

# Estimating the Density of States of Boolean Satisfiability Problems on Classical and Quantum Computing Platforms

**Tuhin Sahai, Anurag Mishra**  
United Technologies Research Center  
Berkeley, CA

**Jose Miguel Pasini**  
United Technologies Research Center  
Hartford, CT

**Susmit Jha**  
Computer Science Laboratory  
SRI International

## Abstract

Given a Boolean formula  $\phi(x)$  in conjunctive normal form (CNF), the density of states counts the number of variable assignments that violate exactly  $e$  clauses, for all values of  $e$ . Thus, the density of states is a histogram of the number of unsatisfied clauses over all possible assignments. This computation generalizes both maximum-satisfiability (MAX-SAT) and model counting problems and not only provides insight into the entire solution space, but also yields a measure for the *hardness* of the problem instance. Consequently, in real-world scenarios, this problem is typically infeasible even when using state-of-the-art algorithms. While finding an exact answer to this problem is a computationally intensive task, we propose a novel approach for estimating density of states based on the concentration of measure inequalities. The methodology results in a quadratic unconstrained binary optimization (QUBO), which is particularly amenable to quantum annealing-based solutions. We present the overall approach and compare results from the D-Wave quantum annealer against the best-known classical algorithms such as the Hamze-de Freitas-Selby (HFS) algorithm and satisfiability modulo theory (SMT) solvers.

## 1 Introduction

The density of states (DOS) for a given propositional logic (Boolean) formula not only provides insight into the complete solution space but also serves as an accurate measure of the difficulty or hardness of the problem instance. The ability to compute DOS of Boolean formula has critical applications in system-requirements engineering of complex aerospace products. It provides a metric for requirements engineers to compare constraints, prescribed requirements (Ferrante et al. 2016), and requirements decompositions (Kirkman 1998). This computation is particularly germane to the design and optimization of complex aerospace systems (Sommerville 2005). Current classical methods for computing density of states (Wang and Landau 2001; Ermon, Gomes, and Selman 2010; 2011) have limited scalability. While the focus of this paper is on Boolean formulae, we note that constrained programming and feasibility prob-

lems can be easily mapped to equivalent Boolean satisfiability instances (Walsh 2000; Tamura et al. 2009).

The DOS problem in the standard  $k$ -satisfiability ( $k$ -SAT) setting can be elucidated as follows: instead of deciding whether a given logical formula is satisfiable or not, one aims to compute the *entire histogram of the number of clauses satisfied over all possible variable assignments*. Note the DOS, for a given instance of an optimization or decision problem, captures its hardness (distributions with a low footprint for all satisfied clauses are harder to compute or satisfy). The DOS histogram sheds light on the fundamental nature of the feasible solution set and difficulty of solving the optimization problem. Such problems frequently arise when constructing complex systems for aerospace and defense applications (Leveson and Weiss 2009).

The lack of methodical approaches that enable the comparison of competing safety-critical system requirements, while optimizing performance, stymie the development of next-generation complex systems. Note there are often multiple paths to decompose the overall system safety requirements down to subsystems requirements. Some of these decompositions may lead to costly design and redesign cycles to achieve desired levels of performance. Decompositions that have a higher DOS in the satisfiable range result in greater freedom to optimize performance and, consequently, result in quicker design cycles and fewer redesigns. The ability to quickly estimate the DOS of satisfiability problems will enable the specification engineer to ensure the prescribed requirements are satisfiable, internally consistent, and amenable to design space exploration very early in the design requirement step.

In this work, we aim to construct novel approaches for rapidly computing the DOS for a SAT problem (Boolean formula) (Biere, Heule, and van Maaren 2009). Our approximate approach to estimate DOS of SAT instances exploits the concentration of measure inequalities (Boucheron, Lugosi, and Massart 2013). These inequalities provide bounds on the tails of the distributions of random functions and have been used to construct the theory of generalization in machine learning (Abu-Mostafa, Magdon-Ismael, and Lin 2012) compute optimal bounds on uncertainty (Owhadi et al. 2013), compute bias of statistical estimators (Gourgoulis

et al. 2017), and derive results in random matrix theory (Tao 2012).

In this paper, we make the following contributions:

1. We introduce a novel approach to estimate the density of states for SAT problems by using concentration of measures (McDiarmid’s inequality), and bound the deviation of the number of unsatisfied clauses (energy) from the expected (mean) number of unsatisfied clauses for uniformly distributed assignments.
2. The deviation of the energy function from its expected value depends on its diameter (function variability), which can be computed by solving an optimization (maximization) problem (Owhadi et al. 2013). We show this maximum deviation computation can be posed in the form of a quadratic unconstrained binary optimization (QUBO) that is particularly amenable to quantum annealers and results in tight bounds on the DOS histogram.
3. We demonstrate our approach on classical platforms by computing the diameter and associated concentration of measure bounds using Selby’s implementation (Selby 2013; 2014) of the Hamze-de Freitas- Selby (HFS) algorithm (Hamze and de Freitas 2004).
4. We use satisfiability modulo theory (SMT) solvers (Barrett et al. 2009) to solve the QUBO formulation as an alternative approach on classical platforms. The solutions from SMT solvers provide tighter estimates but require significantly higher computational effort and do not scale.
5. We then compare the classical results to the computations on the D-Wave quantum annealer, a commercially available noisy intermediate-scale quantum (NISQ) device (Preskill 2018). We find the D-Wave machine provides higher-quality solutions when compared to the HFS algorithm, and scales better than SMT solvers. We further note the search for useful problems that are appropriate for present day NISQ devices is a very active area of research within quantum computing (Preskill 2018). We propose the DOS computation task as a potential test problem that can be used to *benchmark current- and next-generation quantum annealers against their classical counterparts*.

