in the metrics, since they generate only worsening edges and they do no not really provide additional information on the difficulty of the search process.

4.4 ILS and WalkSAT performance

We measured the performance of both ILS and WalkSAT. WalkSAT[4] is a well-known local search algorithm for SAT and MAX-SAT [10] with a wide influence among modern local search algorithms and is known to be very efficient in solving satisfiable random 3-SAT and MAX-3SAT instances.

We used the ILS described in Subsection 4.3, slightly changing the stopping condition, which is now to reach 10 000 iterations. As a measure of performance we use the number of iterations (visited local optima) to find a solution with the best known quality, identified using the approaches in Subsection 4.2. We ran ILS 100 times for each of the 19 200 instances and computed the median of the number of iterations as a measure of difficulty of the instance for ILS. In the case of WalkSAT, we did 100 restarts (runs) per instance with a maximum of 10 000 steps each and we collected the median number of steps as a measure of difficulty of the instance for WalkSAT.

5 RESULTS

For each of the 100 instances per α and β value combinations, we extracted the LON and CLON models and computed the network measurements described in Table 1.

Our aim is to identify fitness landscape features (in this case network metrics) that correlate with and help to explain the search difficulty of stochastic search algorithms. We therefore, consider a set of stochastic local search performance measures (described in Subsection 4.4). To estimate the location of the critical threshold, we follow a similar procedure as in [4], we compute the satisfiability of each formula as indicated in Subsection 4.2, and compute the proportion of formulas out of the 100 generated that are satisfiable.

Figure 2 shows the performance metrics as defined in Table 1. The vertical line indicates the critical $\alpha \approx 4.26$ for the uniform 3-SAT transition. The number of variables is $n = 500$. The top left image depicts the proportion of satisfiable formulas as a function of the clause-to-variable ratio α and for the power-law exponent β going from 2.5 to 8 in steps of 0.5. This is in very good qualitative agreement with the results in [4] confirming that the critical threshold depends on β and it becomes sharper with growing β, tending to the random uniform value 4.26 for large β. The number of WalkSAT steps (bottom left image) correlates very well with the problem hardness expressed by the SAT probability. The ILS results (right images) are in qualitative agreement with WalkSAT, in the sense that the ordering and shape of the curves is the correct one, but there is a systematic shift of the curves towards the right. We do not yet know the origin of this phenomenon but our hypothesis is that it originates in the sampling procedure performed by ILS, which is highly influenced by the landscape (more details in Subsection 5.1). We also observe that, in general, WalkSAT has a higher success rate than ILS.

[4]We used version 56, available at https://gitlab.com/HenryKautz/Walksat/-/tree/master/Walksat_v56

Table 1: Description of Metrics.

Hardness Metric	
prop-sat	Proportion of satisfiable formulas.
Performance Metrics	
walksat-steps	Number of bit flips (steps) by WalkSAT before reaching the global optimum (max. 10 000).
walksat-success	Success rate of WalkSAT, i.e. number of runs where the global optimum was achieved.
ils-steps	Number of perturbations by ILS before reaching the global optimum (max. 10 000 steps).
ils-success	Success rate of ILS, i.e. number of runs where the global optimum was achieved.
Network Metrics	
lon-nodes	Number of nodes (unique optima) in the LON model.
clon-nodes	Number of nodes in the compressed model CLON.
sinks	Number of sub-optimal nodes without outgoing edges.
strength	Incoming weighed degree (strength) of the global optima nodes.

5.1 Network Metrics

Once a system is modeled as a graph, many structural properties can be computed. The most basic metrics are the number of nodes and edges, but a variety of other metrics could be calculated related to the degree distribution, length of paths, community structure, centrality of nodes to name a few. To keep things simple, however, we concentrate on four network properties that help us to asses the landscape global structure. These metrics are summarized in Table 1. The number of unique optima sampled (*lon-nodes*) is the number of nodes in the (uncompressed) LON model. Since we constructed the compressed model (CLON), we also report the number of compressed optima (*clon-nodes*).

