New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG IOP Institute of Physics

PAPER • OPEN ACCESS

# Max 2-SAT with up to 108 qubits

To cite this article: Siddhartha Santra *et al* 2014 *New J. Phys.* **16** 045006

View the article online for updates and enhancements.

New Journal of Physics
The open access journal at the forefront of physics
Deutsche Physikalische Gesellschaft DPG | IOP Institute of Physics

# Max 2-SAT with up to 108 qubits

**Siddhartha Santra[1], Gregory Quiroz[1], Greg Ver Steeg[2,3] and Daniel A Lidar[4]**
[1] Department of Physics and Center for Quantum Information Science and Technology, University of Southern California, Los Angeles, California 90089, USA
[2] Information Sciences Institute, University of Southern California, Marina del Rey, California, USA
[3] Department of Computer Science and Center for Quantum Information Science and Technology, University of Southern California, Los Angeles, California 90089, USA
[4] Departments of Electrical Engineering, Chemistry, and Physics, and Center for Quantum Information Science and Technology, University of Southern California, Los Angeles, California 90089, USA
E-mail: santra@usc.edu

Received 27 July 2013, revised 3 February 2014
Accepted for publication 17 February 2014
Published 8 April 2014
*New Journal of Physics* **16** (2014) 045006
doi:10.1088/1367-2630/16/4/045006

**Abstract**
We experimentally study the performance of a programmable quantum annealing processor, the D-Wave One (DW1) with up to 108 qubits, on maximum SAT problem with 2 variables per clause (MAX 2-SAT) problems. We consider ensembles of random problems characterized by a fixed clause density, an external parameter which we tune through its critical value in our experiments. We demonstrate that the DW1 is sensitive to the critical value of the clause density. The DW1 results are verified and compared with akmaxsat, an exact, state-of-the-art algorithm. We study the relative performance of the two solvers and how they correlate in terms of problem hardness. We find that the DW1 performance scales more favorably with problem size and that problem hardness correlation is essentially non-existent. We discuss the relevance and limitations of such a comparison.

Keywords: adiabatic quantum computation, quantum annealing, computational complexity

## 1. Introduction

Adiabatic quantum computation (AQC) is a model of solving computational problems, in particular hard optimization problems, by evolving a closed system in the ground state manifold

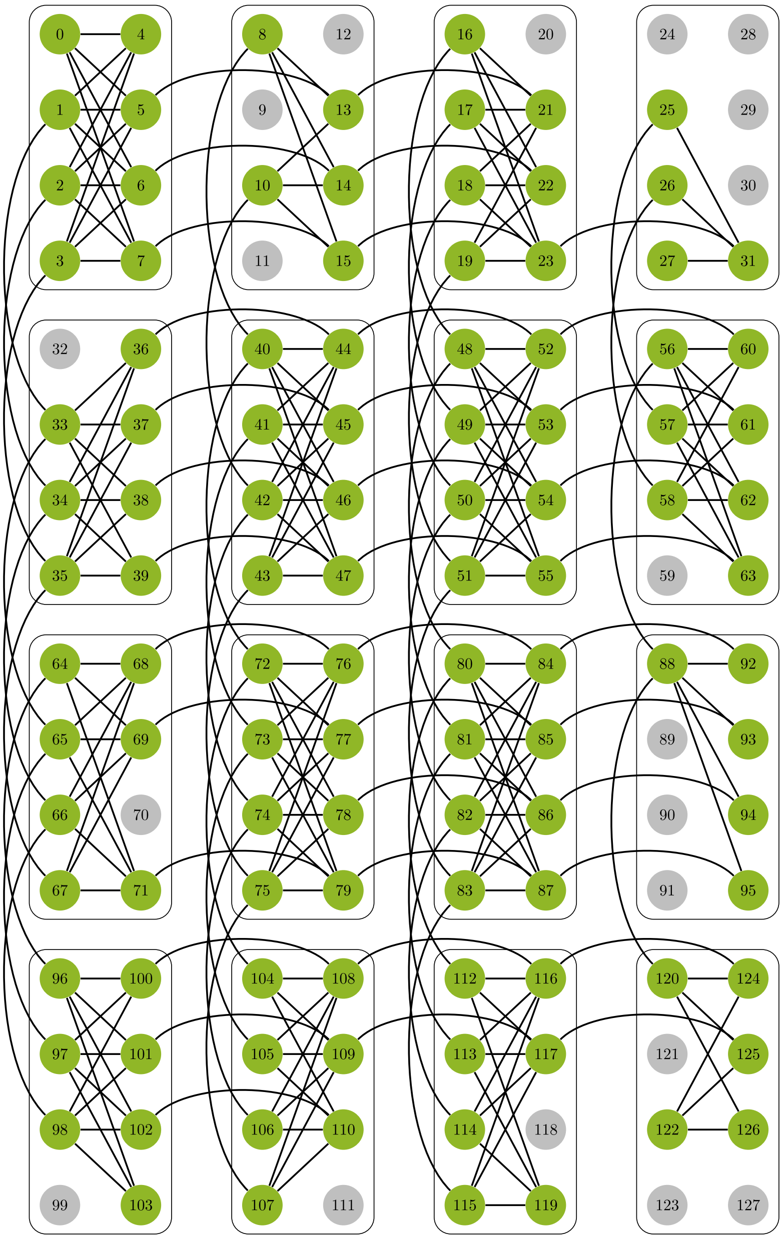

**Figure 1.** The D-Wave One Rainer DW1 processor consists of $4 \times 4$ unit cells of eight qubits (circles), connected by programmable inductive couplers (lines). The 108 green (grey) circles denote functional (inactive) qubits. Most qubits connect to six other qubits. In the ideal case, where all qubits are functional and all couplers are present (as in the central four unit cells), one obtains the non-planar 'Chimera' connectivity graph of the DW1 processor [44, 45].

of an adiabatic Hamiltonian $H(t)$ with $t \in [0, t_f]$ [1, 2]. The ground state of the beginning Hamiltonian $H_B = H_{\mathrm{ad}}(0)$ is assumed to be easily prepared, while the ground state of the problem Hamiltonian $H_P = H_{\mathrm{ad}}(t_f)$, represents the solution to the computational problem. AQC has been proven to be polynomially equivalent to standard, closed-system, circuit model QC [3–7], but so far it is unclear whether this equivalence extends to the open system, non-zero temperature setting. There is some theoretical evidence of inherent robustness of open system AQC [8–13] and scalability using currently available technology. A case in point are the D-Wave processors, comprised of superconducting rf SQUID flux qubits [14]. Recent experimental evidence [15–19] suggests that the first commercial generation D-Wave One (DW1) 'Rainier' processor (with up to 128 qubits) implements physical quantum annealing (QA) [20], a non-zero temperature, non-universal form of AQC, whose algorithmic performance has been extensively discussed in the literature [21–30]. QA can also be understood as the quantum-mechanical version of the SA [31] algorithm for optimization problems. While SA employs the slow annealing of (classical) thermal fluctuations to converge on the ground state manifold, QA additionally uses quantum fluctuations. There is extensive numerical [23–25] as well as analytical [28] evidence which shows that QA can be more efficient than SA for the problem of finding ground states of classical Ising-type Hamiltonians.

In this work we experimentally study the performance of physical QA, using the DW1 processor, on MAX 2-SAT optimization problems (maximum SAT problem with two variables per clause) [32]. We examine both the scaling with problem size and the classical phase transition in problem hardness as a function of the clause density, i.e., the ratio of the number of clauses to variables. The clause density is related to computational complexity, is associated with rigorous bounds, and is a natural external parameter for random MAX 2-SAT, as the problem exhibits a 'hardness' phase transition at a critical value $\alpha_c = 1$ [33, 34]. We present evidence for this transition in our DW1 experiments. Thus the clause density serves as a tunable hardness parameter for analyzing performance that is specific to the MAX 2-SAT problem.

One might hope to be able to detect a quantum speedup by comparing physical QA to highly optimized classical solvers. While recent work [35] attempted to show that the latest generation of D-Wave processors (the D-Wave Two 'Vesuvius' processor, with up to 512 qubits) could outperform the best classical solvers on random instances of MAX 2-SAT[5] concurrent results already demonstrated a classical stochastic solver outperforming the DW1

[5] In more detail, the McGeoch and Wang (MW) study [35], working with the DW2, used a 439 qubit subgraph of Chimera and considered three problems: (1) Chimera-structured QUBO instances (this is actually an ensemble of uniform samples from the Ising model on Chimera with $J_{ij}$, $h_j \in \{-1, 1\}$ [36]), (2) Weighted Max 2-SAT, (3) the Quadratic Assignment Problem. Their main conclusion is that in their experiments the DW2 (together with a software layer called Blackbox) outperformed the software against which it was tested. In the case of problem (1), the DW2 is reported as outperforming its nearest rival (CPLEX), amongst those tried, by a factor of 3600. The times recorded by MW were for the first point that CPLEX found the optimal solution, and not the time at which it proved it optimality. However, several researchers have reported classical implementations for all three problems which outperform the DW2 and in particular the MW benchmarks [36, 37]. Our ensemble of MAX 2-SAT problems differs from the weighted MAX 2-SAT problems considered by MW, since we used uniform weights with the additional constraint of a fixed clause density. In addition, unlike MW's case (2), our MAX 2-SAT problem ensemble inherits the native Chimera graph structure along with the connectivity contraints of the processor by design, which eliminates the need for using the Blackbox layer (that uses conventional heuristics along with hardware queries), resulting in a more transparent comparison. While we do not tune the implementation of `akmaxsat` to the structure of processor-compatible problems, this lets us compare its performance on truly random and processor-compatible problems without any bias. Comparison of our results to the native (Chimera-structured) case (1) studied by MW is not straightforward because the resulting Ising ensemble does not correspond to MAX 2-SAT with a fixed clause density. Moreover, our focus is not on the absolute time to solution (which is somewhat arbitrary anyway since it depends on variables such as the type of processor, its clock speed, and the number of cores), but rather on the scaling with problem size.

processor for an ensemble of random Ising spin glass problems with a native embedding on the Chimera graph [18] (see figure 1). Nevertheless, the competitive nature of the results along with the mounting evidence of quantum phenomena [15–19] suggests the intriguing but currently unproven possibility that the D-Wave QA architecture may some day be capable of outperforming any classical solver for some ensembles of problems, though it seems inevitable that some form of error correction will eventually be required [38–42]. Our study attempts to shed light on this possibility by studying a random ensemble of MAX 2-SAT problems characterized by a given clause density $\alpha$. We verify the empirical solutions of the MAX 2-SAT problem obtained by the DW1 using `akmaxsat` [43], a state-of-the-art, exact branch and bound algorithm, which we also use as a performance benchmark. We find that the DW1 and `akmaxsat` exhibit not only distinct scaling, but also very different sensitivities to problem hardness. In fact, we show that over the random ensemble that is characterized by a fixed clause density and is compatible with the Chimera graph, the DW1 has a scaling with problem size that is better than `akmaxsat`'s, and there is no correlation between the two solvers in terms of problem hardness.

However, we hasten to point out that our comparison is not unproblematic. The same work that most definitively established that the DW1 performs QA [18] also found that SA [31] was significantly faster than the DW1 on its ensemble of random Ising spin glass instances. While our ensemble is different, this does suggest that stochastic classical algorithms such as SA, rather than exact and deterministic ones such as `akmaxsat`, are the better benchmark. Moreover, [18] also found that other exact, deterministic classical solvers scale better than `akmaxsat` for its ensemble of random spin glass problems[6]. Our study does therefore certainly not settle the question of a quantum speedup for physical QA, but should rather be seen as a first indication that MAX 2-SAT is an interesting candidate for such a speedup, when perceived through the lens of fixed clause density ensembles. Follow-up studies using the 512-qubit D-Wave Two (DW2) will shed more light on the scaling question.

The structure of the paper is as follows: in section 2 we provide pertinent background on the MAX 2-SAT optimization problem and its phase transition as a function of the clause density. In section 3 we briefly review the DW1, describe the procedure for mapping MAX 2-SAT instances into Ising problems, and define the restricted ensemble of DW1-compatible problems. Section 4 presents our results: we compare the MAX 2-SAT success probabilities and time to solution using the DW1 processor and `akmaxsat`, discuss the evidence for a transition at $\alpha_c$, and study the problem hardness correlation between the DW1 and `akmaxsat`. We present our conclusions and discuss scope for future work in section 5. Various supplementary details are presented in the appendices.

