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
[2] ???
[3] Volkswagen Group of America
[4] OTH Regensburg

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: Quantum Computing · Quantum Annealing · 3SAT · Boolean satisfiabiliy · NP · phase transition.

1 Introduction

Following the great paper [16] we thought, let's do another one!

0.5 Introduction

2 Preliminaries

1 SAT w/ Phase Transition

2.1 Phase transitions in SAT solving

Diese Phasenübergangsgrenze teilt den Problemraum in zwei Regionen. In der einen Region kann eine Lösung relativ leicht gefunden werden, da die Lösungsdichte für diese Probleme hoch ist, in der anderen Region ist es sehr unwahrscheinlich, dass Probleme überhaupt eine korrekte Lösung enthalten können. Sehr schwer zu lösende Probleme befinden sich direkt an der Phasenübergangsgrenze [7]. Bei den in dieser Arbeit betrachteten 3SAT-Problemen liegt diese Phasenübergangsgrenze bei einem Klausel zu Variablenverhältnis von ca. 4.267 [29].

TODO: See Figure 1.

Das K -Erfüllbarkeitsproblem (im Folgenden *KSAT* genannt) gehört zur Klasse der NP-Vollständigen Probleme. KSAT ist, für K $>$ 3, ein zentrales Problem in der kombinatorischen Optimierung und war das erste Problem, für das die NP-Vollständigkeit gezeigt werden konnte [29].

Problembeschreibung: K - Erfüllbarkeitsproblem (nach [29]) Gegeben sei eine Menge $\mathcal{B}$, bestehend aus N booleschen Variablen. Aus dieser Menge $\mathcal{B}$ werden

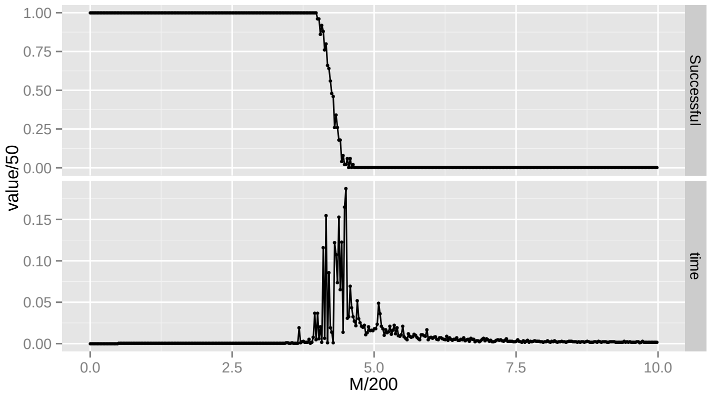

Fig. 1. Loesungsfindungsdauer von 3SAT-Instanzen mit 1000 Klauseln für verschiedene Anzahl von Variablen.

dann K Variablen ausgewählt. Die ausgewählten Variablen, oder deren Verneinungen, werden dann durch (K-1) Oder-Operatoren zu einer *Klausel* zusammengefasst. Auf diese Art und Weise werden M Klauseln gebildet. Durch anwenden von (M-1) Und - Operatoren entsteht so eine KSAT-Instanz.
Das KSAT-Problem ist nun die Frage, ob eine Belegung der boolschen Variablen aus $\mathcal{B}$ existiert, so dass eine gegebene KSAT-Instanz erfüllbar ist.

In dieser Arbeit werden ausschließlich zufällig generierte KSAT-Instanzen für K=3 betrachtet. Diese Probleme werden dann 3SAT genannt und sind nach [11] NP-Vollständing.

2.2 Kritischer Punkt bei zufällig generierten KSAT-Instanzen

Bei zufällig generierten KSAT-Instanen kann beobachtet werden, die Wahrscheinlichkeit, eine korrekte Lösung für die Instanz zu finden, abrupt sinkt, wenn ein kritischer Wert α_c (gebildet aus dem Verhältnis von Anzahl der Klauseln zu Anzahl der Variablen) überschritten wird [30].

Nach [29] liegt dieser kritische Punkt bei zufällig erzeugten 3SAT-Instanzen bei
$\alpha_c \approx 4,267$. In der Umgebung des kritschen Punkts ist die Lösungsfindung (damit ist hier nicht nur eine konkrete Belegung gemeint, falls die Instanz lösbar ist, sondern auch die Erkenntnis, dass diese Instanz nicht gelöst werden kann) algorithmisch komplex. Abb. **??** verdeutlicht dieses Phänomen visuell.

