Assessing Solution Quality of 3SAT on a Quantum Annealing Platform

Thomas Gabor[1(✉)], Sebastian Zielinski[1], Sebastian Feld[1], Christoph Roch[1], Christian Seidel[2], Florian Neukart[3], Isabella Galter[2], Wolfgang Mauerer[4], and Claudia Linnhoff-Popien[1]

[1] LMU Munich, Munich, Germany
thomas.gabor@ifi.lmu.de
[2] Volkswagen Data:Lab, Munich, Germany
[3] Volkswagen Group of America, San Francisco, USA
[4] OTH Regensburg/Siemens Corporate Research, Regensburg, Germany

Abstract. When solving propositional logic satisfiability (specifically 3SAT) using quantum annealing, we analyze the effect the difficulty of different instances of the problem has on the quality of the answer returned by the quantum annealer. A high-quality response from the annealer in this case is defined by a high percentage of correct solutions among the returned answers. We show that the phase transition regarding the computational complexity of the problem, which is well-known to occur for 3SAT on classical machines (where it causes a detrimental increase in runtime), persists in some form (but possibly to a lesser extent) for quantum annealing.

Keywords: Quantum computing · Quantum annealing · D-wave · 3SAT · Boolean satisfiability · NP · Phase transition

1 Introduction

Quantum computers are an emerging technology and still subject to frequent new developments. Eventually, the utilization of intricate physical phenomena like superposition and entanglement is conjectured to provide an advantage in computational power over purely classical computers. As of now, however, the first practical breakthrough application for quantum computers is still sought for. But new results on the behavior of quantum programs in comparison to their classical counterparts are reported on a daily basis.

Research in that area has cast an eye on the complexity class NP: It contains problems that are traditionally (and at the current state of knowledge regarding the P vs. NP problem) conjectured to produce instances too hard for classical computers to solve exactly and deterministically within practical time constraints. Still, problem instances of NP are also easy enough that they can be executed efficiently on a (hypothetical) non-deterministic computer.

S. Feld and C. Linnhoff-Popien (Eds.): QTOP 2019, LNCS 11413, pp. 23–35, 2019.
https://doi.org/10.1007/978-3-030-14082-3_3

The notion of computational complexity is based on classical computation in the sense of using classical mechanics to describe and perform automated computations. In particular, it is known that in this model of computation, simulating quantum mechanical systems is hard. However, nature itself routinely "executes" quantum mechanics, leading to speculations [20] that quantum mechanics may be used to leverage greater computational power than systems adhering to the rules of classical physics can provide.

Quantum computing describes technology exploiting the behavior of quantum mechanics to build computers that are (hopefully) more powerful than current classical machines. Instead of classical bits $b \in \{0, 1\}$ they use qubits $q = \alpha |0\rangle + \beta |1\rangle$ where α, β, $|\alpha|^2 + |\beta|^2 = 1$, are probability amplitudes for the basis states $|0\rangle, |1\rangle$. Essentially, a qubit can be in both states 0 and 1 at once. This phenomenon is called superposition, but it collapses when the actual value of the qubit is measured, returning either 0 or 1 with a specific probability and fixing that randomly acquired result as the future state of the qubit. Entanglement describes the effect that multiple qubits can be in superpositions that are affected by each other, meaning that the measurement of one qubit can change the assigned probability amplitudes of another qubit in superposition. The combination of these phenomena allows qubits to concisely represent complex data and lend themselves to efficient computation operations.

The technological platform of quantum annealing is (unlike the generalized concept of quantum computing) not capable of executing general quantum-mechanical computations, but is within current technological feasibility and available to researchers outside the field of quantum hardware. The mechanism specializes in solving optimization problems and can (as a trade-off) work larger amounts of qubits in a useful way than current quantum-mechanically complete platforms.

In this paper, we evaluate the performance of quantum annealing (or more specifically, a D-Wave 2000Q machine) on the canonical problem of the class NP, propositional logic satisfiability for 3-literal clauses (3SAT) [14]. As we note that there is still a remarkable gap between 3SAT instances that can be put on a current D-Wave chip and 3SAT instances that even remotely pose a challenge to classical solvers, there is little sense in comparing the quantum annealing method to classical algorithms in this case (and at this early point in time for the development of quantum hardware). Instead, we are interested in the scaling behavior with respect to problem difficulty. Or more precisely: We analyze if and to what extent quantum annealing's performance suffers under hard problem instances (like classical algorithms do).

We present a quick run-down of 3SAT and the phenomenon of phase transitions in Sect. 2 and continue to discuss further related work in Sect. 3. In Sect. 4 we describe our experimental setup and then present the corresponding results in Sect. 5. We conclude with Sect. 6.