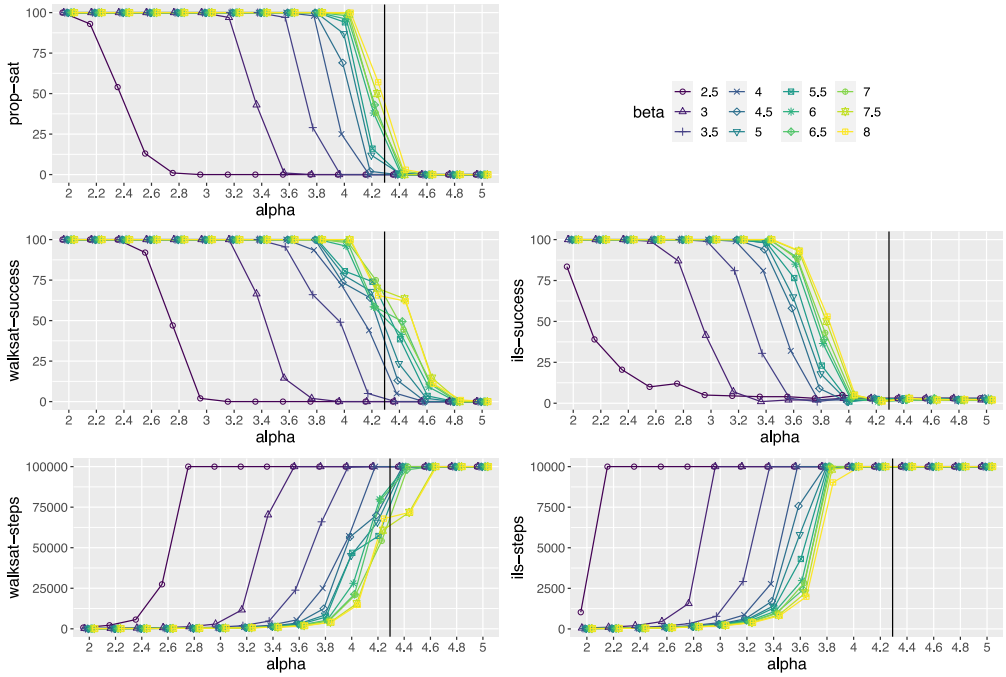


Figure 2: Proportion of satisfied formulas and median values of the performance metrics described in Table 1 for all values of α and β. The vertical line indicates the critical density of the uniform random 3-SAT model, $\alpha \approx 4.26$.

A funnel refers to a grouping of local optima, forming a coarse-level gradient towards a low cost solution at the end. Funnels can be considered as basins of attraction at the level of local optima. When sub-optimal funnels exist, search can get trapped and fail to reach the global optimum despite a large computing time. For characterizing the multi-funnel structure, we computed the number of *sinks*, that is the number of sub-optimal nodes without outgoing edges. Notice that the sinks represent dead ends in the search process, once a search trajectory reaches a sink, there is no way to escape to other local optima with the perturbation operator modeled. Therefore, sinks have been associated to the end of funnel structures [15].

The centrality of good solutions has been found to correlate with search difficulty [7]. As a measure of the centrality and reachability of the global optima (there can be more than one), we compute the incoming strength (weighted degree) of the global optimal nodes (*strength*). This is calculated as the sum of the incoming strengths of the globally optimal nodes.

Figure 3 shows the medians (across 100 instances) of the four network metrics for all the studied values of α and β. For all values of β, the number of compressed nodes (*clon-nodes* in Fig. 3) shows an opposite tendency than the number of uncompressed nodes (*lon-nodes* in Fig. 3). That is, while the number of unique local optima *lon-nodes* decreases, the number of compressed nodes increases with increasing α. This indicates that for low values of α, many unique local optima exist but they have the same fitness and are connected, which suggests that might be part of large plateaus. For the same α value, the number of nodes in the LON is larger, the lower the β value. The same tendency is observed for the number of *clon-nodes*, but the tendency is reversed around α =3.6.

Notice that the variation in the curves for the *lon-nodes* and *clon-nodes* metrics across the range of α values is smooth. However, for the *sinks* and *strength* metrics, there is a more sudden change in the shape of curves around a critical value of α. As observed for the performance metrics (Fig. 2), the location of this critical value depends on β and it becomes sharper with growing β. Both of these network metrics (*sinks* and *strength*) correlate well with the performance metrics of ILS (Fig. 2). They are also in qualitative agreement with the performance metrics for WalkSAT and the problem hardness (also in Fig. 2), but again show a systematic shift towards the left. These networks metrics (*sinks* and *strength*) seem to capture better the hardness phase transition than the metric related to the number of local optima (*lon-nodes*).