Zur Erstellung von Abb. **??** wurden mit Hilfe des *Tough SAT Generators*[5] 3SAT-Instanzen zu je 1000 Klauseln erstellt. Die Größe der Menge der boolschen Variablen wurde ausgehend von 1000 schrittweise um 10 verringert, bis ein abschließendes Klausel zu Variablenverhältnis von ca. 7 erreicht wurde. Für jede

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

Größe der so erzeugten Variablenmenge wurden 10 Instanzen erzeugt. Die verschiedenen Instanzen wurden dann mit dem SAT-Solver *Minisat*[6] gelöst. In Abb. **??** ist dann einmal die durchschnittliche sowie die maximale Lösungsdauer der 10 Instanzen für ein bestimmtes Verhältnis dargestellt.

3 Related Work

It is one of the cornerstones of complexity theory that solving NP-complete or even NP-hard decision problems is not efficiently possible (e.g., [11, 33]). Any NP-complete problem can also be cast as an optimisation problem, which allows for employing well-known optimisation algorithms to find approximate solutions – typical methods include tabu search (see, e.g., [19], [18]) and simulated annealing (see, e.g., [22], [8]). Countless other efficient approximation methods, together with an elaborate taxonomy on approximation quality (what is the solution's distance 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 (see, e.g., [**?**]).

Some problems, for instance knapsack, exhibit favourable properties when cast as an optimisation problem. The latter is a member of the complexity class FPTAS (fully polynomial-time approximation scheme), which means that a solution with distance $(1+\epsilon)$ $(\epsilon > 0)$ from an optimal solution can be determined in polynomial time in both, input size n and $1/\epsilon$ [8].

An intriguing connection that has received substantial attraction exists between (computational) NP-complete problems and the (physical) concept of phase transitions, as detailled in Sec. **??**. First investigations of the phenomenon have been performed by Kirkpatrick et al. [23], TODO [**?**] and TODO [**?**]; more recent investigations include [**?**,**?**,**?**].

The idea of obtaining solutions for NPO (optimisation) problems by finding the energy ground state(s) of a quantum mechanical system appeared at around 1988 [**?**,1]. Ausgangspunkt dieser Art kombinatorische Optimierungsprobleme zu lösen waren Apolloni et al. (vgl. [3], [4]). The idea of quantum annealing has been independently re-discovered multiple times, see [1, 17, 2, 21]).

Seit der ersten Erfindung des Quantum Annealings wurde für viele NP-vollständige Probleme eine Problemformulierung gefunden, so dass diese Probleme auf einem Quantenannealer gelöst werden können (vgl. [25]) und bis heute findet viel Forschung für Probleme wie das Problem des Handlungsreisenden oder das 3SAT-Problem im Kontext des Quantencomputings statt (vgl. [20], [37], [32], [34], [5]). Neben dem Quantum Annealing gibt es ein weiteres Model des Quantencomputings, genannt Quantum-Gate-Computing, welches polynomiell Äquivalent zum Model des Quantum Annealings ist [28], welches in dieser Arbeit jedoch nicht berücksichtigt wird. Mit der Erfindung des Quantencomputings, einer neunen Herangehensweise an das Lösen von Problemen, entstanden auch neue Komplexitätsklassen (vgl. [24], [31]), deren Beziehungen zur klassischen Komplexitätstheorie ebenfalls Teil der Forschung war (vgl. [6], [27]).

[6] http://minisat.se/

Zu den in dieser Arbeit im Kontext des Quantencomputing betrachteten 3SAT-Problemen fand bereits viel Foschung statt. So wurden verschiedene Methoden vorgestellt, diese Probleme auf einem Quantenannealer zu lösen (vgl [10], [9], [15]). Ebenso fanden Untersuchungen zur Lösungskomplexität von 3SAT-Problemen im Kontext des Quantencomputings statt (vgl. [36], [13]) und für bestimmte 3SAT-Instanzen die nicht effizient von einfachen Quantenalgorithmen gelöst werden können wurde eine mögliche Lösungsmehtode vorgeschlagen (vgl. [12]. Ebenso wurden Variationen des 3SAT-Problems, wie zum Beispiel das Exact Cover-Problem angewandt auf 3SAT-Formeln, betrachtet (vgl. [14]).

4 Experimental Setup

Quantum annealing is an optimization process that can be implemented in hardware. It is built on the adiabatic theorem which in theory guarantees the evolution of an initial configuration of the system to a configuration minimizing a specific user-defined energy function [28]. As in the real world the required conditions for the theorem can only be approximated, the results of quantum annealing are usually not exact but show a probabilistic distribution, ideally covering the desired optimal value as well.

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 [].

4.1 Defining 3SAT as a QUBO matrix

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

For this purpose, we use the approach given in [9, 10] by translating 3SAT into the Weak Maximum Independent Set (WMIS) problem first and then translating the WMIS instance into a QUBO matrix. We omit the details of this process and instead refer to literature [9, 10, 25]. However, we shall shortly discuss the implications of that translation process.