2 Preliminaries

Propositional logic satisfiability (SAT) is the problem of telling if a given formula in propositional logic is satisfiable, i.e., if there is a assignment to all involved Boolean variables that causes the whole formula to reduce to the logical value *True*. As such, the problem occurs at every application involved complex constraints or reasoning, like (software) product lines, the tracing of software dependencies or formal methods.

It can be trivially shown that (when introducing a linear amount of new variables) all SAT problems can be reduced to a specific type of SAT problem called 3SAT, where the input propositional logic formula has to be in conjunctive normal form with all of the disjunctions containing exactly three literals.

For example, the formula $\Psi = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$ is in 3SAT form and is satisfiable because the assignment $(x_1 \mapsto \mathit{True}, x_2 \mapsto \mathit{True}, x_2 \mapsto \mathit{True})$ causes the formula to reduce to *True*. The formula $\Phi = (x_1 \vee x_1 \vee x_1) \wedge (\neg x_1 \vee \neg x_1 \vee \neg x_1)$ is also in 3SAT form but is not satisfiable.

Definition (3SAT). A 3SAT instance with m clauses and n variables is given as a list of clauses $(c_k)_{0 \leq k \leq m-1}$ of the form $c_k = (l_{3k} \vee l_{3k+1} \vee l_{3k+2})$ and a list of variables $(v_j)_{0 \leq j \leq n-1}$ so that l_i is a literal of the form $l_i \in \bigcup_{0 \leq j \leq n-1} \{v_j, \neg v_j\}$. A given 3SAT instance is *satisfiable* iff there exists a variable assignment $(v_j \mapsto b_j)_{0 \leq j \leq n-1}$ with $b_j \in \{\mathit{True}, \mathit{False}\}$ so that $\bigwedge_{0 \leq k \leq m-1} c_k$ reduces to *True* when interpreting all logical operators as is common. The problem of deciding whether a given 3SAT instance is satisfiable is called 3SAT.

3SAT is of special importance to complexity theory as it was the first problem which was shown to be NP-complete [14]. This means that every problem in NP can be reduced to 3SAT in polynomial time. It follows that any means to solve 3SAT efficiently would thus give rise to efficient solutions for any problem in NP like graph coloring, travelling salesman or bin packing.

Despite the fact that for NP-complete problems in general no algorithm is known that can solve all problem instances of a problem efficiently (i.e., in polynomial time), it is within the scope of knowledge that "average" problem instances of many NP-complete problems, including 3SAT, are easy to solve [10]. In Ref. [37] this characteristic is described with a phase transition. The boundary of the phase transition divides the problem space into two regions. In one region, a solution can be found relatively easily, because the solution density for these problems is high, whereas in the other region, it is very unlikely that problems can contain a correct solution at all. Problems that are very difficult to solve are located directly at this phase boundary [10].

It can be observed that, with randomly generated 3SAT instances, the probability of finding a correct solution decreases abruptly when the ratio of clauses to variables $\alpha = m/n$ exceeds a critical value of α_c [36]. According to [35] this critical point is $\alpha_c \approx 4.267$ for randomly generated 3SAT instances. In the surrounding area of the critical point, finding a solution (i.e., deciding if the instance is satisfiable) is algorithmically complex. Figure 1 illustrates this phenomenon.