The number of sinks is zero or close to zero for low values of α, indicating that in this range, there are no sub-optimal funnels and thus all trajectories can easily reach the global optimum/a. This is supported by the high values of centrality (incoming strength) of the compressed nodes containing the global optimum/a. High values of the incoming strength indicate that the search trajectories converge into the global optima. This is congruent and correlates well with the performance metrics for ILS as mentioned above.

To shed some light on why the ILS performance and network metric curves could be shifted we present the following result.

Theorem 5.1. *For a distribution of random k-SAT instances with n variables and m clauses where a variable cannot appear twice in*

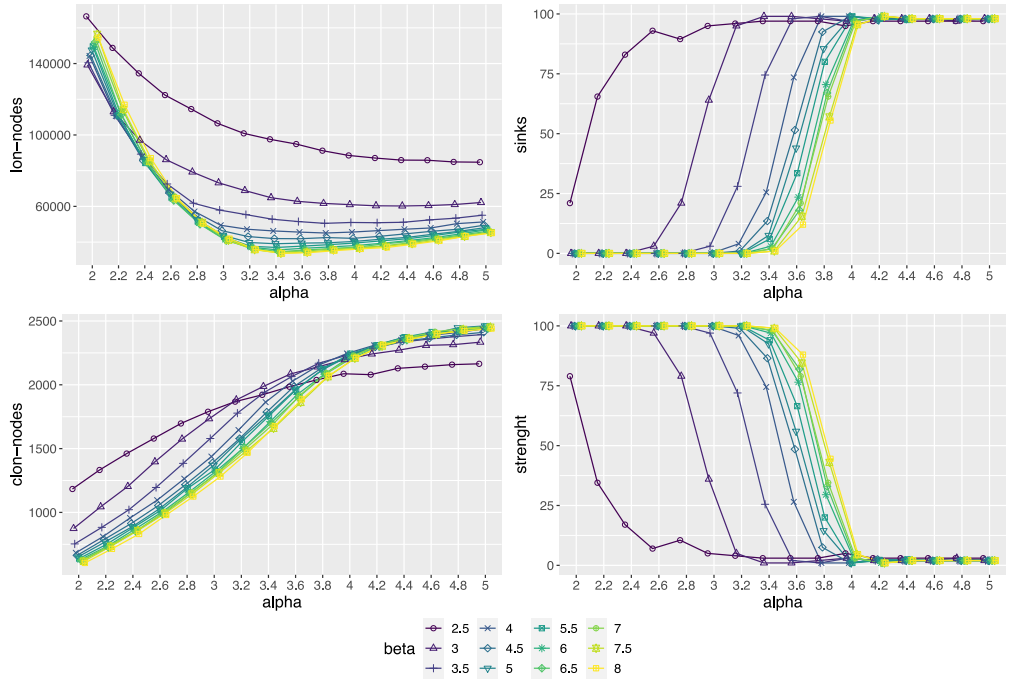


Figure 3: Median values of the networks metrics described in Table 1 for all values of α and β.

a clause and is negated with probability 1/2, the average number of SAT assignments is $2^n(1-2^{-k})^m = (2(1-2^{-k})^\alpha)^n$.

Proof. Given an instance $I \in \mathcal{P}$, let c_i denote the set of k variables in clause i and s_i the signs of the variables in the clause, the *signature* of clause i. An instance is characterized by a list $c_1, c_2, \ldots, c_m$ of sets of variables and a list $s_1, s_2, \ldots, s_m$ of signatures. The probability of selecting a particular signature in each clause is 2^{-k}. The probability of generating a particular instance is given by $p(c_1, c_2, \ldots, c_m)2^{-km}$, where $p(c_1, c_2, \ldots, c_m)$ is the probability of selecting the list $c_1, c_2, \ldots, c_m$ of sets of variables (observe that we do not assume that the variables in a clause are selected independently). The number of SAT assignments in a particular instance I is:

$$\#SAT(I) = \sum_{x\in\{0,1\}^n} \prod_{i=1}^{m} c_i^{s_i}(x), \tag{1}$$

where $c_i^{s_i}(x)$ is the evaluation of clause i, with variables in c_i and signature s_i in the assignment x (1 if the clause is satisfied and 0 if not). The expected number of SAT assignments when we consider all the instances in $\mathcal{P}$ with the probability distribution described above is:

$$\begin{aligned}
E[\#SAT(I)] &= \sum_{I\in\mathcal{P}} p(c_1,\ldots,c_m)2^{-km} \sum_{x\in\{0,1\}^n} \prod_{i=1}^{m} c_i^{s_i}(x) \\
&= \sum_{c_1}\cdots\sum_{c_m}\sum_{s_1}\cdots\sum_{s_m} p(c_1,\ldots,c_m)2^{-km} \sum_{x\in\{0,1\}^n} \prod_{i=1}^{m} c_i^{s_i}(x) \\
&= \sum_{x\in\{0,1\}^n}\sum_{c_1}\cdots\sum_{c_m} p(c_1,\ldots,c_m)2^{-km} \sum_{s_1}\cdots\sum_{s_m} \prod_{i=1}^{m} c_i^{s_i}(x) \\
&= \sum_{x\in\{0,1\}^n}\sum_{c_1}\cdots\sum_{c_m} p(c_1,\ldots,c_m)2^{-km} \prod_{i=1}^{m}\sum_{s_i} c_i^{s_i}(x)
\end{aligned}$$

(since each clause is satisfied in x for all except one signature)

$$\begin{aligned}
&= \sum_{x\in\{0,1\}^n}\sum_{c_1}\cdots\sum_{c_m} p(c_1,\ldots,c_m)2^{-km} \prod_{i=1}^{m}(2^k-1) \\
&= \sum_{x\in\{0,1\}^n}\sum_{c_1}\cdots\sum_{c_m} p(c_1,\ldots,c_m)2^{-km}(2^k-1)^m
\end{aligned}$$

(the sum of $p(c_1,\ldots,c_m)$ for all the lists of c_i is one)

$$= \sum_{x\in\{0,1\}^n} 2^{-km}(2^k-1)^m = 2^n(1-2^{-k})^m$$

□

The results of the theorem applies to the generator we are using in this paper. A direct consequence of Theorem 5.1 is that for fixed n and k, the average number of satisfying assignments decreases with α because $(1-2^{-k}) < 1$, and the landscape has to change also

with this reduction in the number of global optima. In particular, global optima plateaus decrease in size and/or frequency. This is in agreement with the observed change in the (C)LON metrics before the phase transition.

A second consequence of Theorem 5.1 is that the average number of SAT assignments is independent of the parameter β. Furthermore, the result applies also to k-SAT instances following a uniformly random distribution (all clauses are sampled with equal probability). Then, why do the curves of SAT probability (Fig. 2) depend on β? The reason is that the average number of SAT assignments and the SAT probability of an instance are not the same thing. Let p_i denote the probability that a random instance has i SAT assignments. Then:

$$E[\#SAT(I)] = \sum_{i=1}^{2^n} i \cdot p_i, \tag{2}$$

$$Pr[\#SAT(I) \geq 1] = \sum_{i=1}^{2^n} p_i, \tag{3}$$

which have different expressions. Definitely, β has an influence in the probability distribution p_i, but the average of this distribution is independent of β. For a fixed n and α, increasing β also increases the SAT probability, but keeps the average number of assignments constant. This can only happen if the value of p_i for high i is reduced while the value of p_i for low i in increased. That is, it is less probable to find instances with many SAT assignments and more probable to find instances with a few SAT assignments.

5.2 Network Visualization

An advantage of a network model is that it allows a graphical representation using the range of tools available for complex networks visualization. Here, we used the R statistics language and the igraph library [6] which implements a number of force-directed graph layout algorithms. Network plots are decorated to reflect features relevant to search dynamic. The LON compressed model allows us to visualize landscapes with several hundreds of unique local optima because each compressed node aggregates a large number of individual optima. It would not be possible to visualize these landscapes without compressing the nodes, and we argue that given the neutrality present in MAX-SAT landscapes, this compressed view is useful to understand the landscape's global structure.