## 2 Background

Logical specification is widely used in the design and verification of hardware and software systems (Malik et al. 1988; Jha et al. 2010) as well as interpretable abstraction of machine learning models (Vazquez-Chanlatte et al. 2018; Jha et al. 2018). In this paper, we focus on propositional logic (Boolean) formula. A  $k$ -SAT Boolean formula  $\phi(x)$  of  $N$  Boolean variables and  $m$  clauses,  $\phi : \{0, 1\}^N \rightarrow \{0, 1\}$ , written in the conjunctive normal form (CNF) (Biere, Heule, and van Maaren 2009) as follows,

$$\phi(x) = \bigwedge_{i=1}^m C_i = \bigwedge_{i=1}^m (x_{i_1} \vee x_{i_2} \vee \dots \vee x_{i_k}), \quad (1)$$

where  $x_{i_l}$  is the  $l^{\text{th}}$  literal in clause  $C_i$ . A SAT formula is said to be satisfiable if there exists an assignment for the binary variables  $\mathbf{x}$  such that  $\phi(\mathbf{x}) = 1$  (true). It is well known

that the satisfiability problem is NP-complete (Cook 1971). A critical parameter associated with the satisfiability problem is the clause density  $\alpha = m/N$  (Biere, Heule, and van Maaren 2009). In particular, the probability that a random  $k$ -SAT instance is satisfiable undergoes a phase transition as a function of  $\alpha$  ( $N \rightarrow \infty$ ) (Xu and Li 2000; Biere, Heule, and van Maaren 2009). The MAX-SAT problem (and the corresponding weighted version) (Krentel 1988; Chieu and Lee 2009) requires one to find that assignment (or assignments) that maximize the number (or the cumulative weights) of satisfied clauses. Consider a SAT formula  $\phi$ , then every assignment  $x$  can be mapped to an “energy”  $\Phi(x)$  such that,

$$\Phi(\mathbf{x}) = \sum_{i=1}^m C_i, \quad (2)$$

where  $C_i = 1$ , if the  $i$ -th clause evaluates to true. In other words, the goal under the MAX-SAT problem is to find the assignment for  $\mathbf{x}$  such that the number of satisfied clauses (or energy) is maximized. Using De Morgan’s laws, one can easily show that,

$$\Phi(\mathbf{x}) = m - \sum_{i=1}^m \prod_{l=1}^k f(x_{i_l}), \quad (3)$$

$$\text{where } f(x_{i_l}) = \begin{cases} x_{i_l}, & \text{if } x_{i_l} \text{ is negated in the clause} \\ (1 - x_{i_l}), & \text{otherwise.} \end{cases} \quad (4)$$

Using the above formula, it is easy to see that if each literal  $x_{i_l}$  were random with equal probability for values  $\{0, 1\}$ , then the expected number of satisfied clauses is,

$$\mathbb{E}[\Phi(\mathbf{x})] = \frac{m(2^k - 1)}{2^k}. \quad (5)$$

Thus, even though the satisfiability is NP-complete, a random assignment is expected to satisfy a large fraction of the clauses. For the 3-SAT, the above formula reduces to,

$$\mathbb{E}[\Phi(\mathbf{x})] = 7m/8. \quad (6)$$

One can use this expected value of the number of satisfied clauses to estimate the DOS using concentration of measure inequalities. For SAT instances that arise from specific application domains (thus, not random), one can estimate the expected number of satisfied clauses by sampling over the independent variables in the Boolean formula.

The DOS  $d(e)$  of a SAT formula  $\phi$  is equal to the number of assignments  $\mathbf{x}$  for which  $\Phi(\mathbf{x}) = e$ . In other words, it is the histogram of the number assignments as a function of  $e$  satisfied clauses. Note the value of the number of satisfied clauses  $e$  lies between 0 and  $m$  where  $m$  is the total number of clauses in the SAT formula. Following the terminology from the physics community, we will also call this the energy of SAT formula. Since the total number of possible assignments is  $2^N$ , one can define the normalized density of states as follows,

$$p(e) = \frac{d(e)}{2^N}. \quad (7)$$

The normalized DOS acts as a discrete probability distribution. Note it is not necessary that all energies  $e$  have a valid assignment. For example, if the SAT formula cannot be satisfied then  $p(m) = 0$ . As explained previously, for a random 3-SAT, the mean of  $p(e)$  vs  $e$  is at  $7m/8$  (Ermon, Gomes, and Selman 2010).

The density computation problem generalizes computationally hard problems of MAX-SAT and model counting (Birnbaum and Lozinskii 1999). The state-of-the-art algorithm for computing DOS is inspired by the Wang and Landau random walk algorithm (Wang and Landau 2001). In (Ermon, Gomes, and Selman 2010; 2011), the authors propose an adaptive Markov Chain Monte Carlo (MCMC) approach called MCMC-FlatSAT that aims to sample from a steady-state distribution such that probability of a particular assignment  $\sigma$  is inversely proportional to its DOS. Thus, the sampling approach effectively converges to a flat-visit histogram (captured by a flatness parameter). In (Ermon, Gomes, and Selman 2010; 2011), the authors test the algorithm on multiple benchmark examples. We use this method to compute the DOS for a series of random SAT instances and compare the resulting histograms to estimated DOS using our concentration of measure-based approach.

Fig. 1 describes a simple example of the DOS problem for a Boolean satisfiability problem with  $N = 100$  and  $\alpha = 4.0$ , and shows an example output of the MCMC-FlatSAT algorithm. In our experiments, we found that for  $k$ -SAT instances close to the phase transition (Monasson et al. 1999), the mixing times of the Markov chain (Levin and Peres 2017) increase significantly.

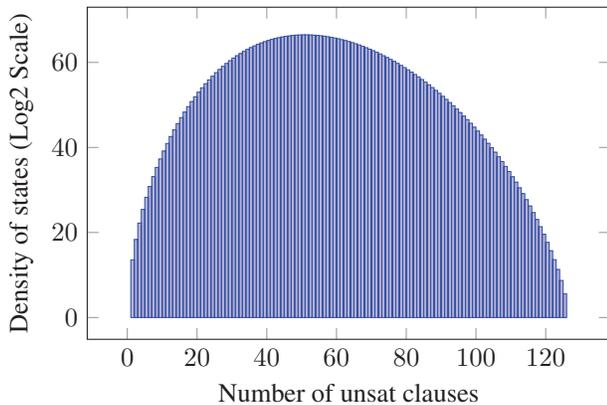


Figure 1: This example SAT problem has  $N = 100$  variables, and the clause density ratio  $\alpha$  is 4.0. The x-axis is the number of UNSAT clauses and the y-axis provides a numerical value for the number of assignments with the corresponding number of UNSAT clauses. The DOS over all possible variable assignments are captured in the histogram.

### 3 SAT and Concentration of Measure Inequalities

The concentration of measure phenomena bounds the deviation of functions of random variables around their

mean (Boucheron, Lugosi, and Massart 2013). There are a host of inequalities associated with various situations, see (Boucheron, Lugosi, and Massart 2013; Tao 2010) for more details. For our setting, we use McDiarmid’s inequality (McDiarmid 1989) summarized in the theorem below.