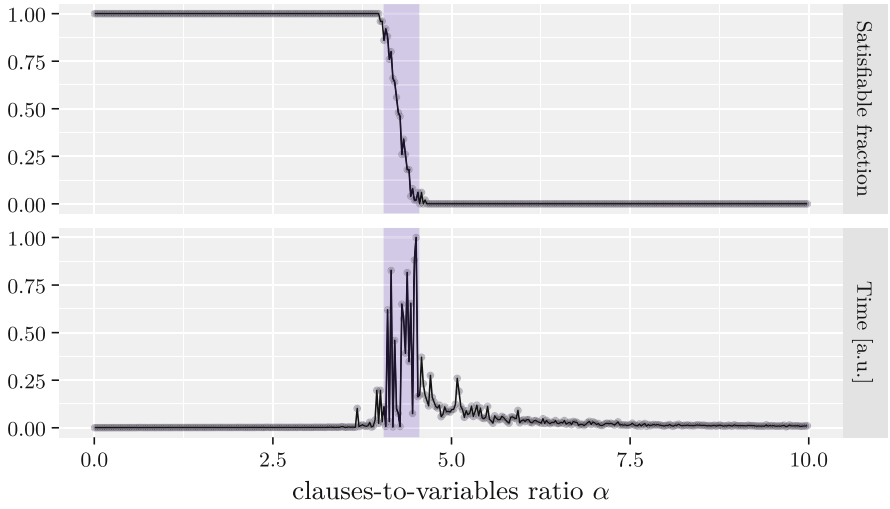

Fig. 1. Phase transition of 3SAT. The bottom plot shows the computational time required to determine satisfiability of randomly chosen 3SAT instances with a specific clauses-to-variables ratio α on a standard solver. The area around the critical point $\alpha_c \approx 4.267$ is shaded in blue. The upper portion shows the probability that instances with a particular ratio α are solvable. In the region around the critical point, it is hard to determine whether a problem instance can be fulfilled with a concrete allocation or not. (Color figure online)

To assess the solution quality of randomly generated 3SAT instances we generate instances in every complexity region. The results are discussed in Sect. 5.

3 Related Work

It is one of the cornerstones of complexity theory that solving NP-complete decision problems is strongly believed to be not efficiently possible [14,40]. Any NP-complete problem can also be cast as an optimization problem, which allows for employing well-known optimization algorithms to find approximate solutions—typical methods include tabu search [22,23] and simulated annealing [11,27]. Countless other efficient approximation methods, together with an elaborate taxonomy on approximation quality (how much does a given solution differ from a known global optimum?) and computational effort (how many time steps are required until an approximate solution that satisfies given quality goals is available?), have been devised [6].

An intriguing connection that has received substantial attraction exists between (computational) NP-complete problems and the (physical) concept of phase transitions, as detailed in Sect. 2. First investigations of the phenomenon have been performed by Kirkpatrick et al. [28]; Monasson et al. first suggested a connection between the type of phase transition and the associated computational costs of a problem [37]. From the abundant amount of more recent inves-

tigations, we would like to highlight the proof by Ding et al. [16] that establishes a threshold value for the phase transition. Our work benefits from the above insights by selecting the "most interesting", i.e., computationally hardest, scenarios as investigation target.

The idea of obtaining solutions for NPO (NP optimization) problems by finding the energy ground state (or states) of a quantum mechanical system was used, for instance, by Apolloni et al. [4,5] to solve combinatorial optimization problems. The general idea of quantum annealing has been independently rediscovered multiple times [2,3,21,26].

Quantum annealing techniques are usually applied to solving NP-complete decision problems, or optimization problems from class NPO. Lucas [30] reviews how to formulate a set of key NP problems in the language of adiabatic quantum computing, i.e., quadratic unconstrained binary optimization (QUBO). In particular, problems of the types "travelling salesman" or "binary satisfiability" that are expected to have a major impact on practical computational applications if they can be solved advantageously on quantum annealers have undergone a considerable amount of research [7,24,39,41,44,46]. Further effort has been made on combining classical and quantum methods on these problems [19].

Comparing the computational capabilities of classical and quantum computers is an intriguing and complex task, since the deployed resources are typically very dissimilar. For instance, the amount of instructions required to execute a particular algorithm is one of the main measures of efficiency or practicability on a classical machine, whereas the notion of a discrete computational "step" is hard to define on a quantum annealing device. Still, multiple efforts have been made towards assessing quantum speedup [42,43]. For the quantum gate model, a class of problems exhibiting quantum speedup has been found lately [9]. Interest in quantum computing has also spawned definitions of new complexity classes (e.g., [29,38]), whose relations to traditional complexity classes have been and are still subject to ongoing research [8,32].

These questions hold regardless of any specific physical or conceptual implementation of quantum computing since their overall computational capabilities are known to be largely interchangeable; for instance, McGeoch [33] discusses the equivalence of gate-based and adiabatic quantum computing. Consequently, our work focuses not on comparing quantum and classical aspects of solving particular problems, but concentrates on understanding peculiarities of solving one particular problem (3SAT, in our case) in-depth.

Formulating 3SAT problems on a quantum annealing hardware has been previously considered [12,13,18], and we rely on the encoding techniques presented there. Van Dam [45] and Farhi [17] have worked on analyzing the complexity of solving general 3SAT problems. Hsu et al. have considered the complexity-wise easier variation 2SAT as a benchmarking problem to compare various parameter configurations of their quantum annealer [25].

4 Experimental Setup

Quantum annealing is an optimization process that can be implemented in hardware. It is built upon the adiabatic theorem that provides conditions under which an initial ground-state configuration of a system evolves to the ground state of another configuration that minimizes a specific user-defined energy function [33]. As in the real world the required conditions for the theorem can only be approximated, the results of quantum annealing are not deterministically optimal but show a probabilistic distribution, ideally covering the desired optimal value.