## 2. MAX 2-SAT background

In this section we briefly review pertinent theoretical background concerning the random MAX 2-SAT problem and its solution methods. We focus on random ensembles characterized by a fixed clause density.

[6] Specifically, the biqmac algorithm [46] used in the spin glass server [47], exact belief propagation using bucket sort [48] and a related divide-and-conquer algorithm.

### *2.1. 2-SAT and MAX 2-SAT*

Many multivariate problems of practical interest involve determining values of variables $\{x_i\}_{i=1}^N$, collectively called a configuration $x$, that extremise the value of some objective. Satisfiability (SAT) problems are one such class of problems defined in terms of $N$ Boolean variables and a set of $M$ constraints between them where each constraint takes the special form of a clause. An example of a problem instance of a 2-SAT problem, written in conjunctive normal form (CNF), and also called a formula, is:

$$F(x) := (x_1 \vee x_2) \wedge (x_1 \vee \neg x_4) \wedge \cdots \wedge (x_N \vee x_4), \tag{1}$$

which is the logical AND of $M$ clauses, where each clause itself is a logical OR of exactly two literals, defined as a variable or its negation. A clause that evaluates to TRUE (FALSE) is called SAT (UNSAT). The 2-SAT problem is a *decision problem* of determining whether there exists a collection of Boolean values for $x$ such that equation (1) evaluates to TRUE, in which case the formula is said to be satisfiable. A related *optimization problem* known as MAX 2-SAT is to find the variable assignment which maximizes the number of satisfied clauses. While 2-SAT is in the complexity class P, i.e., it admits a polynomial (in fact linear [49]) time solution, MAX 2-SAT, like 3-SAT, is NP-hard [50].

### *2.2. Clause density and phase transition*

In this work we are interested in random MAX 2-SAT, where 2-SAT problem instances are generated by choosing uniformly at random, with fixed clause density, from among all possible clauses, without clause repetition (the same clause may not appear twice). The clause density is

$$\alpha = M/N. \tag{2}$$

As the number of clauses grows at a fixed number of variables, it becomes harder to satisfy all clauses. Indeed, the probability of SAT versus the clause density $\alpha = M/N$ exhibits a phase transition at a critical value $\alpha_c = 1$ in the thermodynamic limit ($N \to \infty$), whose finite-size scaling has also been studied [51, 52]. Thus, the clause density is an important external parameter. An intuitive explanation for the specific value of $\alpha_c$ is that for each clause only one of the two variables needs to be TRUE. Therefore when there are $N$ variables and $N$ clauses, one essentially uses up one variable to satisfy each clause.

The phase structure of MAX 2-SAT has also been extensively studied. Coppersmith *et al* proved that MAX 2-SAT exhibits a phase transition at the same critical clause density as 2-SAT [33]. Namely, in the large $N$ limit, to the left of the critical clause density the maximum fraction of clauses that are satisfiable is almost always 1, while to the right a fraction of all clauses are unsatisfiable. In the large clause density limit, the fraction of clauses that are satisfiable is almost surely the same as that expected from a random assignment, 3/4. Near the phase transition, we have the most uncertainty about the correct fraction of satisfied clauses. For k-SAT with $k > 2$, this type of uncertainty near the phase transition has been linked to the appearance of an easy-hard-easy pattern of computational complexity for backtracking solvers [53]. The relevance of the phase structure for the difficulty of the decision version of MAX 2-SAT has been empirically confirmed by Shen and Zhag [54]. The finite-size scaling window of the MAX 2-SAT phase transition has a clause density width of $\Theta(N^{-1/3})$ [33].

These and other results [32] suggest that it is natural to explore random MAX 2-SAT ensembles characterized by a fixed clause density, and this is our focus here.

### *2.3. Polynomial time approximation scheme*

Many NP-hard problems can be *approximately* solved to arbitrary precision in polynomial time. A $\rho$-polynomial time approximation scheme ($\rho$-PTAS) for MAX 2-SAT is an algorithm that provides an assignment of variables that provably satisfies a number of clauses within at least a fraction $\rho$ of the maximum number of clauses that can be satisfied for any formula. Goemans and Williamson [55] demonstrated a $\rho$-PTAS for MAX 2-SAT with $\rho \approx 0.87$, and an improved version of their result achieves $\rho \approx 0.94$ [56]. On the other hand, it has also been shown that no $\rho$-PTAS exists for $\rho > 21/22 \approx 0.95$ unless P = NP [57]. Thus, in the worst case, it is not only difficult to find an assignment that satisfies the maximum number of clauses, it is also difficult to find an assignment that comes close. We shall return to the $\rho$-PTAS issue from an experimental perspective in section 5.

### *2.4. Optimized classical numerical solvers*

Optimized exact classical MAX 2-SAT solvers have been extensively studied and regularly compete in an annual competition [58]. Here by 'exact' we mean that the solver is guaranteed to eventually return a correct answer. The basic idea behind the most successful exact solvers is to combine a branch and bound algorithm that searches the (exponentially large) tree of possible assignments [59] with heuristics to improve performance. Improvements come about in two ways. First, branches of the search space are avoided by intelligently upper bounding the maximum number of clauses that can be satisfied in that branch. Second, heuristics are used to simplify a formula, reducing the number of clauses or variables. In this work we benchmark the DW1 processor against a recent MAX SAT competition winner, `akmaxsat` [43], that incorporates all of these techniques. We motivate this choice in section 4.

## 3. Ensembles of 2-SAT problems and their restriction to the DW1 processor

We focus here on the average complexity for an ensemble of MAX 2-SAT problems characterized by a fixed clause density $\alpha$. The behavior of algorithms with respect to an ensemble may be taken to signify the *typical* behavior when given a specific problem instance. We note that it is not known whether this typical behavior implies anything about the worst case complexity, i.e., MAX 2-SAT is not known to be random self-reducible, unlike certain NP problems [60, 61].

### *3.1. QA using the DW1*

The DW1 implements the QA Hamiltonian:

$$H(t) = A(t)H_B + B(t)H_P, \qquad t \in [0, t_f], \tag{3}$$

where the 'annealing schedules' $A(t)$ and $B(t)$ (shown in appendix A) are, respectively, monotonically decreasing and increasing functions of time, satisfying $A(0) \gg \max(\mathrm{k}_B\mathrm{T}, \mathrm{B}(0))$ and $B(t_f) \gg A(t_f)$. The beginning and problem Hamiltonians implemented on the DW1 correspond to a transverse-field, non-planar Ising model, i.e.,

$$H_B = \sum_{j\in V}\sigma_j^x \tag{4a}$$

$$H_P = \sum_{j\in V} h_j\sigma_j^z + \sum_{(i,j)\in E} J_{ij}\sigma_i^z\sigma_j^z \tag{4b}$$

where $\sigma_j^{x(z)}$ represent the spin-1/2 Pauli matrices for the *j*th qubit. Thus to solve MAX 2-SAT problems using the DW1 we map these problems to the Ising model. In the DW1 the $N$ rf SQUID flux qubits occupy the vertices $V$ of the so-called 'Chimera' graph (see figure 1), with maximum degree 6, and are coupled inductively along the edges $E$ of this graph. The local fields $h_j$ and the couplers $J_{ij}$ are programmable and once chosen they define a 'problem instance'. Each 'annealing run' corresponds to evolving $H(t)$, with a preprogrammed and fixed set of local fields and couplers, from $t = 0$ to a predetermined annealing time $t_f$, followed by a projective measurement of all qubits in the computational basis, i.e., the eigenbasis of the Ising Hamiltonian $H_P$. Each such measurement results in a spin configuration $\{s_1,\ldots,s_N\}$, where $s_j = \pm 1$ is the eigenvalues of $\sigma_j^z$. By repeating these annealing runs many times one builds up statistics of spin configurations for a given problem instance. The processor can then be reprogrammed to generate statistics for a new problem instance.

### *3.2. Mapping MAX 2-SAT to equivalent Ising problems*

In order to solve instances of MAX 2-SAT on the DW1, we must construct the problem Hamiltonian $H_P$ of equation (4*b*), such that the ground state configuration encodes the satisfying assignment for the problem instance. Following the prescription of [1] for the conversion of SAT problems to finding the ground state(s) of quantum Hamiltonians, as a first step we transform from Boolean to binary variables, letting TRUE = 0 and FALSE = 1, so that the truth table of the OR function becomes the multiplication of the binary variables. We next identify the binary variables $\{x_j\}_{j=1}^N$ of a 2-SAT formula with the ±1 eigenstates of the Pauli spin operator $\sigma_j^z$ acting on qubit $j$, i.e.,

$$\sigma_j^z \left| x_j \right\rangle = (-1)^{x_j} \left| x_j \right\rangle, \quad x_j \in \{0,\, 1\}. \tag{5}$$

We also define variables $v_j^k \in \{-1,\, 0,\, 1\}$, where the indices $j = \{1,\, 2 \ldots N\}$ and $k = \{1,\, 2 \ldots M\}$ label the variables and clauses respectively, with $v_j^k = -1$ $(+1)$ if $x_j$ appears negated (unnegated) in the *k*th clause and $v_j^k = 0$ for all clauses that $x_j$ does not appear in. Each two-variable clause, $\Omega_k$, $k \in \{1,\, 2,\, \ldots,\, M\}$, in an arbitrary 2-SAT formula $F = \Omega_1 \wedge \Omega_2 \wedge \ldots \wedge \Omega_M$ is then translated into a corresponding 2-local term in the problem Hamiltonian of the form

$$H_{\Omega_k} = \frac{\mathbb{1} - v_{j_1}^k\sigma_{j_1}^z}{2}\frac{\mathbb{1} - v_{j_2}^k\sigma_{j_2}^z}{2}. \tag{6}$$

It is easy to check that in this manner if $\{x_{j_1},\, x_{j_2}\} \in \{0,\, 1\}^2$ violate the clause then $H_{\Omega_k}$ is associated with an energy penalty of 1, and zero otherwise. Rather than taking the logical AND of all clauses as in the original 2-SAT problem, the problem Hamiltonian is now constructed as

$$H_P = \sum_{\Omega_k \in F} H_{\Omega_k}, \tag{7}$$

i.e., the sum of the energies of all $M$ clauses contained in the 2-SAT instance $F$. This means that the ground state of $H_P$ corresponds to the bit assignment that violates the minimal number of clauses, i.e., the ground state is the solution to the MAX 2-SAT problem for the problem instance $F$. A generic computational basis state of the system can be written as $|\,\psi\rangle = \left|\,x_1 x_2 \dots x_N\right\rangle$. In the case that $\{x_j^*\}$ is a satisfying assignment for the formula $F$ we have the correspondence

$$F|_{\{x_j^*\}} = 1 \Rightarrow H_P\,|\,\psi^*\rangle = 0, \tag{8}$$

where $|\,\psi^*\rangle = \left|\,x_1^* x_2^* \dots x_N^*\right\rangle$, while all non-satisfying assignments correspond to computational states with positive energy. In the case of MAX 2-SAT the ground state might have a positive energy $E_{\min} > 0$ and the question becomes to determine the assignment $|\psi\rangle$ such that $H_P\,|\,\psi\rangle = E_{\min}\,|\,\psi\rangle$.

Written out in detail, equation (7) becomes

$$H_P = \frac{1}{4} \sum_{\Omega_k \in F} 2\mathbb{1} - v^k_{j_1}\sigma^z_{j_1} - v^k_{j_2}\sigma^z_{j_2} + v^k_{j_1} v^k_{j_2} \sigma^z_{j_1}\sigma^z_{j_2}\ , \tag{9}$$

and upon equating with the problem Hamiltonian in equation (4*b*), after rescaling by a factor of 4 and dropping the constant term, we obtain the local fields $h_j$ and the couplings $J_{ij}$ in terms of the parameters of the given MAX 2-SAT instance

$$h_{j_i} = -\sum_k v^k_{j_i}\ , \quad J_{j_1 j_2} = \sum_k v^k_{j_1} v^k_{j_2}\ , \tag{10}$$

where $i \in \{1, 2\}$ and the indices $j_1, j_2$ are the qubit indices on the Chimera graph.