**Theorem 1. (McDiarmid’s inequality)** Let  $x_1, x_2, \dots, x_N$  be independent random variables taking values in the range  $R_1, R_2, \dots, R_N$  and let  $F : R_1 \times R_2 \times \dots \times R_N \rightarrow \mathbb{R}$  be a function with the property that if one freezes all but the  $i$ -th variable, then  $F(x_1, x_2, \dots, x_N)$  fluctuates by at most  $D_i > 0$ ,

$$D_i = \sup_{x_1, x_2, \dots, x_i, \hat{x}_i, \dots, x_N} |F(x_1, x_2, \dots, x_i, \dots, x_N) - F(x_1, x_2, \dots, \hat{x}_i, \dots, x_N)|. \quad (8)$$

Then the probability that  $F$  deviates from its expected value is given by,

$$\mathbb{P}[|F(x_1, x_2, \dots, x_N) - \mathbb{E}[F(x_1, x_2, \dots, x_N)]| \geq \epsilon] \leq C \exp(-c \frac{2\epsilon^2}{D^2}), \quad (9)$$

where  $D = \sqrt{\sum_i D_i^2}$  is called the diameter and  $C, c$  are constants.

*Proof.* See (Boucheron, Lugosi, and Massart 2013; Tao 2010).  $\square$

So instead of computing the DOS using the MCMC approach outlined in section 2, one can exploit McDiarmid’s inequality to compute bounds on the histogram of number of satisfied (or unsatisfied) clauses. In the setting of the  $k$ -SAT problem, the  $x_1, x_2, \dots, x_N$  in the McDiarmid’s inequality are replaced by the variables  $\mathbf{x}$  present in the logical formula. That is,  $x_1, x_2, \dots, x_N$  are the unique set of Boolean variables that occur in the formula. The MCMC computation is now replaced by the set of optimization problems for computing the diameter as shown in Eqn. 8. Note that for the density of states computation, the variables are independent (since we are searching over all possible assignments).

For ease of presentation, we focus on the 3-SAT problem instead of the generic  $k$ -SAT formulation. Polynomial time reductions from  $k$ -SAT to 3-SAT make this translation non-restrictive. Every  $k$ -SAT instance can be converted to a 3-SAT instance by introducing additional (ancillary) variables. We now show that the diameter computations for the 3-SAT problem give rise to a QUBO (Boros, Hammer, and Tavares 2007; Rieffel and Polak 2011) problem that is particularly amenable to quantum annealers (Kochenberger et al. 2014).

### 4 QUBO formulation for diameter computation

To estimate  $D_i$  in Eqn. 8 for the 3-SAT setting, consider the following form for the SAT formula,

$$\phi(\mathbf{x}) = \bigwedge_{i=1}^m C_i = \bigwedge_{i=1}^m (x_{i_1} \vee x_{i_2} \vee x_{i_3}),$$

where  $x_{i_l}$  is the  $l$ -th literal in clause  $C_i$ . It is easy to check that the number of satisfied clauses can be expressed as:

$$\Phi(\mathbf{x}) = \sum_{i=1}^m C_i = \sum_{i=1}^m (x_{i_1} + x_{i_2} + x_{i_3} - x_{i_1}x_{i_2} - x_{i_1}x_{i_3} - x_{i_2}x_{i_3} + x_{i_1}x_{i_2}x_{i_3}). \quad (10)$$

While this expression has a cubic term, the cubic term disappears when computing the diameter in Eqn. (8) (shown later). We now state our central result that formulates the estimation of diameters  $D_i$  needed for computing the DOS as a quadratic unconstrained Boolean optimization problem.

**Theorem 2.** *The diameter  $D_i$  for the variable  $x_i$  in McDiarmid's inequality can be computed by solving the following optimization problem,*

$$D_i = \max_{\mathbf{x} \setminus \mathbf{x}(i)} \left| \sum_{p \in S_i^+} [1 - x_{p_2} - x_{p_3} + x_{p_2}x_{p_3}] - \sum_{p \in S_i^-} [1 - x_{p_2} - x_{p_3} + x_{p_2}x_{p_3}] \right|,$$

where  $S_i^+$  and  $S_i^-$  are the sets of clauses in which  $x(i)$  appears in direct and negated forms, respectively.

*Proof.* To compute the diameter  $D_i$ , pick the  $i$ -th variable of  $\mathbf{x}$  denoted as  $\mathbf{x}(i)$  and compute the worst-case variation in the number of satisfied clauses. Since  $\Phi(\mathbf{x})$  is a sum over different clauses, only those clauses that include  $\mathbf{x}(i)$  in their literal set will contribute to the diameter.  $S_i^+$  and  $S_i^-$  are the sets of clauses in which  $x(i)$  appears in direct and negated forms respectively, that is,

$$S_i^+ = \{p : C_p = \mathbf{x}(i) \vee x_{p_2} \vee x_{p_3}\},$$

$$S_i^- = \{p : C_p = \neg \mathbf{x}(i) \vee x_{p_2} \vee x_{p_3}\},$$

are the set of clauses in which the variable  $\mathbf{x}(i)$  appears in the corresponding literal sets as either  $\mathbf{x}(i)$  or  $\neg \mathbf{x}(i)$ , respectively. Furthermore, because  $\vee$  is commutative, we can assume without loss of generality that the variable  $\mathbf{x}(i)$  appears as the first literal of the clause. Therefore, the expression for the number of satisfied clauses in the 3-SAT instance is:

$$\begin{aligned} \forall p \in S_i^+ \quad \Phi_p &= \mathbf{x}(i) + x_{p_2} + x_{p_3} \\ &\quad - \mathbf{x}(i)x_{p_2} - \mathbf{x}(i)x_{p_3} - x_{p_2}x_{p_3} + \mathbf{x}(i)x_{p_2}x_{p_3}, \\ \forall p \in S_i^- \quad \Phi_p &= (1 - \mathbf{x}(i)) + x_{p_2} + x_{p_3} \\ &\quad - (1 - \mathbf{x}(i))x_{p_2} - (1 - \mathbf{x}(i))x_{p_3} \\ &\quad - x_{p_2}x_{p_3} + (1 - \mathbf{x}(i))x_{p_2}x_{p_3}. \end{aligned} \quad (11)$$