D-Wave's quantum annealer is the first commercial machine to implement quantum annealing. Its interface is built on two equivalent mathematical models for optimization problems called Ising and QUBO, the latter of which will be used for the work of this paper. Quadratic Unconstrained Binary Optimization (QUBO) problems can be formulated as a quadratic matrix Q_{ij}. Quantum annealing then searches for a vector $x \in \{0,1\}^n$ so that $\sum_i \sum_{j<i} Q_{ij} x_i x_j + \sum_i Q_i x_i$ is minimal. The promise of quantum annealing is that—using quantum effects—specialized hardware architectures are able to solve these optimization problems much faster than classical computers in the future.

The main goal of this paper is to analyze the inherently probabilistic distribution of return values generated by quantum annealing when trying to solve hard optimization problems. We choose to demonstrate such an analysis on 3SAT because it is the canonical problem of the class NP, which is a prime target for research on performance improvements via quantum technology with respect to classical computers [30,34].

4.1 Defining 3SAT as a QUBO

3SAT is usually not formulated as an optimization problem (see Sect. 2), or defined by an equivalent QUBO problem, as is required by the annealer. Thus, we require a (polynomial-time) translation of any 3SAT instance into a QUBO so that the solutions generated by the quantum annealer can be translated back to solutions of the initial 3SAT instance.

Following [12,13], we translate 3SAT into the Weighted Maximum Independent Set (WMIS) problem and then translate the WMIS instance into a QUBO (we find it convenient to specify the polynomial coefficients in matrix form). We omit the details of this process and instead refer to *op. cit.* and Lucas [30]. However, we shall briefly discuss the implications of the translation process.

A 3SAT instance, that is, a formula with m clauses for n variables, requires a QUBO matrix of size $3m \times 3m$ with the solution vector $x \in \{0,1\}^{3m}$. The solution can be thought of as using a qubit for each literal in the initial formula and thus consisting of a triplet of qubits for each 3SAT clause. This usually means that we have much more qubits than variables in the formula. Nonetheless, a QUBO solution is mapped to a value assignment for the variables in the 3SAT formula. Thus, when running successfully, the quantum annealer will output a satisfying assignment for a given 3SAT formula. We can check if the assignment really is correct (i.e., each variable has a value assigned and the whole formula

reduces to *True*) using few instructions of classical computation. Obviously, if among several experimental runs the quantum annealer does return just one correct assignment, the corresponding 3SAT formula is satisfiable. If the quantum annealer only returns incorrect assignments, we will regard the formula as unsatisfiable (although the prove of that is only probabilistic).

There are some aspects to note about how the QUBO solution vectors are mapped to variable assignments. Given a QUBO solution vector $(x_i)_{0\leq i\leq 3m-1}$ for a 3SAT formula with literals $(l_i)_{0\leq i\leq 3m-1}$, a variable v is assigned the value *True* if it occurs in a literal $l_i = v$ and $x_i = 1$. Likewise, a variable v is assigned the value *False* if it occurs in a literal $l_i = \neg v$ and $x_i = 1$. It is important to note that $x_i = 0$ has *no implication* on the value of the variable in l_i.

Intuitively, we can interpret $x_i = 1$ to mean "use the value of l_i to prove the satisfaction of clause $c_{(i \mod 3)}$". From our QUBO optimization, we expect to find one (and only one) suitable l_i for every clause in the 3SAT formula.[1]

This is important as it opens up a wide range of different QUBO solutions which may just encode the exact same variable assignment at the 3SAT level. However, it also means that seemingly suboptimal QUBO solutions may encode correct 3SAT assignments. For example, consider the (a little redundant) 3SAT formula $(v_0 \vee v_1 \vee v_2) \wedge (v_0 \vee v_1 \vee v_2)$: The QUBO solution $x = 100001$ would imply the assignment of $v_0 = \mathit{True}$ and $v_2 = \mathit{True}$, which indeed is theoretically sufficient to prove the formula satisfiable. The exact same assignment would be implied by $x = 001100$. However, note that none of these imply a full assignment of every variable in the 3SAT instance since none say anything about the value of v_1. Still, we can trivially set v_1 to any arbitrary value and end up with a correct assignment. Also note that while the QUBO is built in such a way to opt for one single value 1 per triplet in the bit string, even bitstrings violating this property can encode correct solution. In our example, the suboptimal QUBO solution $x = 100000$ still encodes all necessary information to prove satisfiability.