### *3.3. Restricted ensemble of DW1 processor-compatible 2-SAT problems*

The DW1 process-compatible problem instances must satisfy a number of constraints, namely $N \leqslant 108$ and the DW1 Chimera graph connectivity. To account for these constraints we generated restricted ensembles $\mathcal{E}_{\mathrm{DW}}(N, \alpha)$ with 13 different numbers of variables and 20 different clause density values, as follows:

- $N \in \{16, 24, 32, 39, 46, 53, 60, 67, 75, 80, 87, 98, 108\}$, and $\alpha = [0.1, 2.0]$ in increments of 0.1.
- We define DW1 processor-compatible problem instances as those instances whose clauses are formed by two literals $x_{j_1}, x_{j_2}$ which correspond to qubits $j_1, j_2$ on the Chimera graph $G = (E, V)$ that are active as well as coupled, i.e., $\{j_1, j_2\} \in V$ and $(j_1, j_2) \in E$. Recall that not all qubits are active (see figure 1).
- When $\alpha < 1/2$ there are more variables than can fit into the clauses. For $\alpha < 1/2$, any variable that did not appear in a clause was not used in $H_P$.
- At each value of $N$ and $\alpha$ we generated an ensemble, $\mathcal{E}_{\mathrm{DW}}(N, \alpha)$, of 500 DW1 processor-compatible random 2-SAT problem instances. We excluded all instances involving

identical clauses. In our ensembles we applied negation to each of the two variables representing the qubits uniformly at random.

We thus have a total of $13 \times 20 = 260$ such ensembles $\mathcal{E}_{\text{DW}}(N, \alpha)$, with a total of $260 \times 500 = 130,\,000$ DW1 processor-compatible problems.

To cover a range of interesting clause density values we used a maximum value of $2\alpha_c = 2$, thus ensuring that our instances were all well within the range of the the finite-size scaling window of the phase transition, whose width is $\Theta(N^{-1/3})$ [33]: four our range $16 \leqslant N \leqslant 108$ we have $0.40 \geqslant N^{-1/3} \geqslant 0.21$. The maximum value of $\alpha$ supported by the DW1 processor is discussed in appendix B. We note that having enforced an equal probability for negated and unnegated variables somewhat restricts the hardness as it is known that an unbalanced probability of negation can lead to harder instances [62].

To test how well our DW1-restricted instances approximate an unrestricted random ensemble, we also generated ensembles, $\mathcal{E}(N, \alpha)$, of 500 random 2-SAT problems with a given number of variables and clause density and no constraint on the literals that comprise a single clause, except that no clauses are repeated. A comparison between $\mathcal{E}_{\text{DW}}(N, \alpha)$ and $\mathcal{E}(N, \alpha)$ is presented in the next section.

## 4. Experimental and numerical results

In this section we report on our experimental and numerical results for the ensembles of MAX 2-SAT problems described above. A complete description of the settings under which we ran the `akmaxsat` algorithm is given in appendix C.

We compare the scaling of the solution time required by `akmaxsat` to the empirical probability of the correct ground state found by the DW1 processor, from which we compute an extrapolated time required to achieve a certain solution threshold accuracy. This is, of course, not entirely an 'apples-to-apples' comparison, since it compares an exact algorithm with a probabilistic machine. Moreover, there exist faster stochastic classical algorithms [18]. However, to check the proposed DW1 solutions for correctness an exact algorithm is required, and we decided to use `akmaxsat` due to its excellent performance on the benchmark problem sets of MAX 2-SAT used at the MAXSAT-2009 and MAXSAT-2010 evaluations for state-of-the-art MAX SAT solvers [58]. We make no claims here as to the significance of our results in the larger context of whether experimental QA can outperform *all* classical algorithms. Rather, we focus on the scaling comparison with a state-of-the-art exact classical algorithm, and on whether there is any correlation in problem hardness between this classical solver and the DW1. We find that the DW1 has a scaling that is better than that of `akmaxsat`, a performance gap which increases with the clause density, and that there is rapidly decreasing correlation between the DW1 and `akmaxsat` for the hardness of problem instances, as $\alpha$ increases. In addition, we closely examine the behavior in the vicinity of the critical clause density.

### 4.1. Comparison of `akmaxsat` for the random and DW1-compatible ensembles

As noted above, the DW1 processor-compatible problem instances are restricted by the Chimera graph connectivity (see figure 1). To test how the resulting ensemble $\mathcal{E}_{\text{DW}}(N, \alpha)$

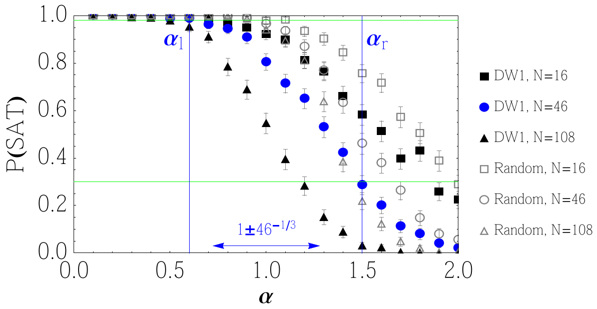

**Figure 2.** Probability of a random formula being satisfiable as a function of $\alpha$ for DW1-compatible (solid) and random (hollow) instances. An example of how to estimate the width of the scaling window is highlighted in blue for the DW1 processor instances at $N$ = 46 (solid circles). Error bars represent 95% confidence intervals according to a bootstrap estimation procedure.

compares with an unrestricted random ensemble $\mathcal{E}(N, \alpha)$, we analyzed the performance of `akmaxsat` for instances drawn from each ensemble.

The time to solution is generally greater for the random ensemble; however, substantial differences between $T_{\text{soln}}$ for DW1 processor-compatible and random instances are not observed for a majority of the $\alpha$ and $N$ values considered, as described in section 3.3. The `akmaxsat` solver is deterministic, yet the solution time for a given instance may deviate due to variations in the initial starting point of the algorithm; hence, all instances are averaged over 10 algorithm implementations. (A view of this data is offered in appendix D; see figure D1.)

We find that the solution time $T_{\text{soln}}$ is generally somewhat larger for the unrestricted random ensemble $\mathcal{E}(N, \alpha)$, implying (unsurprisingly) that on average the unrestricted instances are somewhat harder for `akmaxsat` than the DW1 processor-compatible case. The differences between the solution times of the two ensembles are relatively small, differing by at most 10%, around $N = 67$ with $\alpha$ = 2.0. Solution times are essentially identical for $\alpha \lesssim 1$ and $N \lesssim 40$, with small regions of the $(\alpha, N)$-space where the $\mathcal{E}_{\text{DW}}(N, \alpha)$ ensemble requires somewhat larger average solution times. The solution times differ somewhat more for $N > 40$ and $\alpha > 1$. It is important to remember in any case that hardness is also dependent upon the solution method. Indeed, we show below that instances which are hard for `akmaxsat` need not be hard for the DW1. Moreover, we show later (see figure 2) that according to another hardness measure, processor-compatible instances are harder than random ones.

### *4.2. Experimental results*

In this subsection we describe our DW1 results. Details about the experimental settings are given in appendix A.

*4.2.1. Mean success probability as a function of N and $\alpha$.* In figure 3, we present experimental DW1 processor results for the mean success probability as a function of $\alpha$ for various $N$ values. The average for each data point is over the number of annealing runs and instances. Each set of data points includes a least squares fit using the following function as an ansatz:

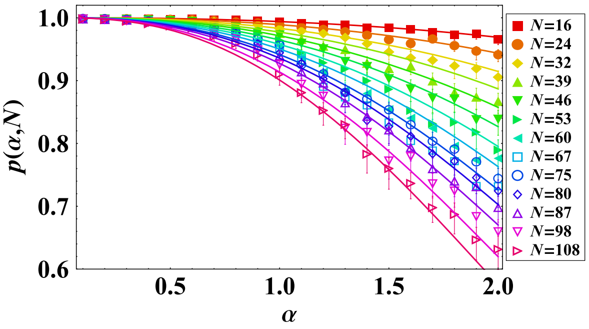

**Figure 3.** Data points are the mean experimental success probability of obtaining a correct MAX 2-SAT solution for the DW1, as a function of the clause density $\alpha$ for various values of $N$. Solid lines are a fit using equation (11) (see text). The probability decreases as the number of clauses increases for a given $N$. The $\alpha$-dependence becomes more pronounced as the problem size $N$ increases, indicating that problems become harder. All data points are averages over 500 random DW1 processor-compatible instances with each instance averaged over 100 annealing runs. Error bars denote 95%-confidence intervals. Note the absence of a specific feature at the critical clause density $\alpha_c = 1$. However, see figure 5.

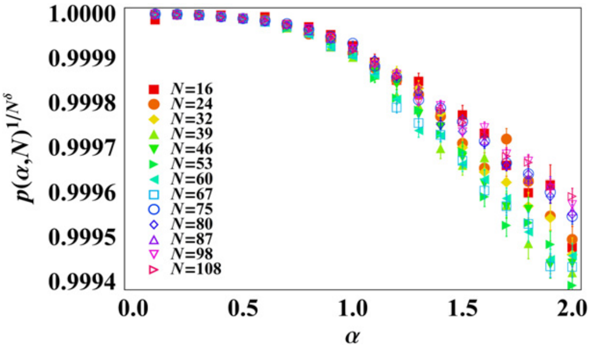

**Figure 4.** Data collapse analysis for the mean experimental success probability. Shown is the function $p(\alpha, N)^{N^{-3/2}}$ evaluated at the experimental $p(\alpha, N)$ values shown in shown in figure 3. Error bars denote 95%-confidence intervals. While we found that $\delta = 3/2$ works well, we do not have an explanation for this particular value.

$$p(\alpha, N) = \exp\left(-A\alpha^{\gamma}N^{\delta}\right), \tag{11}$$

where we normalized $p(0, 0) = 1$, and we find $A = 9.28[6] \times 10^{-5}, \gamma = 2.40[9]$ (the number in square brackets is the round-off of the remaining digits), and $\delta = 3/2$ for all $\alpha$ and $N$ considered. To test the universality of this ansatz we show a data collapse in figure 4, which indicates a reasonable fit to the data, yielding a good collapse for larger values of $N$ ($N \geqslant 75$) and slightly overestimating the success probability at low $N$ values, as can also be seen from figure 3. Note that the success probability decreases more rapidly with increasing $\alpha$ than with increasing $N$.

Several fundamental factors contribute to the decrease of the success probability: the inherent increase in problem difficulty as $N$ increases results in a smaller ground state gap, which in turn enhances coupling to the finite-temperature environment in the form of thermal

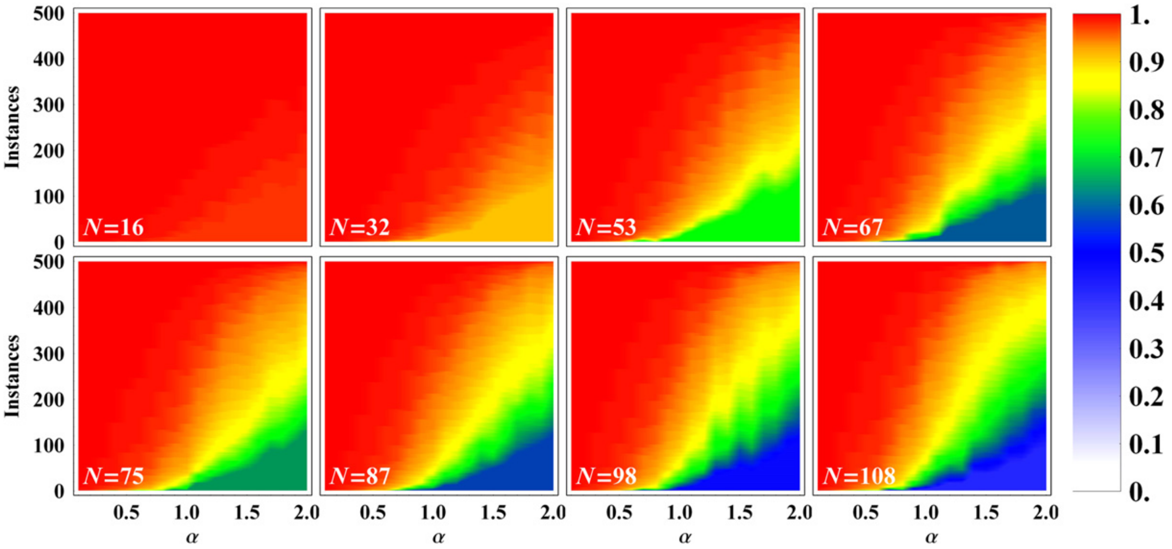

**Figure 5.** Probability of obtaining a correct MAX 2-SAT solution for each of the 500 DW1 processor-compatible instances sorted from least to greatest probability for $\alpha \in [0.1, 2.0]$ and various $N$ values. In agreement with figure 3, problem instances become harder (i.e., have a lower success probability) with increasing $\alpha$ and $N$. Note the hardness transition around $\alpha_c = 1$.

excitations, and non-adiabaticity due to the finite annealing time [63]. Added to this is the qubit approximation to the rf SQUID [64]. Control errors such as miscalibration and the finite digital-to-analog (DAC) converter resolution contribute as well. A larger number of these occur as both $N$ and the number of clauses increase. A detailed discussion of the effect of control errors and their contribution to the observed success probabilities is presented in appendix E.