In the end for a 3SAT instance, i.e., a formula, with m clauses for n variables we end up with 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 a correct assignment, the corresponding 3SAT formula is satisfiable. If the quantum annealer only return 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.[7]

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 bit 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 $v_0 =$ *True* and $v_2 =$ *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 not that while the QUBO is built in such a way to opt for one single value 1 per triplet in the bitstring, 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.

[7] Perhaps interestingly, 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 argument for each part of the formula.

4.2 Evaluating Postprocessing

As can be seen from the last example, postprocessing is an integral part of solving problem with quantum annealing. As discussed in the previous section, we consider a QUBO solution correct not only if its 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, where the expected structure for QUBO optimizes so that the amount bits 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 next to 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 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 prove to be more powerful than logical postprocessing in the following Section 5. Again, for greater detail we refer to the D-Wave documentation on that matter [35]. 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*.

In order to evaluate the solution quality regarding 3SAT we employ both methods. The goal is to assess the 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 tool chain but not that interesting on its own for the scope of 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[8] we generated 3SAT instances of various difficulty (i.e., with various values for α). However, as discussed in Section 2, for $|\alpha - 4.2| \gg 0$ problem instances become very easy to solve. We observed that effect on the quantum annealer as well, i.e., all of these instances were trivially solved on the D-Wave machine. Thus, for the remainder of this section, we focus on hard instances (with $\alpha = 4.2$) to assess the quality in the really hard cases.

Experiments have shown that using the standard embedding tools delivered with the D-Wave platform, we can only reliably find a working embedding on

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

the D-Wave 2000Q chip for 3SAT instances with at most 42 clauses. In order to maintain $\alpha = 4.2$, it follows that the generated 3SAT instance contain 10 different variables. We only assess solution quality for 3SAT instances that are satisfiable (but of course we are not telling the solver that).

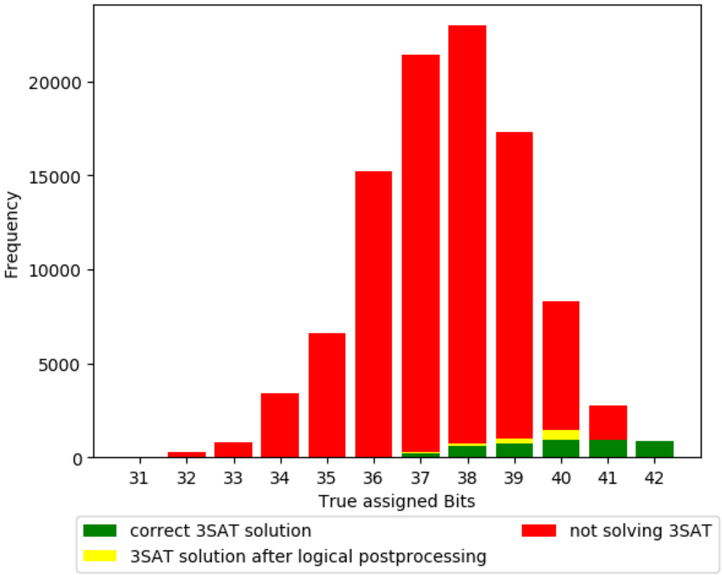

Fig. 2. Distribution of correct (green) and incorrect (red) answers returned by the quantum annealer *without D-WAVE postprocessing*. Answers that can trivially be transformed into valid answers using logical postprocessing are marked in yellow. The plot shows 100,000 answers in total for 100 different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

Figure 2 shows the result distribution of these runs on the D-Wave machine. In the x-axis, we sorted the returned results according to the bits that have been assigned the value 1 or *True*. As discussed in Section **??** 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 are answers that only set 10 bits that still map to complete and valid solution for 3SAT. From Figure 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 38 and no answer 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. It is important to note that although there is some amount of correct solutions, 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 Section **??** to the incorrect answers in Figure 2. However, it shows little improvement on the total amount of correct answers collected. As discussed, we expect the postprocessing method delivered with the D-Wave software package to be more powerful. So we ran the complete evaluation experiment again, only this time turning on the integrated postprocessing. The results are shown in Figure 3.