4.2 Evaluating Postprocessing

As can be seen from the last example, postprocessing is an integral part of solving problems with quantum annealing. As discussed earlier in this section, we consider a QUBO solution correct, if it not only matches the expected structure for minimizing the QUBO energy function, but instead iff it directly implies a correct assignment in the definition given above. Thus, while the expected structure for QUBO optimizes x so that the amount bits x_i assigned 1 equals the amount of clauses m, we also consider less full answers correct.

On top of that, there are solutions that cannot be mapped to an assignment immediately, but still with almost no effort. We want to regard these as well and implemented a postprocessing step we call *logical postprocessing*. It is applied whenever none of the qubits corresponding to a single clause c_k are set to 1 by

[1] This intuition matches the concept of constructivism in logic and mathematics. We are not only looking for the correct answer, but are looking for a correct and complete proof of an answer, giving us a single witness for each part of the formula.

the quantum annealer and the respective QUBO solution is not already correct. In that case, we iterate through all literals l_i in that clause c_k and check if we could set $x_i = 1$ without contradicting any other assignment made within x. If we find such an l_i, we set $x_i = 1$ and return the altered bitstring x.

The software platform provided by D-Wave to use the quantum annealer already offers integrated postprocessing methods as well, which we will also empirically show to be more powerful than logical postprocessing in the following Sect. 5. Again, for greater detail we refer to the D-Wave documentation on that matter [15]. At a glance, the employed postprocessing method splits the QUBO matrix into several subproblems, tries to optimize these locally, and then integrates that local solution into the complete solution if it yields an improvement. We call this method *D-Wave postprocessing*.

To evaluate the solution quality regarding 3SAT, we employ both methods. The goal is to assess expected quality on a 3SAT-to-3SAT level, i.e., we measure how well we can solve the given 3SAT instance and regard the translation to and from QUBO as a mere technical problem that is not of interest for this paper.

5 Evaluation

To assess the solution quality of 3SAT on a quantum annealing platform, using the previously discussed method of encoding 3SAT problems, we ran several experiments on a D-Wave 2000Q system. Using ToughSAT[2] we generated 3SAT instances of various difficulty (i.e., with various values for α). However, as discussed in Sect. 2, for $|\alpha - 4.2| \gg 0$ problem instances become very easy to solve. We observed that effect on the quantum annealer as well, since *all* of these instances were easily (i.e. 100% of the time) solved on the D-Wave machine. Thus, for the remainder of this section, we focus on hard instances (approximated by $\alpha = 4.2$) to assess solution quality in the interesting problem domain.

Experiments have shown that using the standard embedding tools delivered with the D-Wave platform, we can only reliably find a working embedding on the D-Wave 2000Q chip for 3SAT instances with at most 42 clauses [1]. To maintain $\alpha \approx 4.2$, the generated 3SAT instances contain 10 different variables. We only assess solution quality for 3SAT instances that are satisfiable, but do not provide this information to the solver.

Figure 2 shows the result distribution of these runs on the D-Wave machine. On the x-axis, we sorted the returned results according to the bits that have been assigned the value 1 or *True*. As discussed in Sect. 4 the optimal solution is supposed to set one bit for each clause, i.e., is supposed to contain 42 bits set to *True*. However, as there are only 10 different variables, there theoretically exist answers that only set 10 bits but that still map to a complete and valid solution for the given 3SAT instance. From Fig. 2 we can see that some of these solution are found for bitcounts starting from 37 through 41. Interestingly, the complete range of answers gathered seems to follow a distribution centered around 37 or

[2] https://toughsat.appspot.com/.

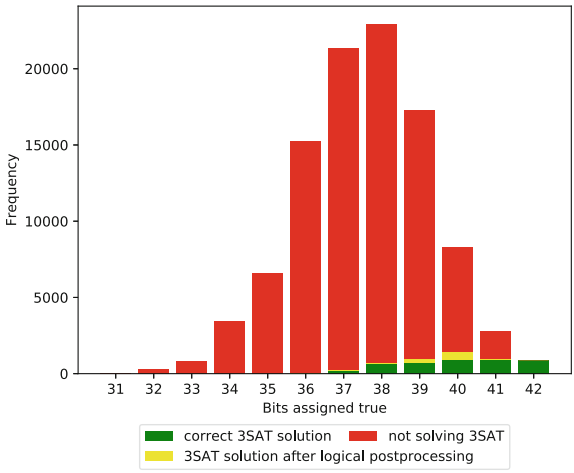

Fig. 2. Distribution of correct (green) and incorrect (red) answers returned by the quantum annealer *without D-WAVE postprocessing*. Answers that can trivially be transformed into valid answers using logical postprocessing are marked in yellow. The plot shows 100,000 answers in total for 100 different hard 3SAT instances ($\alpha \approx 4.2$). (Color figure online)

38 and no answers with more than 42 bits are returned. This means that the constraint of never setting multiple bits per clause is fully respected in the evaluation of our QUBO matrix. Note that although there are 5,283 correct solutions in total, these are only distributed across 24 of the 100 randomly generated problem instances. Thus, most of them have not been solved at all.