*4.2.2. Sorted success probabilities as a function of N and α.* A fine-grained characterization of the ensemble of problem instances for fixed $N$ is given in figure 5, where we present contour plots of the success probability for each of the 500 problem instances as a function of $\alpha$. That is, for each value of $\alpha$ we first sort the instances by success probability, and then plot the probabilities of the sorted instance set for that value of $\alpha$ (thus the instance number varies as one moves horizontally through each panel). The predominance of the color red in the panels indicates that most problem instances had a probability of success close to unity. As suggested by the mean case results shown in figure 3, the success probabilities decrease with increasing $\alpha$ and $N$. Perhaps most interesting is the appearance of a soft transition around $\alpha \approx 1$. That is, for all $N$ most of the problems have success probability very close to 1 for $\alpha < 1$, while lower success probabilities appear for $\alpha > 1$. This can be interpreted as an experimental indication of the critical clause density.

We take a closer look at the ensembles of problem instances with $N = 108$ in figure 6, where we show the success probability distribution for all values of $\alpha$ we studied. The distribution has a peak that moves from $p = 1$ to $p = 0.9$ as $\alpha$ increases, and develops a broad

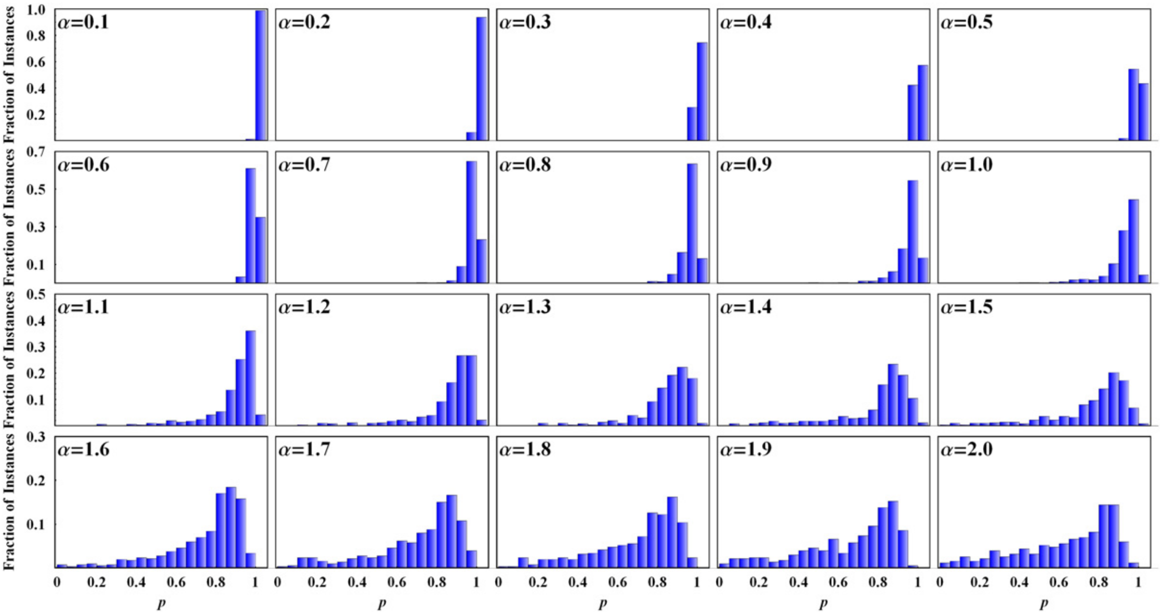

**Figure 6.** Histogram of success probabilities $p$ for each of the 500 problem instances with $N = 108$ and $\alpha \in \{0.1, \ldots, 2.0\}$. Each bar is the fraction of instances (relative to 500) with a given empirical success probability.

wing at lower success probabilities. Its unimodality is a feature that persists for all values of $\alpha$, $N$ we tried[7].

*4.2.3. Transition at the critical clause density.* While the phase transition discussed in section 2.2 becomes sharp in the limit $N \to \infty$, analytic bounds provide a more nuanced characterization for finite $N$. The bounds of Coppersmith *et al* [33] provide useful intuition but should be interpreted cautiously as they only apply for finite, but very large $N$. In the so-called 'finite-scaling window', $\alpha \in [1 - N^{-1/3}, 1 + N^{-1/3}]$, the probability that a random formula is satisfied is bounded away from 1 and 0. Additionally, the average number of clauses that cannot be satisfied is a small constant fraction of $N$. Another, softer transition occurs as we increase the clause density. The average fraction of clauses that cannot be satisfied goes as $1/4 - O(1/\sqrt{\alpha})$ ([33], Theorem 4). In the limit of large clause density, a random assignment (which fails to satisfy each clause with probability 1/4) is nearly optimal, however, our experiments are far from this parameter regime.

[7] We note that [18] found that for an ensemble of Ising spin-glass problems the histogram of success probabilities solved using the DW1 was bimodal. This result agreed with predictions of a simulated quantum annealer, but differed from that of classical SA. The difference is most likely primarily due to the different annealing time, 5 $\mu$s in [18] compared to our 1 ms. While we did not vary the annealing time in the present study, [18] observed that a longer annealing time resulted in increasingly large fraction of problems being solved with high probability, thus shifting some of the weight of the distribution from the 'hard' (low success probability) problems to the 'easy' ones. Moreover, the ensemble of Ising spin glass problems considered in [18] differed from ours in several important ways. Their coupling strengths were randomly allowed to be only ±1 with no fractional couplings. Another difference was that the couplings between two neighboring spins were completely independent of the local fields, unlike our equation (10).

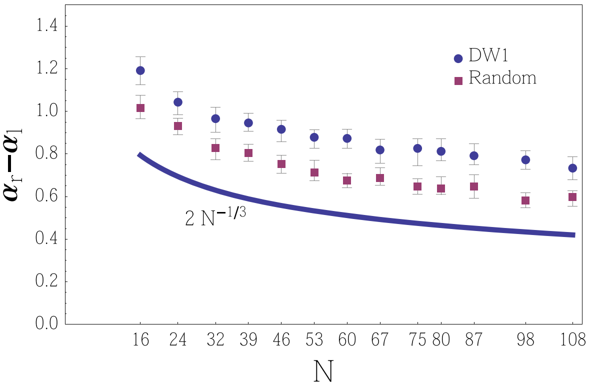

**Figure 7.** Comparison of the width of the finite-scaling window estimated empirically for random and DW1-compatible instances. Error bars represent 95% confidence intervals according to a bootstrap estimation procedure.

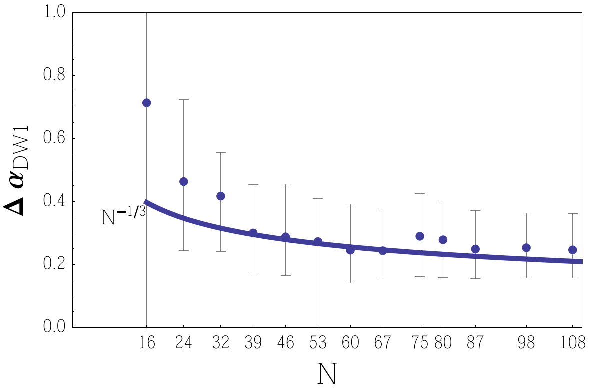

**Figure 8.** An estimate of the finite scaling effects for solving instances on DW1.

We can empirically estimate the width of the finite-scaling window by estimating the width of the region in which the probability of a formula being satisfied is far from 0 or 1 [54]. figure 2 compares the probability a formula is satisfiable for DW1-compatible and random instances for various $N$ as a function of $\alpha$. Analytically, the width of the finite-scaling window should be $2N^{-1/3}$. We estimate the value of the clause density $\alpha_l$ ($\alpha_r$) at which the probability of a random formula being satisfiable drops below 0.98 (0.3), as demonstrated in figure 2. Then we plot our empirical estimate of the width of window, $\alpha_r - \alpha_l$ as a function of $N$ in figure 7. We confirm that the scaling of the size of this window is indeed proportional to $N^{-1/3}$. Note that the observed width of the scaling window is larger than the analytic predictions (valid only for asymptotically large $N$) for both random instances and processor-compatible instances. Referring again to figures 2 and 7, we note that the DW1 processor-compatible instances have a slightly wider window whose center is shifted slightly towards lower values of $\alpha$. In other words, at a given value of $\alpha$ a processor-compatible instance will tend to be harder to satisfy. This is in interesting contrast to the situation for the solution times we found for `akmaxsat` as discussed in section 4.1.

We now show that the phase transition has computational consequences for both `akmaxsat` and DW1. Again, we use $p(\alpha, N)$, the probability that DW1 produced the correct

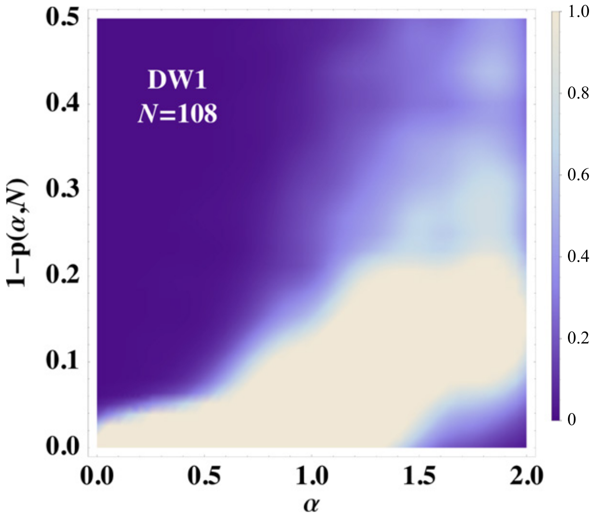

**Figure 9.** A density histogram showing the distribution of failure probabilities for random instances as a function of clause density $\alpha$ for DW1.

solution, as a proxy for problem difficulty. In figure 8, we consider a plot analogous to figure 7, except now we estimate for each $N$ the value, $\Delta\alpha_{\mathrm{DW1}} = \alpha_{\mathrm{DW1,r}} - \alpha_{\mathrm{DW1,l}}$ where $\alpha_{\mathrm{DW1,r}}$ ($\alpha_{\mathrm{DW1,l}}$) indicates the value of $\alpha$ at which the probability of successfully solving a problem for DW1 falls below the threshold 0.97 (0.99). The threshold 0.97 was set high enough that the $p(\alpha, N)$-$\alpha$ curve passes through it for every $N$. The largest minimum success probability was 0.967 for $N = 16$, as seen in figure 3. (Note that $\alpha_l$, $\alpha_r$ are empirical estimates for the boundaries of the finite-size scaling window of the probability of SAT of problems and are ensemble properties *independent* of the particular method of solution. On the other hand, $\alpha_{\mathrm{DW1,l}}$, $\alpha_{\mathrm{DW1,r}}$ are experimentally determined boundaries where we observed the signature of the scaling window on DW1's performance.) For very small $N$, many factors including control errors may affect this curve. While it seems plausible that the $N^{-1/3}$ scaling will continue to hold for larger $N$, solving larger problems on next-generation processors (such as the 512 qubit DW2) will be required to verify this hypothesis.

We also see the effect of the phase transition in figure 9, in which we compare $p(\alpha, N = 108)$ to $\alpha$ using a density histogram. A sharp change in the number of difficult instances clearly occurs around $\alpha = \alpha_c = 1$. Furthermore, the variance in problem difficulty is much higher to the right of the transition. We compare the time to solution for `akmaxsat` to $\alpha$ in figure 10. While the difficulty clearly increases with clause density, we see again that the variance in difficulty increases at the phase transition. While the appearance of difficult instances is linked to the phase transition in both cases, surprisingly the instances that are difficult appear unrelated, as we will see in section 4.4. We discuss further evidence for a transition at $\alpha_c$ in section 4.4 (see, in particular, figure 11).