Let  $S_i^0 = \{1, \dots, m\} \setminus (S_i^+ \cup S_i^-)$  be the set of clauses within which  $\mathbf{x}(i)$  does not occur, thus,

$$\Phi(x) = \sum_{p \in S_i^0} \Phi_p + \sum_{p \in S_i^+} \Phi_p + \sum_{p \in S_i^-} \Phi_p.$$

The first term in the above sum is not affected by changing  $\mathbf{x}(i)$  and does not contribute to the diameter and cancels in the subtraction in Eqn. 8. Now, since  $\mathbf{x}(i)$  can only take one of two values,  $\{0, 1\}$ , the number of clauses satisfied by setting  $\mathbf{x}(i)$  to 1 in  $S_i^+$  is  $\sum_{p \in S_i^+} [1]$ , and in  $S_i^-$  is

$\sum_{p \in S_i^-} x_{p_2} + x_{p_3} - x_{p_2}x_{p_3}$  (computed using Eqn. 11). Symmetrically, the number of clauses satisfied by setting  $\mathbf{x}(i)$  to 0 in  $S_i^+$  is  $\sum_{p \in S_i^+} [1]$ , and in  $S_i^-$  is  $\sum_{p \in S_i^-} x_{p_2} + x_{p_3} - x_{p_2}x_{p_3}$ .

$D_i$  is the maximum deviation between the two, that is,

$$D_i = \max_{\mathbf{x} \setminus \mathbf{x}(i)} \left| \sum_{p \in S_i^+} [1] + \sum_{p \in S_i^-} [x_{p_2} + x_{p_3} - x_{p_2}x_{p_3}] - \sum_{p \in S_i^+} [x_{p_2} + x_{p_3} - x_{p_2}x_{p_3}] - \sum_{p \in S_i^-} [1] \right|.$$

We get the following optimization problem to compute  $D_i$  by collecting the terms for summing over  $S_i^+$  and  $S_i^-$ ,

$$D_i = \max_{\mathbf{x} \setminus \mathbf{x}(i)} \left| \sum_{p \in S_i^+} [1 - x_{p_2} - x_{p_3} + x_{p_2}x_{p_3}] - \sum_{p \in S_i^-} [1 - x_{p_2} - x_{p_3} + x_{p_2}x_{p_3}] \right|. \quad (12)$$

□

This result makes sense intuitively, because the expression inside each bracket is logically equivalent to  $\neg(x_{p_2} \wedge x_{p_3})$ , and if either of the other literals is true, the disjunctive clause  $C_p$  remains false regardless of  $\mathbf{x}(i)$ , and therefore does not contribute to the diameter.

The expression inside the absolute value in (12) is a quadratic form in  $\mathbf{x} \setminus \mathbf{x}(i)$ . Note that Eqn. 12 can easily be cast into a purely quadratic form  $\mathbf{x}^T Q \mathbf{x}$  as the linear terms can be absorbed into the diagonal of the matrix because  $x_p = x_p^2$  for binary variables.

**Remark 1.** *The computation for  $D_i$  involves the maximization of an absolute value. To address the absolute value, we simply perform two separate maximizations as follows  $\sup_{\mathbf{x}} |f(\mathbf{x})| = \max\{\sup_{\mathbf{x}} f(\mathbf{x}), -\sup_{\mathbf{x}} f(\mathbf{x})\}$ . Thus, we compute the two maximizations and choose the larger result to obtain the diameter. Note that for a 3-SAT instance with  $N$  unique variables, one needs to perform  $2N$  optimizations.*

**Remark 2.** *Besides providing a novel approach for estimating the DOS of  $k$ -SAT problems, the diameter computation can be used to benchmark optimization algorithms and computing platforms. In particular, by comparing the value of the computed diameter by different approaches, one can quantify their performance. Higher diameter values correspond to "better" solutions of the optimization problem.*

**Remark 3.** Using the density of states, one can extract the probability of a random assignment being at least  $\epsilon$  (in terms of energy or number of clauses satisfied) away from the average value  $\bar{E} = \sum_{e=0}^m p(e)e$ . Thus,

$$P[|E - \bar{E}| \geq \epsilon] = \sum_{\bar{E} + \epsilon}^m p(e). \quad (13)$$

This quantity can be computed numerically from  $d(e)$ .

In our experiments, we solved this QUBO formulation using quantum and classical computing methods. We describe these in detail in the next section.

## 5 Results

In our experiments, we attempt to answer the following questions.

- How does the proposed DOS approach of using concentration of measures and QUBO compare with the baseline MCMC-FlatSAT approach (Ermon, Gomes, and Selman 2010; 2011)?
- How do the quantum and classical implementations of the proposed DOS approach compare with one another?

To analyze the performance of the proposed approach, we generate 20 random 3-SAT instances for every possible combination of the following sizes ( $N = [30, 50, 75, 100, 125, 150, 200]$ ) and clause density ( $\alpha = [4.0, 4.25, 4.5, 5]$ ). As mentioned earlier, although the proposed technique can also be easily applied to non-random SAT instances, the choice of random SAT instances allows variation from easy-to-hard problems. Our choice of  $\alpha$  values span the phase transition at  $\alpha \approx 4.24$  that demarcates “easy” and “hard” instances of the satisfiability problem (Monasson et al. 1999). Thus, in total, we generate 560 random 3-SAT instances, and for each instance we compute the baseline DOS using MCMC-FlatSAT. These results are then compared with the proposed concentration of measure inequality approach. Additionally, the  $2N$  diameter computations for each instance are performed classically using the HFS algorithm and the D-Wave quantum annealer. The performance of the D-Wave device is then compared with the classical results. We also implemented an SMT-based optimization approach on classical platforms, and compared the D-Wave results with the standard classical solver.

### MCMC-FlatSAT results

We implemented the MCMC-FlatSAT (Ermon, Gomes, and Selman 2010; 2011) algorithm in C++. Depending on the mixing time of the Markov chain (Levin and Peres 2017), there was a large variation in the performance of the code. The computation time ranged from hours to several days (in some instances the code took 3 – 10 days to converge). The computations were performed for all of the 560 instances as outlined above. A few instances of the resulting density of states are shown in Fig. 2. In general, the MCMC approach was found to have a high computational cost.