Furthermore, we applied the logical postprocessing described in Sect. 4 to the incorrect answers in Fig. 2. However, it shows little improvement on the total amount of correct answers collected. We expect the postprocessing method delivered with the D-Wave software package to be more powerful as it runs local search along more axes of the solution space than the logical postprocessing does. So we ran the complete evaluation experiment again, only this time turning on the integrated postprocessing. The results are shown in Fig. 3.

We observed that the D-Wave postprocessing managed to optimize all correct but "incomplete" answers, mapping them to a solution with 42 bits assigned the value *True*. Out of the 100,000 queries, this yielded 25,142 correct answers. Moreover, these correct answers span 99 of the 100 randomly generated 3SAT instances so that we consider the problem solved. Effectively, this shows that quantum annealing does suffer from a breakdown in expected solution quality at the point of the phase transition in the 3SAT problem. In comparison to the immense decrease in performance seen in classical solvers (cf. Sect. 2), a drop to around 25% precision (if it was to persist on larger chip sizes) appears rather desirable, though. A quick example: To achieve a $1 - 10^{-12}$ confidence of returning the correct answer our experimental setup requires around 97 queries.

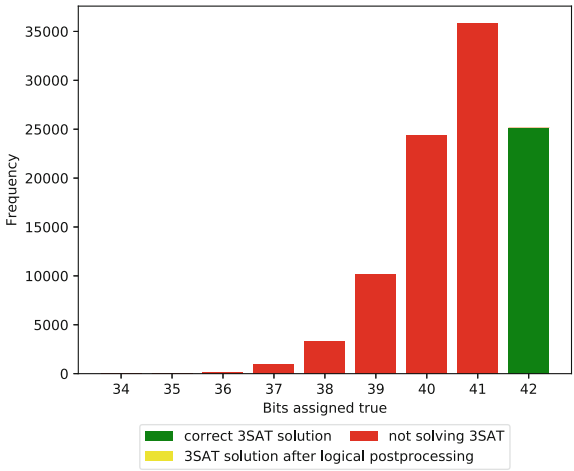

Fig. 3. Distribution of correct (green) and incorrect (red) answers returned by the quantum annealer *using D-WAVE postprocessing*. Answers that can trivially be transformed into valid answers using logical postprocessing are marked in yellow. The plot shows 100,000 answers in total for 100 different hard 3SAT instances ($\alpha \approx 4.2$). (Color figure online)

At a glance, that scaling factor with respect to problem difficulty is much better than what is observed for classical algorithms: For example, in the data used for Fig. 1 we observed performance decrease up to one order of magnitude larger. It is important to note, however, that these experiments were performed for problem instances so small that their evaluation does not pose a challenge to classical processors at all, i.e., below the point of reasonable performance metrics. Thus, these results only proof relevant to practical applications if they scale with future versions of quantum annealing hardware that can tackle much larger problem instances.

So far, we have not discerned between different correct solutions. We were content as long as the algorithm returned but one. However, for the user it is interesting to know if he or she will receive the same solution with every answer or an even distribution across the complete solution space. Our experiments show that when a lot of correct solutions are found for a certain problem instance, there are cases where we can see a clear bias towards a specific solution variant. Figure 4 shows the distributions of specific solutions for formulae that yielded many solutions even when evaluated without any postprocessing. While some formulae seem to yield rather narrow distributions over the different possible answers, others definitely seem to have a bias towards certain solutions. However, the former also tend to have relatively smaller sample sizes as there are less solutions in total to consider. Further investigation could still reveal a distinctive distribution in these cases as well. Thus, we consider this behavior of

the quantum annealer to be roughly in line with the findings of [31], who show an exponential bias in ground-state sampling of a quantum annealer.