### *4.3. Scaling of the time to solution: DW1 versus* `akmaxsat`

Given a success probability $p$, assuming no correlation between successive annealing runs on the DW1 (an assumption that is satisfied to a good approximation [18]), the probability of not getting a single correct answer in $k$ runs is $(1 - p)^k$. We define $p_{\mathrm{desired}}$ as the threshold probability

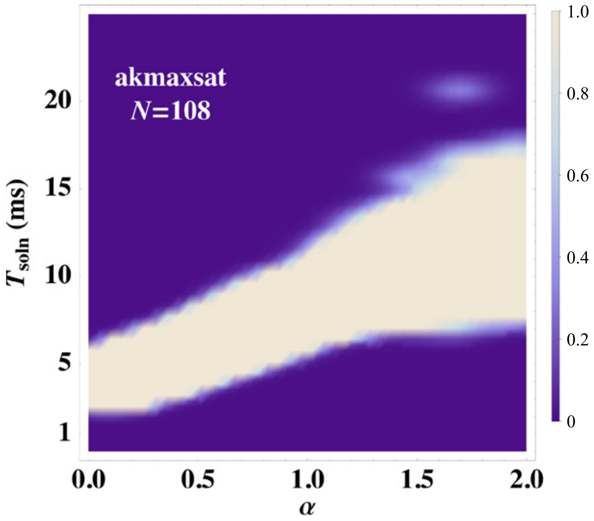

**Figure 10.** A density histogram showing the distribution of the time to solution $T_{\text{soln}}$ for random processor-compatible instances as a function of clause density $\alpha$ for `akmaxsat`.

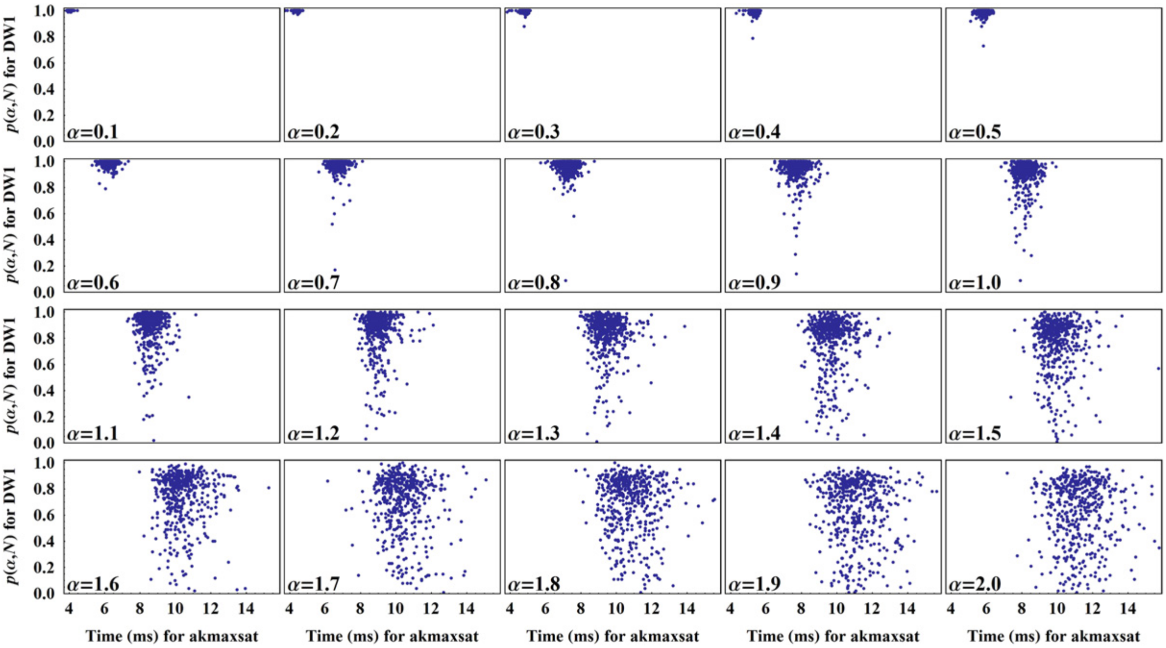

**Figure 11.** Scatter plot of success probability of DW1 vs. time taken by `akmaxsat` for each problem instance with $N = 108$ and $\alpha \in \{0.1, 0.2, \ldots, 2.0\}$ (500 instances per $\alpha$ value). While for very small clause densities all problems are easy for the DW1 and take roughly the same amount of time for `akmaxsat`, as $\alpha$ approaches $\alpha_c = 1$ from below harder problems start to appear for the DW1 and `akmaxsat` timings start to spread. Above $\alpha_c$ there is a significant spread in both the DW1 success probabilities and the `akmaxsat` timings. As a result the correlation between these two variables steadily diminishes as $\alpha$ grows.

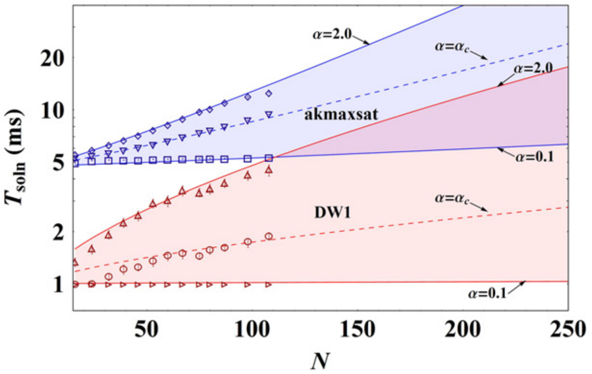

**Figure 12.** Computational effort comparison between `akmaxsat` and the DW1 as a function of the problem size $N$ and for different clause densities $\alpha$. Results are shown along with best fits for the entire data set, to the function in equation (13). Error bars denote 95%-confidence intervals. Shown are (1) `akmaxsat` with $\alpha = 0.1$ (squares), $\alpha = \alpha_c = 1$ (down-triangles), and $\alpha = 2.0$ (diamonds); (2) DW1 with $\alpha = 0.1$ (right-triangles) $\alpha = \alpha_c = 1$ (circles), and $\alpha = 2.0$ (up-triangles). Remaining values of $\alpha$ are summarized by the shaded regions. Note that while extrapolations for `akmaxsat` can be deemed reliable, this is not the case for the DW1 data, as argued in the text. Thus the DW1 extrapolations shown should be considered as merely suggestive for a limited range of $N$ values whose upper bound we do not know at present.

of getting one or more correct answers, i.e., $p_{\rm desired} = 1 - (1 - p)^k$. For a run with annealing time $t_f$ the time required to reach the ground state at least once in $k$ runs with probability $p_{\rm desired}$ is:

$$\begin{aligned} T_{\rm soln}\left(p_{\rm desired}\right) &= t_f k\left(p_{\rm desired}\right) \\ &= t_f \left\lceil \frac{\log\left(1 - p_{\rm desired}\right)}{\log\left(1 - p\right)} \right\rceil, \end{aligned} \tag{12}$$

where $t_f = 1$ ms for our experiments.

Taking $p$ as the mean success probability reported in figure 3, the extrapolated time to solution for $p_{\rm desired} = 0.99$ for $\alpha = 0.1$ (right-triangles) and $\alpha = 2.0$ (up-triangles) is plotted in figure 12 versus problem size $N$ (for complete data see appendix F). Note that for $\alpha = 0.1\ \forall\ N$, the DW1 obtained the correct solution with $p \geqslant 99\%$ on average. Clearly, this subensemble of problems was too easy for the chosen value of the annealing time, since this translates into a single repetition ($k = 1$), i.e., a constant time to solution equal to the (unoptimized) annealing time of 1ms. On the other hand, as $N$ grows the time to solution must of course eventually start to grow as well. This illustrates the danger of extrapolating to large $N$ values from experimental data based on a single (unoptimized) annealing time. In fact this conclusion also applies to the other values of $\alpha$ shown, since for a fixed annealing time the time to solution must eventually blow up as a function of $N$, due to the inevitable reduction in success probability resulting from restricting the time to settle on an optimal solution in an increasingly growing configuration space [18]. This extrapolation caveat does not apply to our `akmaxsat` data, since there we simply let the algorithm run until it finds a solution.

**Table 1.** Numerical values for the parameters appearing in equation (13). The last column is the coefficient of determination of the fit, with $R^2 = 1$ denoting a perfect fit. Note that $A$ has units of msec and all other coefficients are dimensionless.

| | $A$ | $B$ | $\gamma$ | $\delta$ | $\delta - \gamma$ | $R^2$ |
|---|---|---|---|---|---|---|
| AK | 4.750 | 0.003[6] | 3/4 | 1.106[8] | 0.356[8] | 0.999[4] |
| DW1 | 1.0 | 0.026[0] | 3/2 | 0.663[7] | −0.836[2] | 0.991[4] |

*4.3.1. Extrapolation for the fixed-$\alpha$ ensemble.* With the just-stated caveat in mind, we present a best fit to the entire DW1 data set, and separately for the `akmaxsat` timing dataset. We find that the data is well fit by the following function:

$$T_{\text{soln}}(\alpha, N) = A \exp\left(B\alpha^{\gamma} N^{\delta}\right) \tag{13}$$

where the values of the various parameters are given in table 1. The numerical values were obtained from a least squares fit followed by a data collapse of the data shown in figure 12 for both `akmaxsat` and the DW1. More details regarding the data collapse are given in appendix F, where we also present a second fit with more free parameters; this does not change our conclusions below.

Comparing the numerical values of the exponents $\gamma$ and $\delta$ of, respectively, the clause density and the number of variables in table 1 we see that while `akmaxsat` has the smaller value of the clause density exponent, the DW1 has the smaller exponent for the number of variables. This explains the better scaling for the DW1 as a function of $N$ seen in figure 12, in spite of the fact that the DW1 has a much larger value of the exponent $B$.

*4.3.2. The limit of large N and constant M.* Departing momentarily from our emphasis on the fixed-$\alpha$ ensemble, note that equation (13) can also be written as $A \exp\left(BM^{\gamma}N^{\delta-\gamma}\right)$. While viewed in this way `akmaxsat` has a better scaling with the number of clauses $M$ since it has the smaller $\gamma$ value, our fit yields a negative value for the DW1 exponent $\delta - \gamma$ of $N$, while `akmaxsat`'s exponent is positive (see table 1). Now consider the limit of large $N$ and small $\alpha$, while keeping the number of clauses $M$ constant. In this limit the probability of repeated variables vanishes, so that it becomes possible to satisfy each clause independently. A parallel processor capable of updating all clauses simultaneously would therefore solve the MAX 2-SAT problem in constant time in this limit. This is indeed the prediction of equation (13) for the DW1 time to solution, given that $\delta - \gamma < 0$:

$$\lim_{N\to\infty} T_{\text{soln}}^{\text{DW1}}(\text{const}/N, N) = A = \text{const}. \tag{14}$$

In contrast, the predicted scaling of the time to solution for `akmaxsat` diverges with $N$ even in this limit, which clearly shows its suboptimality.

The fact that the DW1 seems capable of 'recognizing' that the fixed-$M$, large $N$ limit is easy, while `akmaxsat` does not, is interesting. It suggests that the DW1 naturally acts as a parallel processor, making 'cluster moves' that simultaneously find SAT solutions for multiple clauses.