### SMT results

We used the Z3 SMT solver (Björner, Phan, and Fleckenstein 2015) to encode the QUBO problem as a bitvector problem exploiting the fixed range of discrete values that can be taken by the diameters. The resulting problem is a pseudo-Boolean optimization problem that we solve iteratively using satisfiability solving by binary search (between 0 and the number of clauses in which the variable occurs) over the optimization goal. We allow the SMT solver a timeout of 100 seconds for every trial to find a larger diameter. The 560 instances took 8 days to compute. The SMT solver found better solutions for the QUBO compared to quantum annealing, and hence placed more accurate bounds on the histogram. However, the scalability declined sharply with the increase in the number of variables. In particular, we found that the Z3 solver took hours to days to complete several instances. The diameters computed using the SMT solver can be seen in Fig. 5 for a few values of  $N$  and  $\alpha$ .

### Quantum annealing results

We use the D-Wave 2X (DW2X) annealer (Johnson et al. 2011) located at the USC Information Science Institute in Marina del Rey as our quantum platform for computations. This DW2X processor is an 1152-qubit quantum annealing device made using superconducting flux qubits (Bunyk et al. 2014). It has 1098 functional qubits that function at 12 mK. The annealer implements the transverse Ising Hamiltonian,

$$H(s) = A(s) \sum_i \sigma_i^x + B(s) \left( \sum_i h_i \sigma_i^z + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z \right), \quad (14)$$

where  $s = t/t_f$  is the normalized time,  $t_f$  is the total evolution time, and  $A(s)$  and  $B(s)$  are the annealing schedules that modulate the transverse field and Ising field strength, respectively. The total annealing time  $t_f$  can be set in the range  $[5, 2000] \mu\text{s}$ . The coupling strengths  $J_{ij}$  between qubits  $i$  and  $j$  can be set in the range  $[-1, 1]$ , and the local fields  $h_i$  can be set in the range  $[-2, 2]$ . Initially,  $A(0) \gg B(0)$  and the system starts in the superposition of all possible computational states. During the evolution from  $s = 0$  to  $s = 1$ , the transverse field is reduced and the Ising field strength is increased such that  $A(1) \ll B(1)$ . If  $t_f$  is large enough, the adiabatic theorem (Born and Fock 1928) guarantees that the final state of the system will be the ground state of  $H(s = 1)$ . The device has been used for machine learning (Biamonte et al. 2017; Adachi and Henderson 2015; Mott et al. 2017), image recognition (Neven, Rose, and Macready 2008), and combinatorial optimization (Ushijima-Mwesigwa, Negre, and Mniszewski 2017; McGeoch and Wang 2013; Neukart et al. 2017; Venturelli, Marchand, and Rojo 2015) to name a few.

We use the above platform to compute the diameters for all the 560 instances of random satisfiability problems and compared the results to MCMC-FlatSAT. As noted in remark 1, each instance of an  $N$ -dimensional 3-SAT problem gives rise to  $2N$  optimizations for  $D_i$ . We chose the smallest possible annealing time  $t_f = 5 \mu\text{s}$ . For each QUBO instance of this study, we did 1000 readouts with 10 gauge transforms

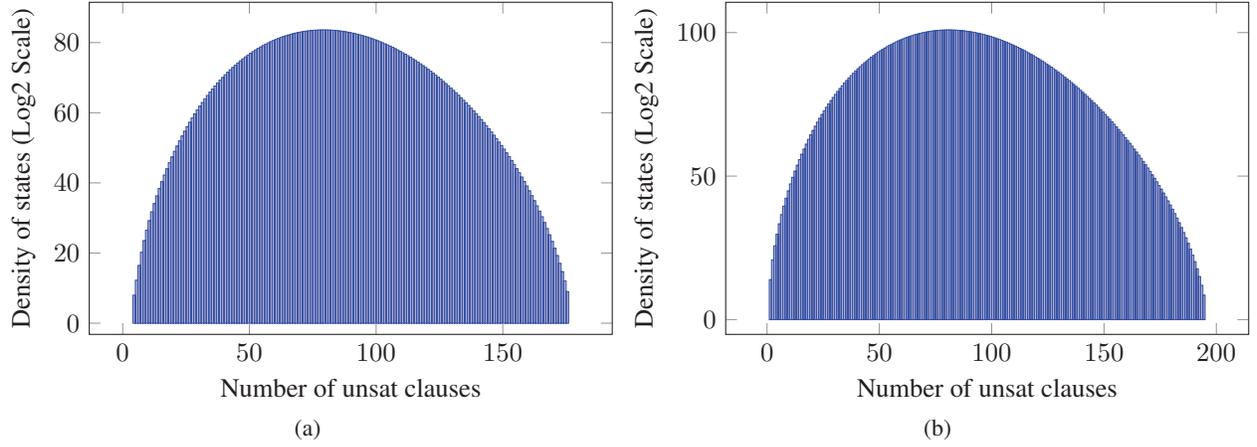


Figure 2: Density of states for random satisfiability instances with varying size and clause density. The X-axis is the number of unsat clauses, Y-axis is the DOS showing number of assignments in log scale. (a)  $N = 125$ ,  $\alpha = 5.0$ , (b)  $N = 150$ ,  $\alpha = 4.25$ .

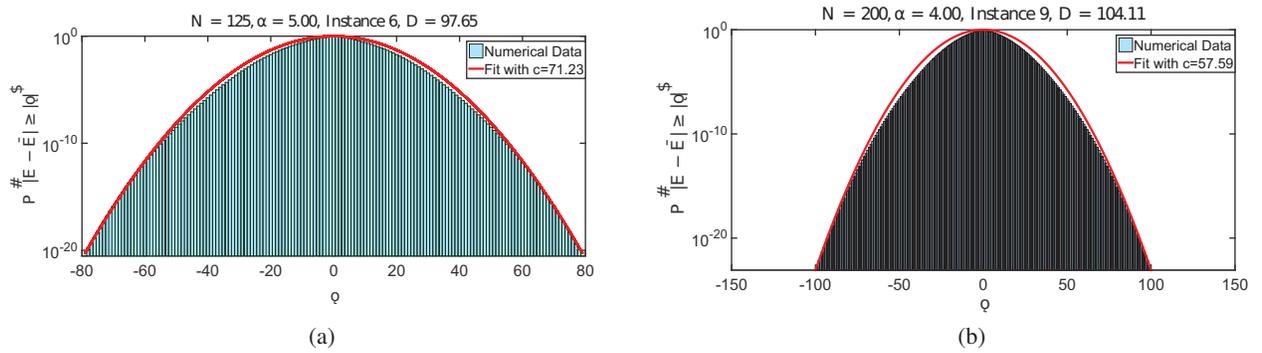


Figure 3: Comparison of the density of states computed by MCMC and the concentration of measure bounds.