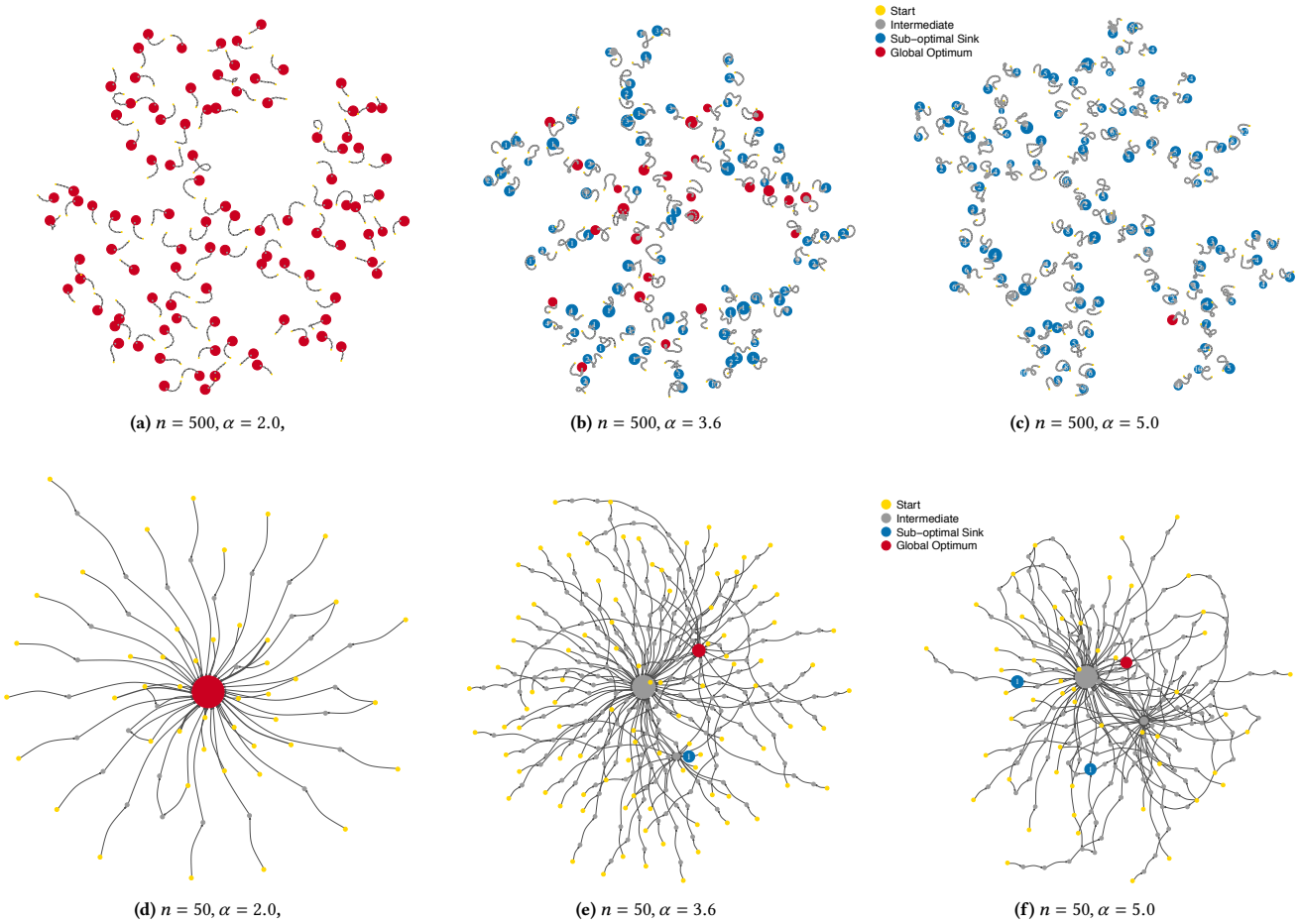
The networks in Figure 4 capture the whole set of sampled compressed nodes and edges for the CLONs of representative instances with $n = 500$ (top row), $\beta = 4.0$ and three values of α, which include the lowest ($\alpha = 2.0$) and the largest ($\alpha = 5.0$) explored, as well as one value located around the observed critical value ($\alpha = 3.6$). Each node is a compressed local optima and edges are perturbation transitions as defined in Subsection 3.2. The size of nodes is proportional to the number of unique local optima they aggregate and the color indicates the type of the nodes as detailed in the legend (right plots).