*4.3.3. Time to solution for different levels of problem hardness.* In figure 13(a) we plot, for $\alpha = 2$, the time taken by `akmaxsat` to solve the problems at eight different percentiles of the 500 instances at each $N$, and compare this in figure 13(b) to the estimated time to solution for

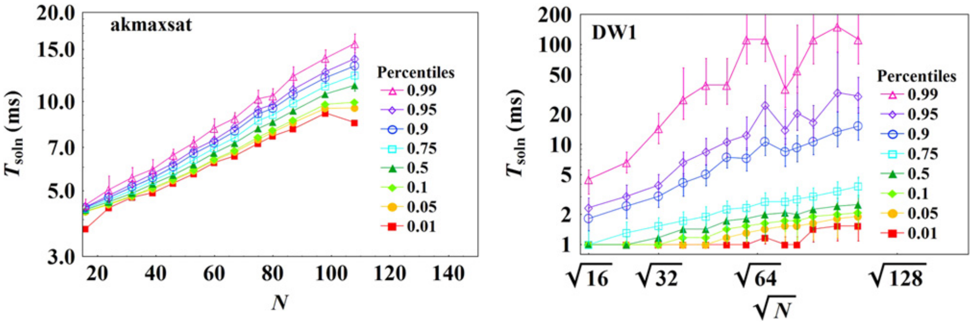

**Figure 13.** Scaling plots of the time to solution at $\alpha = 2.0$ for different percentiles of the distribution of problem hardness. (left) Log-linear for `akmaxsat`, (right) log-square-root for the DW1. In the `akmaxsat` case problem instances were sorted and binned by their required solution time. The percentiles then indicate the value of the solution time for the corresponding set of instances, with shortest and longest solution times indicated by the 0.01 and 0.99 percentiles, respectively. The error bars represent the standard error obtained over the 10 different runs of `akmaxsat`. In the DW1 case we computed the percentiles from the histograms shown in figure 6. The easiest and hardest sets of problem instances are indicated by the 0.01 and 0.99 percentiles, respectively. While the performance of `akmaxsat` appears fairly uniform across the different percentiles, the performance of the DW1 is significantly better for the lower (i.e. easier) percentiles, with a noticeable deterioration seen at the 90th percentile. Error bars represent 95% confidence intervals according to a bootstrap estimation procedure.

DW1 for the same percentiles requiring $p_{\text{desired}} = 0.99$. Note that the same percentile value in the two figures generally represent different, possibly overlapping sets of problem instances. In figure 13(a) lower and upper percentiles correspond to shorter and longer solution times, respectively, with the easiest (hardest) problems being in the 0.01 (0.99) percentile. The scaling for `akmaxsat` is approximately exponential (note the logarithmic time axis), with the higher percentiles having larger exponents. The solution time for the hardest (0.99) percentile is approximately twice that for the easiest (0.01) for $N = 108$. As seen in figure 13(b), the range of DW1 solution times varies significantly more between different percentiles than for `akmaxsat`. Disregarding fluctuations due to the control errors and the small sample size, the scaling of the DW1 solution times appears to be more favorable than for `akmaxsat` for all percentiles, and to match an exponential of $\sqrt{N}$ rather than $N$, in agreement with the scaling of the tree-width of the Chimera graph [18, 44, 45]. Once again we point out that extrapolation to larger $N$ values are unreliable due to our suboptimal annealing time. We also note that the procedure whose results are shown in figure 13 corresponds to first computing the percentiles, then estimating the time to solution; we verified that the order of these operations does not change the results.

### *4.4. Are the DW1 and* `akmaxsat` *correlated?*

In figure 11, we compare the solution time required by `akmaxsat` to the probability $p$ of finding a correct solution by the DW1 for all values of $\alpha \in \{0.1, \ldots, 2.0\}$, for $N = 108$. For $\alpha < \alpha_c = 1$ the `akmaxsat` results are strongly clustered around their mean and the clear majority of problem instances are easy for the DW1. For $\alpha \geqslant 1$ hard problem instances start to

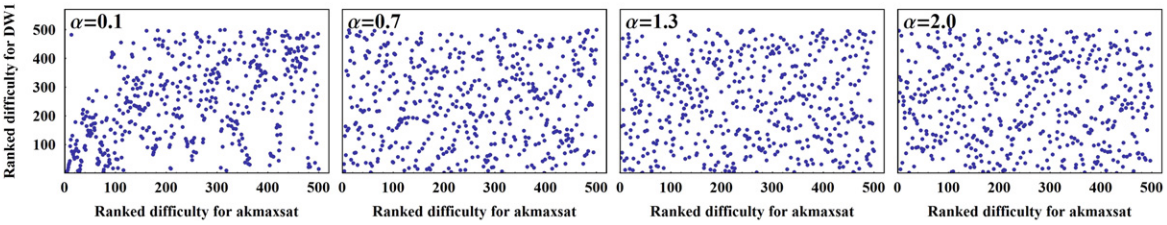

**Figure 14.** Ranked difficulty comparison (copula plot) between DW1 and akmaxsat for $N = 108$ and representative values of $\alpha \in \{0.1, 0.7, 1.3, 2.0\}$ (other values yield very similar looking plots). The instances used here are the same as in figure 11. Clearly there is essentially no correlation for problem difficulty, even for small values of the clause density.

appear and the akmaxsat timing results spread around their mean. The DW1 probabilities are much more scattered; while they still cluster somewhat near $p = 1$, there is an increasingly large spread across the entire range of possible values of $p$. The correlation between the the DW1 success probability and the akmaxsat time to solution thus steadily diminishes with $\alpha$; in particular we see that for $\alpha > \alpha_c$ there is a large spread of $p$ values for any fixed solution time.

A more direct measure of correlations is given in figure 14. Shown in this figure is a ranked difficulty comparison between the DW1 and akmaxsat, where we rank the difficulty of each instance from 1 (easiest) to 500 (hardest) for both solvers. Perfect correlation would then be evidenced as all instances falling on the 45° diagonal. While for $\alpha = 0.1$ there is a slight tendency for clustering of the instances along this diagonal, such evidence of correlation is entirely absent for all higher values of the clause density.

If it is true that QA offers an advantage for certain ensembles of problems, then we also expect that for the random ensemble (which is, in a sense, a uniform average over possible ensembles), we should see specific problem instances for which QA offers an advantage over classical solvers. That we see no correlation in problem difficulty between QA and one particular classical solver is an interesting, though not conclusive, piece of evidence in this direction.

## 5. Conclusion and future work

In this work we undertook a study of the performance of the DW1 QA processor on MAX 2-SAT problems, and compared it to akmaxsat, a competitive exact classical solver. We focused on the random problem ensemble characterized by a fixed clause density, as the latter is a well defined external parameter which allows 'tuning' the problem hardness. After showing how MAX 2-SAT problems can be mapped to the DW1, we studied three main experimental questions:

(i) Is there experimental evidence for a hardness transition at or near the critical clause density?

(ii) Is there a correlation between the DW1 and akmaxsat in terms of problem hardness?

(iii) How does the time to solution scale for the two solvers, and is there evidence of a significant difference?

Our answer to the first question is a qualified 'yes'. Within the limitations of our relatively small number of variables we did indeed find evidence for a significant decrease in DW1 success probability starting around the critical clause density. Moreover, we found that the width of the finite-size scaling window follows the theoretical expectation. This suggests that QA is sensitive to the most essential feature of problem hardness. Our answer to the second question is a resounding 'no'. This means that within the ensemble of hard random MAX 2-SAT problems there are likely to be found problems for which QA has an advantage over exact classical solvers, and vice versa. However, one cannot exclude on the basis of our data that there might exist strong correlations between QA as encapsulated by the DW1 and stochastic classical solvers. Indeed, [18] showed that in terms of random spin glass problem instance hardness, *simulated* QA correlates very well with the DW1 (essentially as well as the DW1 correlates with itself). Since simulated QA has an efficient classical implementation using quantum Monte Carlo algorithms (by mapping to a classical spin problem in one extra dimension), it is undoubtedly desirable to follow up our work with a simulated QA study of the same set of MAX 2-SAT problem instances. This remark and more applies also to the third question: we found that the DW1 scaling with problem size is clearly better than `akmaxsat`'s over the range of problem sizes and clause densities we studied, and this is encouraging for QA, but at the same time additional study with classical stochastic solvers such as SA are needed in order to establish whether the DW1 advantage persists. This would also serve to test the idea that the DW1 is able to make coherent 'cluster moves' that simultaneously solve a subset of non-overlapping clauses, as discussed in section 4.3.2.

There are several other interesting directions for future research (see also [18]). We focused exclusively on the probability of finding the actual ground state, meaning that even a single-qubit error disqualifies a state as a correct solution; this criterion could be relaxed and one could instead focus on the distribution of excited states or Hamming distances from the ground state. The connection between problem hardness and the minimum energy gap between the ground and lowest excited state encountered during the annealing evolution is another question of great interest. Finally, it is obviously interesting to extend the results presented here to larger problem sizes and clause densities using the DW2 processor and its successors.

We conclude with a suggestion for future research that is related to the $\rho$-PTAS discussed in section 2.3, which highlights a particularly interesting aspect of MAX 2-SAT. Recall that a $\rho$-PTAS is an algorithm that provides an assignment of variables that provably satisfies a number of clauses within at least a fraction $\rho$ of the maximum number of clauses that can be satisfied for any formula, and that no $\rho$-PTAS exists for $\rho > 21/22$ unless P = NP [57]. This is a rather tight bound and it is tempting to try to probe it empirically. Of course an experiment cannot satisfy the conditions of rigorous proof required by the definition of a $\rho$-PTAS, but suppose the data is interpreted as a means to estimate an empirical $\rho$, as follows: when the processor finds an incorrect solution one counts the number of satisfied clauses $n_e$ for the excited state it found and compares to the correct solution for that instance, i.e., the true maximal number of clauses $n_t$ that can be satisfied. The ratio $\rho' = n_e/n_t$ is the empirical $\rho$ for that instance. One can then analyze the distribution of empirical $\rho$ values over all instances, and compare it to existing classical bounds. Note that $\rho'$ cannot be used in a straightforward manner to infer anything about the P versus NP question, since even if $\rho' > 21/22$ the inapproximability result states that this violates P $\neq$ NP only if the inequality can be achieved in poly $(N)$ time. Even if we find that $\rho'$ appears to be constant as $N$ increases, it could be that we had not picked the 'worst case'

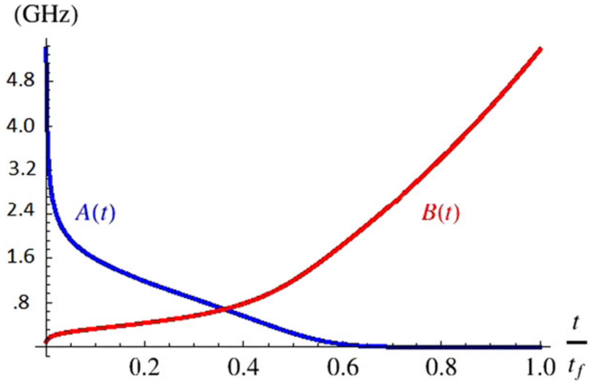

**Figure A1.** The experimental annealing schedules $A(t)$ and $B(t)$ appearing in the system Hamiltonian $H(t)$ (equation (3)).

distribution, and if we had, it would reduce $\rho'$ below 21/22 (at least asymptotically). Still, we believe this is an interesting question. We suspect that instances near $\alpha_c$ are 'hard to approximate', and if one considers the output of the best $\rho$-PTAS available [56], $\rho'$ for random instances will be distributed in the range [0.94, 1], while we expect that QA will yield a distribution of $\rho'$ values peaked closer to 1. The question for the future is then whether this may be used to infer anything about the asymptotic computational efficiency of QA.

## Acknowledgements

We thank Sergio Boixo, Amit Choubey, Matthias Troyer and Zhihui Wang for useful discussions and valuable input. The authors gratefully acknowledge funding by the Lockheed Martin Corporation, by ARO-MURI grant W911NF-11-1-0268, and by ARO-QA grant number W911NF-12-1-0523.

## Appendix A. Experimental settings

Our experiments were performed using the D-Wave One Rainier processor at the USC Information Sciences Institute, comprising 16 unit cells of eight superconducting flux qubits each, with a total of 108 functional qubits. The couplings are programmable superconducting inductances. Figure 1 is a schematic of the device, showing the allowed couplings between the qubits which form a 'Chimera' graph [44, 45]. The qubits and unit cell, readout, and control have been described in detail elsewhere [14, 65, 66]. The processor performs a QA protocol to find the ground state of a classical Ising Hamiltonian, as described by the transverse Ising Hamiltonian in equation (3). The initial energy scale for the transverse field is 5.36 GHz (the $A$ function in figure A1), ensuring that the initial state is to an excellent approximation a uniform superposition in the computational basis, with any deviations mainly due to control errors resulting in non-uniformity in the values of the local transverse fields. The final energy scale for the Ising Hamiltonian (the $B$ function) is 5.35 GHz, about 15 times the experimental temperature of 17 mK $\approx$ 0.37 GHz.

We performed 100 runs for each problem instance on the DW1. Each run returns a state measured in the computational basis (eigenvectors of $\sigma^z$), i.e., a proposed solution. Applying $H_P$

as given in equation (9) to this state yields the corresponding energy. The success probability, $p(\alpha, N)$, is defined as fraction of times (out of 100) the measured state is the ground state, i.e., is the correct solution as verified against the guaranteed-correct solution returned by `akmaxsat`.