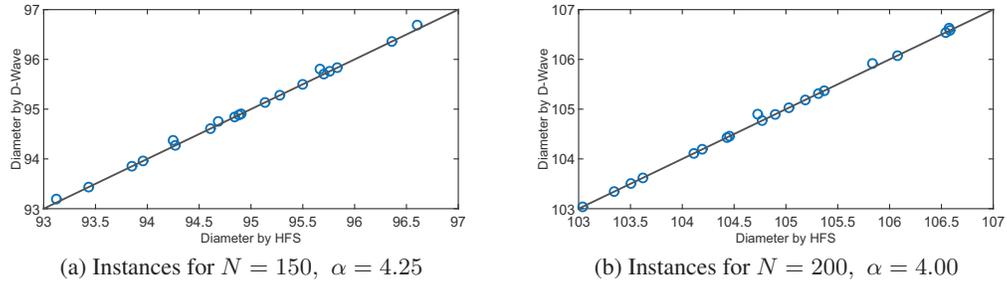


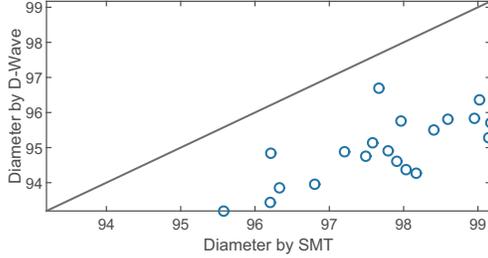
Figure 4: Comparison of diameters computed by the D-Wave quantum annealer and HFS algorithm.

each (Boixo et al. 2014). Additional details of this particular process can be found in, for example, Ref. (Mishra, Albash, and Lidar 2018). Note that the total wall clock time to optimize each instance, which includes overheads such as initializing the qubits and measurements, was  $\approx 0.1$  second. Additionally, 1000 readouts are on the low side; however, we were restricted due to the sheer number of QUBOs ( $\approx 120,000$  instances) coupled with limited affiliate time on

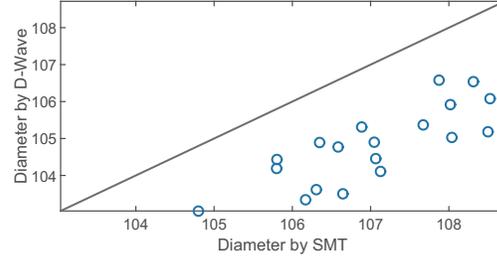
the DW2X annealer.

We map each diameter computation to a QUBO.

$$\begin{aligned}
 Q_i(\vec{x}) = & \sum_{p \in S_i^+} [1 - x_{p_2} - x_{p_3} + x_{p_2}x_{p_3}] \\
 & - \sum_{p \in S_i^-} [1 - x_{p_2} - x_{p_3} + x_{p_2}x_{p_3}].
 \end{aligned} \tag{15}$$



(a) Instances for  $N = 150$   $\alpha = 4.25$



(b) Instances for  $N = 200$   $\alpha = 4.00$

Figure 5: Comparison of diameters computed by the D-Wave quantum annealer and SMT solver.

The size of this QUBO problem depends on the size of the sets  $S_i^+$  and  $S_i^-$  (see Section 4 of the paper). If the SAT problem has  $N$  variables and  $\alpha$  clause density, the number of clauses  $M = N\alpha$  is a loose upper bound on the size of these QUBO problems. Since the clauses can contain arbitrary variables, for which DW2X has a finite connectivity graph, we need to find a minor embedding of the D-Wave graph that can fit this QUBO problem (Choi 2008; 2011). In such embedding, each  $x_{p_i}$  in Eqn. 15 is represented by a chain of physical qubits connected via an ferromagnetic couplings. We used the `sapiFindEmbedding` function provided by the D-Wave application program interfaces (API) to find such embeddings. We used the `sapiEmbedProblem` function to submit the jobs to the processor and `sapiUnembedAnswer` function with the `minimize_energy` option to optimally decode the embeddings back to the variables  $\vec{x}$ . We used the heuristic ferromagnetic chain coupling provided by the API. To find higher-quality solutions, one can optimize this ferromagnetic coupling value such that the chain of physical qubits representing each variable is consistent at the end of the anneal. Thus, our results provide a lower bound on the diameter. Potentially, one may be able to obtain improved results by performing the actions suggested above and optimizing the annealing process.

After computing all the  $D_i$ 's for a given instance, we can plot the concentration of measure bounds for the DOS. Note that in McDiarmid's inequality (Eqn. 9), two constants appear that can be used to make the bounds on the DOS tight. In particular, we find  $C = 1$  and  $c = 56.16 - 12.08 \exp(-0.07(N - 29.78)) + 6.88(\alpha - 4.46)$  give rise to very close approximations of the density of states in the range of  $30 \leq N \leq 200$  (as shown in Fig. 3). These parameters were computed using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (Liu and Nocedal 1989). The functional form for  $c$  was obtained empirically by finding the best  $c$  for each of the 560 instances and performing regression with respect to  $N$  and  $\alpha$  (see Fig. 6).

### Comparison of D-Wave and HFS

We repeat the  $D_i$  computations for each of the 560 instances by forcing the classical device to compute the best possible solution using the HFS algorithm. We again run the com-

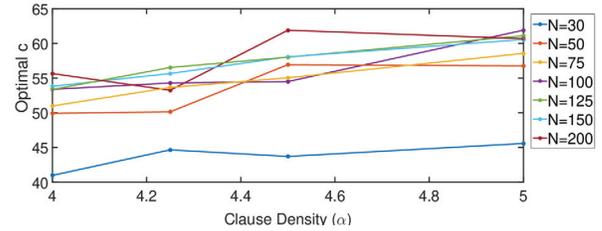


Figure 6: Concentration of measure constant as a function of  $\alpha$ . The different lines correspond to different values of  $N$ .

putation for 0.1 secs, repeated 1000 times for each  $D_i$ . The best  $D_i$  is saved and the rest are discarded. We then compare the  $D_i$  values obtained using quantum annealing with those computed classically. Note that higher diameter values correspond to "better" solutions as they correspond to higher quality solutions of the QUBO. Note we intend to conduct a comprehensive benchmarking study for the D-Wave quantum annealer (Albash and Lidar 2018) (using our DOS framework) in future work.