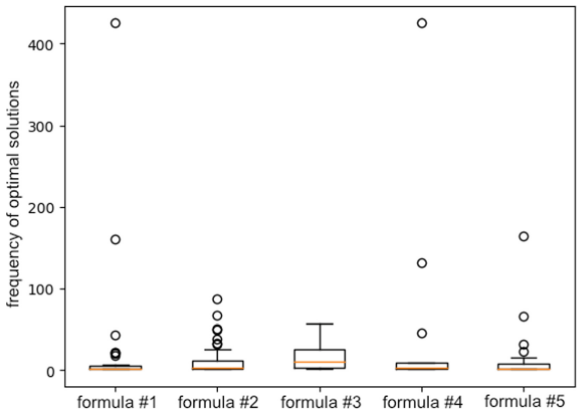

Fig. 4. Frequency of occurrence of different solutions for 5 formulae with many returned solutions without any postprocessing. While most solutions are found once or just a few times, there are specific solutions that are found much more often.

6 Conclusion

We have shown that problem difficulty of 3SAT instances also affects the performance of quantum annealing as it does for classical algorithms. However, bound by the nature of both approaches, the effects are quite different with complete classical algorithms showing longer runtimes and quantum annealing showing less precision. A first quantification of that loss of precision suggests that it may not be too detrimental and comparatively easy to deal with. However, because of the maximum available chip size for quantum annealing hardware at the moment, no large-scale test could be performed. No real assumptions on the scaling of this phenomenon (and thus the eventual real-world benefit) can be made yet.

Our results suggest there are cases where single solutions from a set of equally optimal solutions are much more likely to be returned than others. This observation is in line with other literature on the results of quantum annealing. However, it is interesting to note that it translates into the original problem space of 3SAT.

The observed results will gain more practical relevance with larger chip sizes for quantum annealers. We thus suggest to perform these and/or similar tests for future editions of quantum annealing hardware. If the effects persist, they can indicate a substantial advantage of quantum hardware over other known approaches for solving NP-complete problems.

Acknowledgement. Research was funded by Volkswagen Group, department Group IT.

References

1. Adams, D.: The Hitchhiker's Guide to the Galaxy (1979)
2. Albash, T., Lidar, D.A.: Adiabatic quantum computing. arXiv:1611.04471 (2016)
3. Amara, P., Hsu, D., Straub, J.E.: Global energy minimum searches using an approximate solution of the imaginary time Schrödinger equation. J. Phys. Chem. **97**(25), 6715–6721 (1993)
4. Apolloni, B., Carvalho, C., De Falco, D.: Quantum stochastic optimization. Stoch. Process. Their Appl. **33**(2), 233–244 (1989)
5. Apolloni, B., De Falco, D., Cesa-Bianchi, N.: A numerical implementation of "quantum annealing". Technical report (1988)
6. Ausiello, G., Protasi, M., Marchetti-Spaccamela, A., Gambosi, G., Crescenzi, P., Kann, V.: Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties, 1st edn. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-642-58412-1
7. Benjamin, S.C., Zhao, L., Fitzsimons, J.F.: Measurement-driven quantum computing: Performance of a 3-SAT solver. arXiv:1711.02687 (2017)
8. Bernstein, E., Vazirani, U.: Quantum complexity theory. SIAM J. Comput. **26**(5), 1411–1473 (1997)
9. Bravyi, S., Gosset, D., Koenig, R.: Quantum advantage with shallow circuits. Science **362**(6412), 308–311 (2018)
10. Cheeseman, P.C., Kanefsky, B., Taylor, W.M.: Where the really hard problems are. In: IJCAI, vol. 91 (1991)
11. Chen, L., Aihara, K.: Chaotic simulated annealing by a neural network model with transient chaos. Neural Netw. **8**(6), 915–930 (1995)
12. Choi, V.: Adiabatic quantum algorithms for the NP-complete maximum-weight independent set, exact cover and 3SAT problems. arXiv:1004.2226 (2010)
13. Choi, V.: Different adiabatic quantum optimization algorithms for the NP-complete exact cover and 3SAT problems. Quant. Inform. Comput. **11**(7–8), 638–648 (2011)
14. Cook, S.A.: The complexity of theorem-proving procedures. In: Proceedings of the Third Annual ACM Symposium on Theory of Computing. ACM (1971)
15. D-Wave Systems: Postprocessing Methods on D-Wave Systems (2016)
16. Ding, J., Sly, A., Sun, N.: Proof of the satisfiability conjecture for large k. In: Proceedings of the 47th Annual ACM Symposium on Theory of Computing, STOC 2015. ACM, New York (2015)
17. Farhi, E., Goldstone, J., Gosset, D., Gutmann, S., Meyer, H.B., Shor, P.: Quantum adiabatic algorithms, small gaps, and different paths. arXiv:0909.4766 (2009)
18. Farhi, E., Goldstone, J., Gutmann, S., Sipser, M.: Quantum computation by adiabatic evolution. arXiv preprint quant-ph/0001106 (2000)
19. Feld, S., et al.: A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. arXiv preprint arXiv:1811.07403 (2018)
20. Feynman, R.P.: Simulating physics with computers. Int. J. Theor. Phys. **21**, 467–488 (1982)
21. Finnila, A., Gomez, M., Sebenik, C., Stenson, C., Doll, J.: Quantum annealing: a new method for minimizing multidimensional functions. Chem. Phys. Lett. **219**(5–6), 343–348 (1994)
22. Gendreau, M., Hertz, A., Laporte, G.: A Tabu search heuristic for the vehicle routing problem. Manag. Sci. **40**(10), 1276–1290 (1994)
23. Glover, F., Laguna, M.: Tabu search*. In: Pardalos, P.M., Du, D.-Z., Graham, R.L. (eds.) Handbook of Combinatorial Optimization, pp. 3261–3362. Springer, New York (2013). https://doi.org/10.1007/978-1-4419-7997-1_17