We used default settings for the DW1, including programming and thermalization times of 1ms, and an annealing time of 1ms, which we did not attempt to optimize. Moreover, we did not average over different choices of subsets of active qubits, nor did we consider different 'gauges', i.e., reassignments of qubit up/down values which leave the spectrum invariant [17]. Thus our results may have been affected by systematic flux and coupler biases. However, removing such biases via averaging and gauges would have only improved the DW1 performance, so that our results can be viewed as lower performance bounds.

## Appendix B. Maximum possible value of the clause density

What is the maximum $\alpha$ allowing for a meaningful ensemble of problems to be generated?

Consider a single edge between two nodes $x_1$, $x_2$. Then the largest formula we can generate has 4 clauses:

$$F = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2) \tag{B.1}$$

Of course this problem instance is UNSAT, but is in principle allowed. If we only allow $M = 3$ clauses then in the example above there are $\binom{N}{M}$ possible formulae.

More generally, given a total of $C$ edges between a total of $N$ variables, we can use each edge to generate 4 clauses as in equation (B.1), and thus have one unique formula with $M_{\max} = 4C$ clauses. Again, this particular formula will be UNSAT, but is allowed.

Now, for any $n \leqslant N$ we can always choose a subgraph (of the Chimera graph) that has $c$ edges between the $n$ nodes and contributes up to 4 unique clauses. Thus $\alpha_{\max}(n) = 4c/n$, but for any $n \leqslant N$ there are $\binom{N}{n}$ choices of the subgraph, each of which might have a different number of edges. We claim that the ratio $c/n$ is maximized for $n = N$ which lets us use all $c = C$ edges. To see this suppose we start with all $N = 108$ qubits and $C = 255$ edges. Removing any qubit would result in the loss of at most six edges, and five edges on average given the actual connectivity of the Chimera graph. Thus we should check whether $(C - 5)/(N - 1) < C/N$, which holds at $N = 108$ and $C = 255$. Also, for any number of qubits $n < N$ and edges $c < 255$ between them, we can check that the same inequality holds: $(c - 5)/(n - 1) < c/n$. This means that the ratio $c/n$ increases with $n$ and thus will be maximal at the end point $n = N = 108$.

This establishes that the maximum possible clause density is $\alpha^{*}_{\max} = 4C/N$ corresponding to one unique UNSAT problem.

However, a unique formula is unsuitable when an ensemble is desired, as in our case. Thus consider clause densities $3C/N < \alpha < 4C/N$. For formulae with $\alpha$ in this range we note that there will necessarily be a combination of the type of equation (B.1). This can be proven by the pigeon hole principle: Let $\alpha = \frac{3C+c}{N}$. If we choose three possible clauses from each of the $C$ edges the formula may still be SAT; however we may choose the remaining $c$ clauses only from the unchosen remaining clauses from $c$ edges. This means that the problem is guaranteed to be UNSAT.

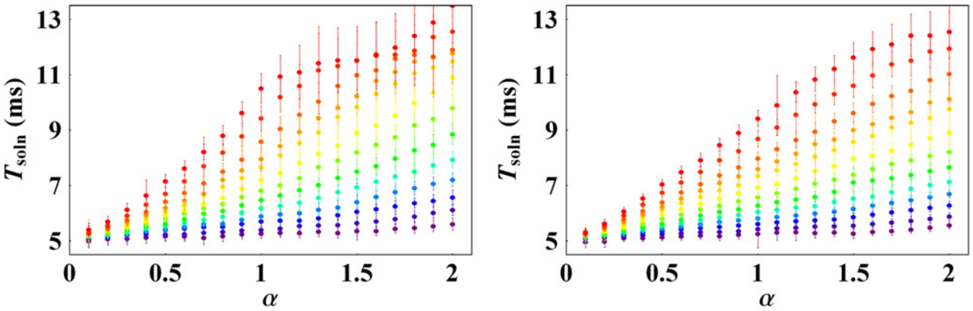

**Figure D1.** Solution time vs $\alpha$ for `akmaxsat` using random (left) and processor-compatible (right) instances for all $N$ values, increasing from bottom (purple, $N = 16$) to top (red, $N = 108$).

This brings us to the range $\alpha < 3C/N$. In this range we can have ensembles with $\binom{4C}{3C-a}$ unique formulae corresponding to problems with $\alpha = \frac{3C-a}{N}$. In practice we chose $\alpha_{\max} = 2$, which guaranteed that some of our problem instances were SAT and that we had large enough ensembles (at least 500 instances for each value of $\alpha$ and $N$).

## Appendix C. Parameter settings for `akmaxsat`

The `akmaxsat` program was run on a Quad-Core AMD Opteron™ Processor 1389 with a clock speed of 2.89 GHz. We chose the version of the algorithm which specifies an initial upper bound of infinity on the solution time as opposed to utilizing the UBCSAT stochastic local search solver to obtain an initial upper bound on the solution. Examining the scaling of the solution time as a function of $N$ for $\alpha = 2$, we found that an initial upper bound of infinity results in shorter mean optimal solution times and a more favorable scaling with $N$. Similar results were obtained when varying $\alpha$ for $N = 108$. Both comparisons utilized UBCSAT with an Iterated Robust TABU Search (IRoTS) of 10 iterations.

## Appendix D. `akmaxsat` on random versus DW1-compatible ensembles

We compared the performance of `akmaxsat` on unconstrained versus DW1 processor-compatible ensembles in section 4.1 of the main text. Figure D1 presents the data. Solution times are slightly larger for the unconstrained random ensemble $\mathcal{E}(N, \alpha)$ than for the constrained one $\mathcal{E}_{\text{DW}}(N, \alpha)$, with a noticeable jump at $\alpha = 1$ in the former, but not in the latter. This suggests that the constrained ensemble might remove some of the hardest problems, which could be understood as a consequence of the maximum degree 6 of the Chimera graph.

In figure D2, we plot the time to solution for `akmaxsat` for fixed $M$ values, for the unrestricted ensemble $\mathcal{E}(N, \alpha)$ and the DW1-compatible ensemble $\mathcal{E}_{\text{DW}}(N, \alpha)$. It can be seen that indeed, `akmaxsat` solution times do not seem to converge to a constant as $N$ increases, and only a mild improvement is seen as $M$ decreases.

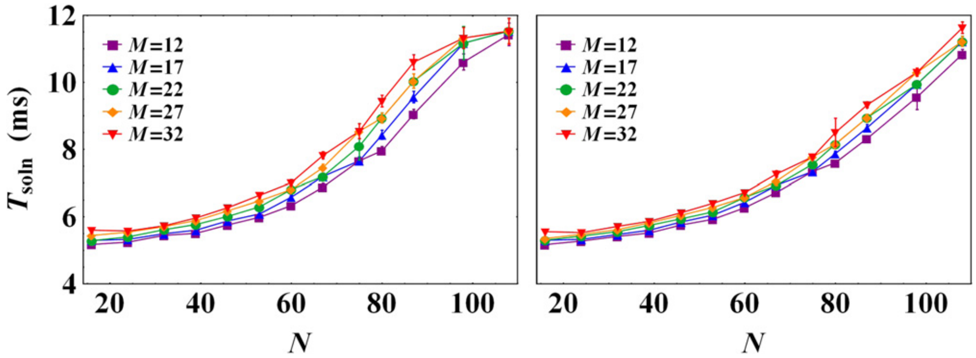

**Figure D2.** Solution time versus $N$ for fixed $M$ for the unrestricted (left) and DW1 processor-compatible (right) ensembles.

## Appendix E. Rescaling the parameter values

The DW1 processor is programmed via a user interface by specifying the qubits and the values of the local fields $h_j$ and couplers $J_{ij}$ as floating point numbers to three bits of precision. The allowed ranges are $h_j \in [-2, 2]$ and $J_{ij} \in [-1, 1]$. When values outside this range are specified, and the 'autoscale' function is used, as in our case, the DW1 rescales all local fields and couplers by max $\left\{ |h_j|/2, |J_{ij}| \right\}$. This means that there is a global factor in front of the problem Hamiltonian which effects the overall energy scale of the system. It is conceivable that larger scale factors reduce the gap and increase the susceptibility to thermal excitations. In addition, control errors result in Gaussian distributed values of the local fields and couplers with respective standard deviations of ~5% and ~10%, i.e., $h_j^{\text{implemented}} = h_j^{\text{specified}} \pm 0.1$ and $J_{ij}^{\text{implemented}} = J_{ij}^{\text{specified}} \pm 0.1$ [67].

Consider a formula $F$ with $M$ clauses, $N$ variables and let us focus on the variable $x_1$ that appears the most, $n_1$ times, in the formula. After we convert the clauses in the formula to local terms in the problem Hamiltonian, suppose that from each of the $n_1$ clauses that $x_1$ participates in, the field contributions are $h_i$, with $i \in \{1, 2, \dots, n_1\}$. The value of the local field for $x_1$, $h_1^F$, corresponding to $F$ is:

$$h_1^F = h_1 + h_2 + \cdots + h_{n_1}. \tag{E.1}$$

Now we note that $n_1 \leqslant 24$, since the Chimera graph degree of $x_1$ is $\leqslant 6$ and for each edge $x_1$ can appear in at most 4 clauses. But if $n_1 = 24$ then $h_1^F = 0$ because it appears negated the same number of times as unnegated. The local field of $x_1$ is maximized if it appears in clauses unnegated for all the couplings that participate in the formula. This means that $n_1 = 2 \times 6$ with $h_1 = h_2 = \cdots = h_{n_1} = +1$, and the maximum possible local field for any of our problems is $h_1^F = 12$. Thus rescaling involve division by at most 6, resulting in fractional coupler and field values, and in such cases the uncertainty of 0.1 in setting the couplers and local fields could have caused the 2SAT formula to be unfaithfully represented. Such cases probably contributed to lowering the success probability, simply because the 'wrong' problem was solved by the DW1 processor. However, *a priori* for our ensemble of problems the situation is somewhat

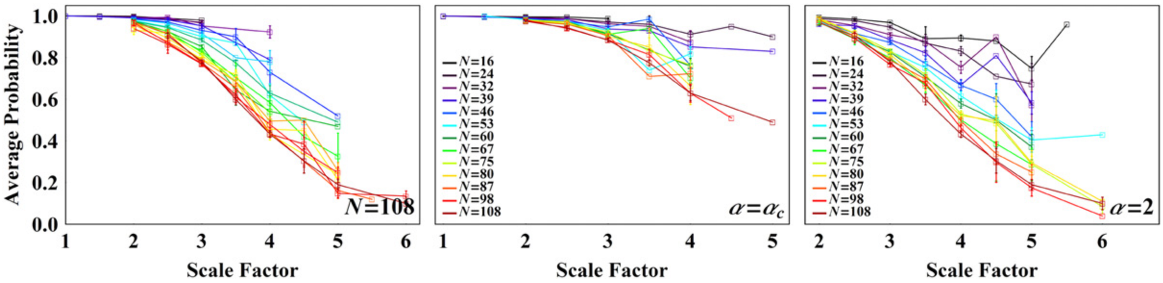

**Figure E1.** Left: probability of obtaining the correct solution for DW1 versus the scale factor determined by max ($|h_i|/2$, $|J_{ij}|$) for $N$ = 108 and clause densities ranging from $\alpha$ = 0.1 (black, top) to $\alpha$ = 2.0 (red, bottom) in increments of 0.1. Middle: probability of obtaining the correct solution for DW1 versus the scale factor for $\alpha$ = 1.0. Right: the same for $\alpha$ = 2.0.

better, since we imposed a maximum value of $\alpha = 2$ which means that, on an average, each qubit participated in two clauses and the maximum local field strength is 2, where rescaling is not required.

To investigate the contribution of such control errors we plot in the left panel of figure E1 the mean success probability at $N = 108$, and for all values of $\alpha$, as a function of the scale factor required to force all local field and coupler values into their allowed ranges. The scale factor is seen to significantly impact the success probability, with an impact that grows with the clause density. At the high end of $\alpha$ values the decrease in success probability is roughly linear with the scale factor. The middle and right panels figure E1 show the impact of the scale factor at $\alpha$ = 1.0 and 2.0 for all values of $N$. From this perspective too it is seen that the larger the scale factor the smaller the success probability, an effect that increases with $N$. Thus rescaling, which results in the ±0.1 uncertainty in fields and couplings becoming important, explains part of the reduction in the experimental success probability.