Out of the 560 random satisfiability instances, the D-Wave quantum computer computes higher quality solutions (higher diameter values) in 306 instances. However, as shown in Fig. 4, the D-Wave provided a marginal improvement on the diameter values. In particular, we found that the average solution computed by the D-Wave machine was around 0.7% higher than the HFS algorithm. The most favorable result for the D-Wave was the computation of a solution that was 7% better than the HFS algorithm. Whether this improvement holds for larger instances remains to be seen and will be tested in higher qubit settings. We would like to point out, however, that in no instance did the HFS algorithm find a higher quality solution when compared to the D-Wave machine. As shown in Fig. 5, the SMT solver does find significantly better solutions than the D-Wave machine. Note that the computational cost of the solver is significantly higher (taking hours to days to compute the diameter of some instances). More detailed results can be found in (Sahai et al. 2019).

## 6 Conclusion

We have developed a novel concentration of measure inequalities-based approach for estimating the density of states of the  $k$ -SAT problem. Existing state-of-the-art Markov Chain Monte Carlo methods for computing density of states are stymied by computational intractability. Our approach provides estimates for the density of states histogram by converting the problem into a set of optimization problems that bound the maximum variability of the cost function. For the 3-SAT, these optimization problems reduce to quadratic unconstrained binary optimizations, thereby making them amenable for commercially available quantum annealers such as the D-Wave machine.

In summary, we propose a new approach for estimating density of states of the  $k$ -SAT problem that can be implemented on both classical and quantum platforms. Moreover, the problem is a particularly interesting test for comparing quantum platforms (annealers and other noisy intermediate-scale quantum devices) to classical computation. This is because the diameter values provide a real number metric for comparison. In other words, the quality of solution is a real number as opposed to the standard satisfiability tests for benchmarking quantum devices that yield inconclusive results of the form “no satisfiable assignments found” for most instances. We hope this problem and the outlined approach can be used to analyze complex aerospace system requirements from a satisfiability standpoint as well as to test emerging quantum platforms against their classical counterparts. DOS estimation can be used for probabilistic inference and, thus, an efficient quantum algorithm for the density of states estimation will enable development of quantum artificial intelligence.

## 7 Acknowledgements

The authors thank Federico Spedalieri and Daniel Lidar of the University of Southern California-Information Sciences Institute (USC-ISI) for discussions and suggestions. The authors thank Lockheed Martin’s quantum computing team (Christopher Elliott, Greg Tallant, and Kristen Pudenz) for discussions related to the approach and generously providing affiliate access to their D-Wave quantum annealer. Dr. Jha acknowledges support from the US Army Research Laboratory Cooperative Research Agreement W911NF-17-2-0196, and National Science Foundation (NSF) grants #1750009, and #1740079. The views, opinions and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

## References

Abu-Mostafa, Y. S.; Magdon-Ismael, M.; and Lin, H.-T. 2012. *Learning from data*, volume 4. AMLBook New York, NY, USA.

Adachi, S. H., and Henderson, M. P. 2015. Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*.

Albash, T., and Lidar, D. A. 2018. Demonstration of a scaling advantage for a quantum annealer over simulated annealing. *Physical Review X* 8(3):031016.

Barrett, C.; Sebastiani, R.; Seshia, S. A.; Tinelli, C.; Biere, A.; Heule, M.; van Maaren, H.; and Walsh, T. 2009. Handbook of satisfiability. *Satisfiability modulo theories* 185:825–885.

Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; and Lloyd, S. 2017. Quantum machine learning. *Nature* 549(7671):195.

Biere, A.; Heule, M.; and van Maaren, H. 2009. *Handbook of Satisfiability*, volume 185. IOS press.

Birnbaum, E., and Lozinskii, E. L. 1999. The good old Davis-Putnam procedure helps counting models. *Journal of Artificial Intelligence Research* 10:457–477.

Bjørner, N.; Phan, A.-D.; and Fleckenstein, L. 2015. *vz*-an optimizing SMT solver. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 194–199. Springer.

Boixo, S.; Rønnow, T. F.; Isakov, S. V.; Wang, Z.; Wecker, D.; Lidar, D. A.; Martinis, J. M.; and Troyer, M. 2014. Evidence for quantum annealing with more than one hundred qubits. *Nat. Phys.* 10(3):218–224.

Born, M., and Fock, V. 1928. Beweis des Adiabatsatzes. *Zeitschrift für Physik* 51(3-4):165–180.

Boros, E.; Hammer, P. L.; and Tavares, G. 2007. Local search heuristics for quadratic unconstrained binary optimization (qubo). *Journal of Heuristics* 13(2):99–132.

Boucheron, S.; Lugosi, G.; and Massart, P. 2013. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press.

Bunyk, P. I.; Hoskinson, E. M.; Johnson, M. W.; Tolkacheva, E.; Altomare, F.; Berkley, A. J.; Harris, R.; Hilton, J. P.; Lanting, T.; Przybysz, A. J.; and Whittaker, J. 2014. Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE Trans. Appl. Supercond.* 24(4):1–10.

Chieu, H. L., and Lee, W. S. 2009. Relaxed survey propagation for the weighted maximum satisfiability problem. *Journal of Artificial Intelligence Research* 36:229–266.

Choi, V. 2008. Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quantum Inf. Process.* 7(5):193–209.

Choi, V. 2011. Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design. *Quantum Inf. Process.* 10(3):343–353.

Cook, S. A. 1971. The complexity of theorem-proving procedures. In *ACM Symposium on Theory of Computing*, 151–158. ACM.

Ermon, S.; Gomes, C. P.; and Selman, B. 2010. Computing the density of states of Boolean formulas. In *International Conference on Principles and Practice of Constraint Programming*, 38–52. Springer.

Ermon, S.; Gomes, C.; and Selman, B. 2011. A flat histogram method for computing the density of states of combinatorial problems. In *International Joint Conference on Artificial Intelligence*.