24. Heim, B., Brown, E.W., Wecker, D., Troyer, M.: Designing adiabatic quantum optimization: a case study for the TSP. arXiv:1702.06248 (2017)
25. Hsu, T.J., Jin, F., Seidel, C., Neukart, F., De Raedt, H., Michielsen, K.: Quantum annealing with anneal path control: application to 2-SAT problems with known energy landscapes. arXiv:1810.00194 (2018)
26. Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse Ising model. Phys. Rev. E **58**(5), 5355 (1998)
27. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
28. Kirkpatrick, S., Selman, B.: Critical behavior in the satisfiability of random Boolean expressions. Science **264**(5163), 1297–1301 (1994)
29. Klauck, H.: The complexity of quantum disjointness. In: Leibniz International Proceedings in Informatics, vol. 83. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2017)
30. Lucas, A.: Ising formulations of many NP problems. Front. Phys. **2**, 5 (2014)
31. Mandra, S., Zhu, Z., Katzgraber, H.G.: Exponentially biased ground-state sampling of quantum annealing machines with transverse-field driving hamiltonians. Phys. Rev. Lett. **118**(7), 070502 (2017)
32. Marriott, C., Watrous, J.: Quantum Arthur-Merlin games. Comput. Complex. **14**(2), 122–152 (2005)
33. McGeoch, C.C.: Adiabatic quantum computation and quantum annealing: theory and practice. Synth. Lect. Quantum Comput. **5**(2), 1–93 (2014)
34. McGeoch, C.C., Wang, C.: Experimental evaluation of an adiabiatic quantum system for combinatorial optimization. In: Proceedings of the ACM International Conference on Computing Frontiers. ACM (2013)
35. Mézard, M., Zecchina, R.: Random k-satisfiability problem: from an analytic solution to an efficient algorithm. Phys. Rev. E **66**(5), 056126 (2002)
36. Monasson, R., Zecchina, R.: Entropy of the k-satisfiability problem. Phys. Rev. Lett. **76**(21), 3881 (1996)
37. Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B., Troyansky, L.: Determining computational complexity from characteristic "phase transitions". Nature **400**(6740), 133 (1999)
38. Morimae, T., Nishimura, H.: Merlinization of complexity classes above BQP. arXiv:1704.01514 (2017)
39. Moylett, D.J., Linden, N., Montanaro, A.: Quantum speedup of the traveling-salesman problem for bounded-degree graphs. Phys. Rev. A **95**(3), 032323 (2017)
40. Murty, K.G., Kabadi, S.N.: Some NP-complete problems in quadratic and nonlinear programming. Math. Program. **39**(2), 117–129 (1987)
41. Neukart, F., Compostella, G., Seidel, C., von Dollen, D., Yarkoni, S., Parney, B.: Traffic flow optimization using a quantum annealer. Front. ICT **4**, 29 (2017)
42. Rønnow, T.F., et al.: Defining and detecting quantum speedup. Science **345**(6195), 420–424 (2014)
43. Somma, R.D., Nagaj, D., Kieferová, M.: Quantum speedup by quantum annealing. Phys. Rev. Lett. **109**(5), 050501 (2012)
44. Strand, J., Przybysz, A., Ferguson, D., Zick, K.: ZZZ coupler for native embedding of MAX-3SAT problem instances in quantum annealing hardware. In: APS Meeting Abstracts (2017)
45. Van Dam, W., Mosca, M., Vazirani, U.: How powerful is adiabatic quantum computation? In: 42nd IEEE Symposium on Foundations of Computer Science. IEEE (2001)
46. Warren, R.H.: Small traveling salesman problems. J. Adv. Appl. Math. **2**(2), 101–107 (2017)