## Appendix F. Data collapse analysis

In figure 12 we display best fits for the solution time as a function of problem size for a range of clause densities. Figure F1 shows the complete data set, where the fit parameters are determined via an initial least squares fit and then a data collapse analysis. As a result, we find that for `akmaxsat` and DW1 $\gamma = 3/4$ and $\gamma = 3/2$, respectively, correspond to a strong collapse of the data particularly for $\alpha \geqslant \alpha_c$.

To complement the fit given in equation (13) and table 1 which use a restricted set of fitting parameters, we present a second, less constrained fit, which includes a purely $\alpha$-dependent part:

$$T_{\rm soln}(\alpha, N) = A\ \exp\left(B\alpha^{\gamma}N^{\delta}\right) + C\ \exp\left(D\alpha^{\zeta}\right) + E\alpha + F. \tag{F.1}$$

We find the results given in table F1, and the fit is shown in figure F2. Note that in this case we did not use a data collapse. The coefficient of determination for the `akmaxsat` data is identical to that for the first fit (table 1), while this coefficient improves from 0.991 for the DW1 data, so the second fit is slightly better. However, this improvement comes at the expense of

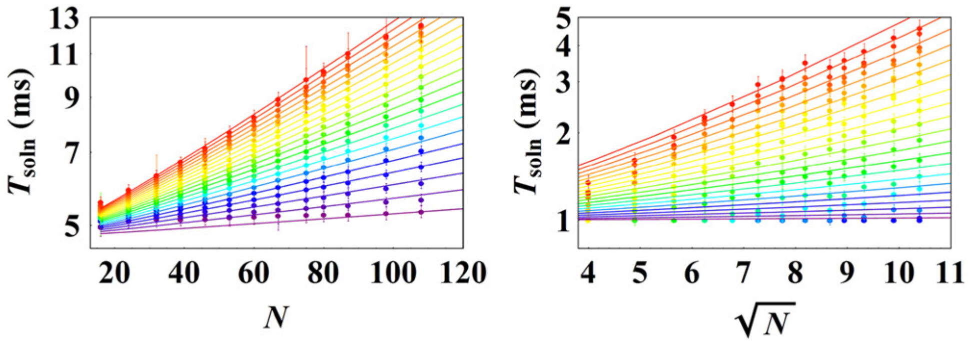

**Figure F1.** Solution time vs $N$ for `akmaxsat` (left) and DW1 (right). Color variations denote $\alpha$ from $\alpha = 0.1$ (purple, bottom) to $\alpha = 2.0$ (red, top) with corresponding best fits. The fit shown is the one in equation (13) with the fit parameters resulting from the data collapse.

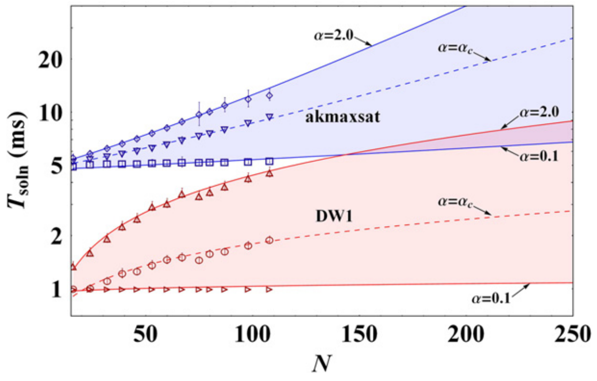

**Figure F2.** Computational effort comparison between `akmaxsat` and the DW1 as a function of the problem size $N$ and for different clause densities $\alpha$. Same as in figure 12 except that the fit is given by equation (F.1) with parameter values from table F1.

**Table F1.** Numerical values obtained from a least squares fit of the data shown in figure 12 for both `akmaxsat` and the DW1. The parameters are the ones appearing in equation (F.1).

| | $\gamma$ | $\delta$ | $\zeta$ | $B$ | $D$ |
|---|---|---|---|---|---|
| AK | 0.673 [1] | 1.143 [3] | 0.188 [7] | 0.003 [2] | 0.188 [7] |
| DW1 | 1.589 [6] | 0.637 [8] | 0.003 [0] | 0.003 [0] | 2.368 [7] |
| | $A$ | $C$ | $E$ | $F$ | $R^2$ |
| AK | 4.537 [4] | 0.292 [0] | 0.097 [4] | 0.000 [6] | 0.999 [6] |
| DW1 | 22.943 [9] | −2.079 [1] | −0.270 [0] | 0.045 [5] | 0.996 [9] |

introducing many more free parameters. Note that again $\delta - \gamma$ is positive for `akmaxsat` and negative for the DW1 (recall the discussion in section 4.3.2).

## References

[1] Farhi E, Goldstone J, Gutmann S and Sipser M 2000 arXiv:quant-ph/0001106
[2] Farhi E, Goldstone J, Gutmann S, Lapan J, Lundgren A and Preda D 2001 *Science* **292** 472
[3] Aharonov D, van Dam W, Kempe J, Landau Z, Lloyd S and Regev O 2007 *SIAM J.Comp.* **37** 166
[4] Kempe J, Kitaev A and Regev O 2006 *SIAM J. Comp.* **35** 1070–97
[5] Siu M S 2005 *Phys. Rev.* A **71** 062314
[6] Oliveira R and Terhal B 2005 *Quantum Inf. Comput.* **8** 0900
[7] Mizel A, A Lidar D and Mitchell M 2007 *Phys. Rev. Lett.* **99** 70502
[8] Childs A M, Farhi E and Preskill J 2001 *Phys. Rev.* A **65** 012322
[9] Sarandy M S and Lidar D A 2005 *Phys. Rev. Lett.* **95** 250503
[10] Roland J and Cerf N J 2005 *Phys. Rev.* A **71** 032330
[11] Tiersch M and Schützhold R 2007 *Phys. Rev.* A **75** 062313
[12] Amin M H S, Averin D V and Nesteroff J A 2009 *Phys. Rev.* A **79** 022107
[13] Amin M H S, Love P J and Truncik C J S 2008 *Phys. Rev. Lett.* **100** 060503
[14] Harris R *et al* 2010 *Phys. Rev.* B **82** 024511
[15] Johnson M W *et al* 2011 *Nature* **473** 194–8
[16] Dickson N G *et al* 2013 *Nat. Commun.* **4** 1903
[17] Boixo S, Albash T, Spedalieri F M, Chancellor N and Lidar D A 2013 *Nat. Commun.* **4** 3067
[18] Boixo S *et al* 2013 arXiv:1304.4595
[19] Wang L *et al* 2013 arXiv:1305.5837
[20] Brooke J, Bitko D, Rosenbaum F T and Aeppli G 1999 *Science* **284** 779–81
[21] Finnila A B, Gomez M A, Sebenik C, Stenson C and Doll J D 1994 *Chem. Phys. Lett.* **219** 343–8
[22] Kadowaki T and Nishimori H 1998 *Phys. Rev.* E **58** 5355–63
[23] Santoro G E, Martoňák R, Tosatti E and Car R 2002 *Science* **295** 2427–30
[24] Das A and Chakrabarti B K 2008 *Rev. Mod. Phys.* **80** 1061–81
[25] Young A P, Knysh S and Smelyanskiy V N 2008 *Phys. Rev. Lett.* **101** 170503
[26] Martoňák R, Santoro G E and Tosatti E 2002 *Phys. Rev.* B **66** 094203
[27] Battaglia D A, Santoro G E and Tosatti E 2005 *Phys. Rev.* E **71** 066707
[28] Morita S and Nishimori H 2008 *J. Math. Phys.* **49** 125210
[29] Farhi E, Gosset D, Hen I, Sandvik A W, Shor P, Young A P and Zamponi F 2012 *Phys. Rev.* A **86** 052334
[30] Garcia-Ripoll J J and Bañuls 2008 arXiv:0812.1760
[31] Kirkpatrick S, Gelatt C D and Vecchi M P 1983 *Science* **220** 671–80
[32] Mezard M and Montanari A 2009 *Information Physics and Computation* (New York: Oxford University Press)
[33] Coppersmith D, Gamarnik D, Hajiaghayi M and Sorkin G B 2004 *Random Struct. Algorithms* **24** 502–45
[34] de la Vega W F 2001 *Theor. Comput. Sci.* **265** 131–46
[35] McGeoch C C and Wang C 2013 Experimental evaluation of an adiabatic quantum system for combinatorial optimization *Proc. 2013 ACM Conf. on Computing Frontiers* (New York: ACM) p 23
[36] Selby A 2013 *D-Wave: Comment on Comparison with Classical Computers* www.archduke.org/stuff/d-wave-comment-on-comparison-with-classical-computers/
[37] Puget J F 2013 *D-Wave vs CPLEX Comparison. Part 2: QUBO* **48** 45 www.ibm.com/developerworks/community/blogs/jfp/entry/dwavevscplexcomparisonpart2qubo
[38] Jordan S P, Farhi E and Shor P W 2006 *Phys. Rev.* A **74** 052322
[39] Lidar D A 2008 *Phys. Rev. Lett.* **100** 160506
[40] Quiroz G and Lidar D A 2012 *Phys. Rev.* A **86** 042333
[41] Paz-Silva G A, Rezakhani A T, Dominy J M and Lidar D A 2012 *Phys. Rev. Lett.* **108** 080501
[42] Young K C and Sarovar M 2012 arXiv:1208.6371

[43] Kuegel A 2012 Improved exact solver for the weighted Max-SAT problem *POS-10* (*EPiC Series* vol 8) ed Berre D L (Manchester: EasyChair) pp 15–27

[44] Choi V 2008 *Quantum Inf. Proc.* **7** 193–209

[45] Choi V 2011 *Quantum Inf. Proc.* **10** 343–53

[46] Rendl F, Rinaldi G and Wiegele A 2010 *Math. Program.* **121** 307–35

[47] Spin glass server www.informatik.uni-koeln.de/spinglass/

[48] Dechter R 1999 *Artif. Intell.* **113** 41–85

[49] Aspvall B, Plass M F and Tarjan R E 1979 *Inf. Proc. Lett.* **8** 121–3

[50] Papadimitriou C 1995 *Computational Complexity* (Reading, MA: Addison Wesley Longman)

[51] Bollobás B, Borgs C, Chayes J T, Kim J H and Wilson D B 2001 *Random Struct. Algorithms* **18** 201–56

[52] Goerdt A 1996 *J. Comput. System Sci.* **53** 469

[53] Monasson R, Zecchina R, Kirkpatrick S, Selman B and Troyansky L 1999 *Nature* **400** 133–7

[54] Shen H and Zhag H 2003 *Electron. Notes Discrete Math.* **16** 80–92

[55] Goemans M and Williamson D 1995 *J. Assoc. Comput. Mach.* **42** 1115–45

[56] Lewin M, Livnat D and Zwick U 2002 *Integer Program. Comb. Optim.* **2337** 67–82

[57] Hàstad J 2001 *J. ACM* **48** 798–859

[58] 2013 *Annual Max-SAT Evaluations* http://maxsat.ia.udl.cat/introduction/

[59] Davis M and Putnam H 1960 *J. ACM* **7** 201–15

[60] Feigenbaum J and Fortnow L 1993 *SIAM J. Comp.* **22** 994–1005

[61] Ajtai M 1999 Generating hard instances of the short basis problem *Proc. XXVI Int. Colloq. on Automata, Languages and Programming (ICAL '99)* ed J Wiedermann, P van Emde Boas and M Nielsen (London: Springer) pp 1–9 http://dl.acm.org/citation.cfm?id=646229.681554

[62] Austrin P 2007 Balanced MAX 2-SAT might not be the hardest *Proc. 39th Ann. ACM Symp. on Theory of Computing (STOC '07)* (New York: ACM) pp 189–97

[63] Albash T, Boixo S, Lidar D A and Zanardi P 2012 *New J. Phys.* **14** 123016

[64] Amin M H, Dickson N G and Smith P 2013 *Quantum Inf. Proc.* **12** 1819–29

[65] Berkley A J, Johnson M W, Bunyk P, Harris R, Johansson J, Lanting T, Ladizinsky E, Tolkacheva E, Amin M H S and Rose G 2010 *Supercond. Sci. Technol.* **23** 105014

[66] Johnson M W *et al* 2010 *Supercond. Sci. Technol.* **23** 065004

[67] Lanting T 2013 *D-Wave Inc.* private communication