Ferrante, O.; Scholte, E.; Pinello, C.; Ferrari, A.; Mangeruca, L.; and Liu, C. 2016. A methodology for increasing the efficiency and coverage of model checking and its application to aerospace systems. *International Journal of Aerospace* 9(2016-01-2053):140–150.

Gourgoulias, K.; Katsoulakis, M. A.; Rey-Bellet, L.; and Wang, J. 2017. How biased is your model? Concentration inequalities, information and model bias. *arXiv preprint arXiv:1706.10260*.

Hamze, F., and de Freitas, N. 2004. From fields to trees. In *Uncertainty in Artificial Intelligence*, 243–250. AUAI Press.

- Jha, S.; Gulwani, S.; Seshia, S. A.; and Tiwari, A. 2010. Oracle-guided component-based program synthesis. In *ICSE*, 215–224. IEEE.
- Jha, S.; Sahai, T.; Raman, V.; Pinto, A.; and Francis, M. 2018. Explaining ai decisions using efficient methods for learning sparse boolean formulae. *Journal of Automated Reasoning* 1–21.
- Johnson, M. W.; Amin, M. H.; Gildert, S.; Lanting, T.; Hamze, F.; Dickson, N.; Harris, R.; Berkley, A. J.; Johansson, J.; Bunyk, P.; et al. 2011. Quantum annealing with manufactured spins. *Nature* 473(7346):194.
- Kirkman, D. 1998. Requirement decomposition and traceability. *Requirements Engineering* 3(2):107.
- Kochenberger, G.; Hao, J.-K.; Glover, F.; Lewis, M.; Lü, Z.; Wang, H.; and Wang, Y. 2014. The unconstrained binary quadratic programming problem: a survey. *Journal of Combinatorial Optimization* 28(1):58–81.
- Krentel, M. W. 1988. The complexity of optimization problems. *Journal of Computer and System Sciences* 36(3):490–509.
- Leveson, N. G., and Weiss, K. A. 2009. Software system safety. In *Safety Design for Space Systems*. Elsevier. 475–505.
- Levin, D. A., and Peres, Y. 2017. *Markov chains and mixing times*, volume 107. American Mathematical Soc.
- Liu, D. C., and Nocedal, J. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming* 45(1-3):503–528.
- Malik, S.; Wang, A. R.; Brayton, R. K.; and Sangiovanni-Vincentelli, A. 1988. Logic verification using binary decision diagrams in a logic synthesis environment. In *[1988] IEEE International Conference on Computer-Aided Design (ICCAD-89) Digest of Technical Papers*, 6–9. IEEE.
- McDiarmid, C. 1989. On the method of bounded differences. *Surveys in Combinatorics* 141(1):148–188.
- McGeoch, C. C., and Wang, C. 2013. Experimental evaluation of an adiabatic quantum system for combinatorial optimization. In *ACM International Conference on Computing Frontiers*, 23. ACM.
- Mishra, A.; Albash, T.; and Lidar, D. A. 2018. Finite temperature quantum annealing solving exponentially small gap problem with non-monotonic success probability. *Nat. Commun.* 9(1):2917.
- Monasson, R.; Zecchina, R.; Kirkpatrick, S.; Selman, B.; and Troyansky, L. 1999. Determining computational complexity from characteristic ‘phase transitions’. *Nature* 400(6740):133.
- Mott, A.; Job, J.; Vlimant, J.-R.; Lidar, D.; and Spiropulu, M. 2017. Solving a Higgs optimization problem with quantum annealing for machine learning. *Nature* 550(7676):375.
- Neukart, F.; Compostella, G.; Seidel, C.; Von Dollen, D.; Yarkoni, S.; and Parney, B. 2017. Traffic flow optimization using a quantum annealer. *Frontiers in ICT* 4:29.
- Neven, H.; Rose, G.; and Mcready, W. G. 2008. Image recognition with an adiabatic quantum computer I. Mapping to quadratic unconstrained binary optimization. *arXiv preprint arXiv:0804.4457*.
- Owhadi, H.; Scovel, C.; Sullivan, T. J.; McKerns, M.; and Ortiz, M. 2013. Optimal uncertainty quantification. *SIAM Review* 55(2):271–345.
- Preskill, J. 2018. Quantum computing in the NISQ era and beyond. *Quantum* 2:79.
- Rieffel, E. G., and Polak, W. H. 2011. *Quantum Computing: A Gentle Introduction*. MIT Press.
- Sahai, T.; Mishra, A.; Pasini, J. M.; and Jha, S. 2019. Estimating the density of states of Boolean satisfiability problems on classical and quantum computing platforms. *arXiv preprint arXiv:1910.13088*.
- Selby, A. 2013. Qubo-chimera. Online: <https://github.com/alex1770/QUBO-Chimera> [Accessed 28 October 2019].
- Selby, A. 2014. Efficient subgraph-based sampling of Ising-type models with frustration. *arXiv preprint arXiv:1409.3934*.
- Sommerville, I. 2005. Integrated requirements engineering: A tutorial. *IEEE Software* 22(1):16–23.
- Tamura, N.; Taga, A.; Kitagawa, S.; and Banbara, M. 2009. Compiling finite linear CSP into SAT. *Constraints* 14(2):254–272.
- Tao, T. 2010. 254A, Notes 1: Concentration of measure. *What’s New (Blog)*.
- Tao, T. 2012. *Topics in random matrix theory*, volume 132. American Mathematical Soc.
- Ushijima-Mwesigwa, H.; Negre, C. F.; and Mniszewski, S. M. 2017. Graph partitioning using quantum annealing on the D-Wave system. In *International Workshop on Post Moores Era Supercomputing*, 22–29. ACM.
- Vazquez-Chanlatte, M.; Jha, S.; Tiwari, A.; Ho, M. K.; and Seshia, S. 2018. Learning task specifications from demonstrations. In *Advances in Neural Information Processing Systems*, 5367–5377.
- Venturelli, D.; Marchand, D. J.; and Rojo, G. 2015. Quantum annealing implementation of job-shop scheduling. *arXiv preprint arXiv:1506.08479*.
- Walsh, T. 2000. SAT v. CSP. In *International Conference on Principles and Practice of Constraint Programming*, 441–456. Springer.
- Wang, F., and Landau, D. 2001. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters* 86(10):2050.
- Xu, K., and Li, W. 2000. Exact phase transitions in random constraint satisfaction problems. *Journal of Artificial Intelligence Research* 12:93–103.