The CLONs for $n = 500$ always show 100 components (connected sub-graphs). This indicates that, in the sampling process, there is no convergence of the search trajectories. That is, the 100 separate ILS runs used to sample the LONs for each instance traverse different portions of the search space, and the visited local optima do not overlap. What the network images clearly indicate is that when $\alpha = 2$, all the 100 trajectories reach the optimum fitness value (red nodes at the end of all trajectories). It might be that they reach different portions of the same plateau, we cannot tell as we are sampling a huge search space. It can also be noticed that the trajectories towards the global optima are shorter for $\alpha = 2.0$. With increasing α the trajectories become larger and an increasing number of components do not reach the global optimum value, but instead end in a sub-optimal sink with worse fitness. These sub-optimal sinks are indicated as blue nodes with a numerical label showing how different they are in fitness from the global optimum. For $\alpha = 3.6$, several blue nodes appear and their number as well as their fitness differences from the global optimum is much higher when $\alpha = 5.0$.

After visualizing the CLONs for $n = 500$, we found it unsatisfactory that the trajectories do not overlap and thus offer a limited grasp of the landscape structure. Therefore, we conducted a similar study now for a smaller search space with $n = 50$. The network visualizations in Figure 4 (bottom row) capture the CLONs of representative instances with $n = 50$, $\beta = 4.0$ and the same four values of $\alpha = \{2.0, 3.6, 5.0\}$. In these images we can now see that the 100 search trajectories do converge and visit overlapping areas of the search space. As indicated in the color legend, the yellow nodes are the start of the trajectories, gray nodes are intermediate local optima, and red nodes indicate the global optima. We can observe that for $\alpha = 2$, the landscape is very easy to traverse with all trajectories converging into a single compressed global optimum after a short number of steps (one or two steps in most cases). For the larger values of α, the trajectories become longer and blue nodes (sub-optimal sinks) start to appear. We can observe one for $\alpha = 3.6$ and two for $\alpha = 5.0$, which are one value away in evaluation from the global optimum, moreover large intermediate (gray) nodes seem to be capturing many trajectories. This is consistent with the increasing search difficulty with larger α values; trajectories are longer, they can get trapped in sub-optimal plateaus and fail to reach the global optimum.

6 CONCLUSIONS

Industrial and pseudo-industrial SAT instances are very important in theory and in practice because many relevant problems in engineering and computer science can be easily encoded as SAT problems. Although random uniform SAT instances are more amenable to mathematical treatment and have been deeply studied, it appears that they are much harder to solve than real-world instances. This difference has been attributed to the fact that industrial instances tend to have highly nonrandom clauses. Among the various deviations from uniformity that have been suggested such as heterogeneity, locality, modularity and self-similarity, we have focused on heterogeneity in the present work. Heterogeneity refers to the variable distribution into clauses and it has been found empirically that in many real-world cases it can be modeled by power-law distributions. Our approach is based on an analysis of fitness landscapes through the concept of local optima networks. Using a suitable power-law SAT instance generator, we built the LONs and CLONs of the corresponding instances and computed a number of network metrics. We have shown that the chosen metrics correlate well with the performance of an iterated local



(a) $n = 500, \alpha = 2.0$, **(b)** $n = 500, \alpha = 3.6$ **(c)** $n = 500, \alpha = 5.0$

(d) $n = 50, \alpha = 2.0$, **(e)** $n = 50, \alpha = 3.6$ **(f)** $n = 50, \alpha = 5.0$

Figure 4: CLONs for the representative instances with $n = 500$ (top), $n = 50$ (bottom) $\beta = 4.0$ and density α and as indicated in the sub-captions. The color of nodes reflects different types of local optima is as indicated in the legend in the right plots. The size of nodes is proportional to the number of unique optima aggregated in the compressed nodes. The labels in the blue nodes indicate their difference in evaluation from the global optimum.

search. We also appreciate a shift in the metrics compared to both the problem hardness, measured as the proportion of SAT instances, and the performance of WalkSAT. We conjecture that this shift can be related to the ILS sampling process, which depends not only on the SAT probability, but also on the landscape features. We defer to future work the detailed explanation of the reasons that produce this shift in the LON metrics. It would also be useful to analyse the correlation between the properties of the CLON and the difficulty found by other algorithms. This could provide contributions in the field of algorithm selection and could be used to enhance current SAT solvers.

ACKNOWLEDGEMENTS

This research is partially funded by Universidad de Málaga, Consejería de Economía y Conocimiento de la Junta de Andaluía and FEDER under grant number UMA18-FEDERJA-003 (PRECOG); Ministerio de Ciencia, Innovación y Universidades and FEDER under contracts RTC-2017-6714-5 (Eco-IoT) and RED2018-102472-T (SEBASENet 2.0). The authors thank Andrew M. Sutton for providing technical details about the results in [4].

REFERENCES

[1] Carlos Ansótegui, Maria Luisa Bonet, and Jordi Levy. 2009. On the structure of industrial SAT instances. In *International Conference on Principles and Practice of Constraint Programming*. Springer, 127–141.
[2] Carlos Ansótegui, Maria Luisa Bonet, and Jordi Levy. 2009. Towards Industrial-Like Random SAT Instances. In *IJCAI*, Vol. 9. 387–392.
[3] Fahiem Bacchus. 2020. *MaxHS in the 2020 MaxSat Evaluation (Report B-2020-2.)*. Technical Report. 19–20 pages. http://hdl.handle.net/10138/318451
[4] T. Bläsius, T. Friedrich, and A. M. Sutton. 2019. On the Empirical Time Complexity of Scale-Free 3-SAT at the Phase Transition. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 117–134.
[5] James M Crawford and Larry D Auton. 1996. Experimental results on the crossover point in random 3-SAT. *Artificial intelligence* 81, 1-2 (1996), 31–57.
[6] G. Csardi and T. Nepusz. 2006. The igraph software package for complex network research. *InterJournal* Complex Systems (2006), 1695.
[7] Sebastian Herrmann, Gabriela Ochoa, and Franz Rothlauf. 2017. PageRank centrality for performance prediction: the impact of the local optima network model. *Journal of Heuristics* (12 May 2017).

[8] Scott Kirkpatrick and Bart Selman. 1994. Critical behavior in the satisfiability of random boolean expressions. *Science* 264, 5163 (1994), 1297–1301.
[9] Olivier C Martin, Rémi Monasson, and Riccardo Zecchina. 2001. Statistical mechanics methods and phase transitions in optimization problems. *Theoretical computer science* 265, 1-2 (2001), 3–67.
[10] David McAllester, Bart Selman, and Henry Kautz. 1997. Evidence for invariants in local search. In *AAAI/IAAI*. Rhode Island, USA, 321–326.
[11] Zongxu Mu and Holger H Hoos. 2015. On the empirical time complexity of random 3-SAT at the phase transition. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
[12] Gabriela Ochoa, Francisco Chicano, and Marco Tomassini. 2020. Global Landscape Structure and the Random MAX-SAT Phase Transition. In *International Conference on Parallel Problem Solving from Nature*. Springer, 125–138.
[13] G. Ochoa, M. Tomassini, S. Verel, and C. Darabos. 2008. A Study of NK Landscapes' Basins and Local Optima Networks. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation* (Atlanta, GA, USA) *(GECCO '08)*. ACM, New York, NY, USA, 555–562. https://doi.org/10.1145/1389095.1389204
[14] Gabriela Ochoa, Nadarajen Veerapen, Fabio Daolio, and Marco Tomassini. 2017. Understanding Phase Transitions with Local Optima Networks: Number Partitioning as a Case Study. In *Evolutionary Computation in Combinatorial Optimization, (EVOCOP) (LNCS, Vol. 10197)*. Springer, 233–248.
[15] Gabriela Ochoa, Nadarajen Veerapen, Fabio Daolio, and Marco Tomassini. 2017. Understanding Phase Transitions with Local Optima Networks: Number Partitioning as a Case Study. In *European on Conference Evolutionary Computation in Combinatorial Optimization (EvoCOP) (Lecture Notes in Computer Science, Vol. 10197)*. 233–248.
[16] Christos. H. Papadimitriou. 1994. *Computational complexity*. Addison-Wesley.
[17] Bart Selman and Scott Kirkpatrick. 1996. Critical behavior in the computational cost of satisfiability testing. *Artificial Intelligence* 81, 1-2 (1996), 273–295.
[18] Bart Selman, David G Mitchell, and Hector J Levesque. 1996. Generating hard satisfiability problems. *Artificial intelligence* 81, 1-2 (1996), 17–29.
[19] Carsten Sinz. 2007. Visualizing SAT instances and runs of the DPLL algorithm. *Journal of Automated Reasoning* 39, 2 (2007), 219–243.
[20] Peter F. Stadler. 2002. Fitness Landscapes. *Appl. Math. and Comput* 117 (2002), 187–207.