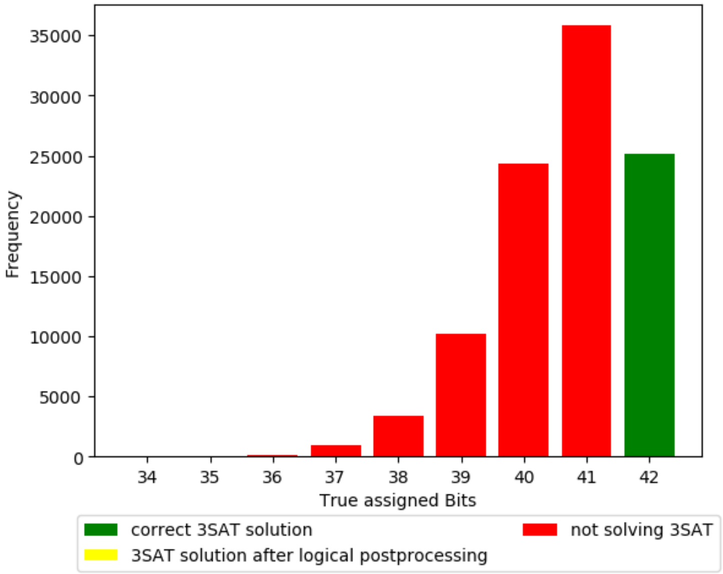

Fig. 3. Distribution of correct (green) and incorrect (red) answers returned by the quantum annealer *using D-WAVE postprocessing*. Answers that can trivially be transformed into valid answers using logical postprocessing are marked in yellow. The plot shows 100,000 answers in total for 100 different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

We observed that the D-Wave postprocessing managed to optimize all correct but "incomplete" answers, mapping them to 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 here. Effectively, this shows that quantum annealing does suffer from a breakdown in performance at the point of the phase transition in the 3SAT problem. In comparison to the immense decrease in performance seen in classical solvers (cf. Section **??**), a drop to around 25% appears rather desirable. A quick example: To achieve a $1 - 10^{-12}$ confidence of returning the correct answer our experimental setup requires around 97 queries. At a glance, that scaling factor with respect to problem difficulty is much better than what is observed for classical algorithms. 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. These results only proof relevant 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 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. 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 [26] who show an exponential bias in ground-state sampling of a quantum annealer.

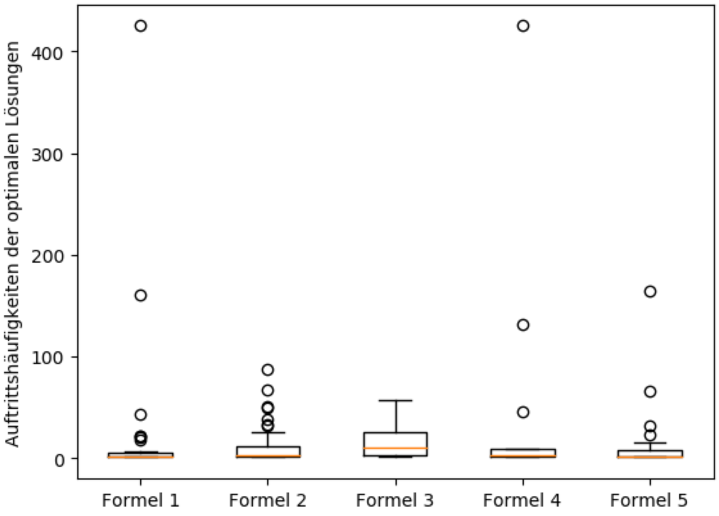

Fig. 4. Frequency of occurrence of different solutions for 5 formulae with many returned solutions. While most solutions are found once or just a few times, there are specific solutions that are found much more often.

Lastly, we performed an additional experiment to check for the contingency of the presented results. As we have considered rather small problem instances (to the current limitations of the D-Wave chip) and a rather over-specified encoding of 3SAT (which allows for things like logical postprocessing to add benefit in the first place), we posed the question how much of solving these 3SAT instances actually requires an "intelligent" solving algorithm and how much part of the solution can be attributed to the environment. To examine this, we generated "solutions" to 3SAT instances using a random generator. We first considered a simple random generator that produced bitstrings of the same length as the D-Wave machine's answers. Figure 5 (left) shows the result for *easy* 3SAT instances, i.e. $\alpha = 0.2$: Out of 100,000 queries for 100 different formulae, 19 returned answers represented a valid 3SAT solution. Interestingly, logical postprocessing was still able to yield another 2,425 solutions. When we increased the problem difficulty to the previously considered $\alpha = 4.2$, none of the randomly generated "solutions" were correct or could be postprocessed to be correct. Futhermore, we performed this test on another random generator which return bitstrings of the form $\{001, 010, 100\}^{42}$, i.e., solutions which are always correct at setting just one bit per 3SAT clause. While this allowed us to find more (i.e., 12,043) correct solutions in the easy instances (cf. the right part of Figure 5), we still id not generate a single correct solution among 10,000 for the transition point instances.

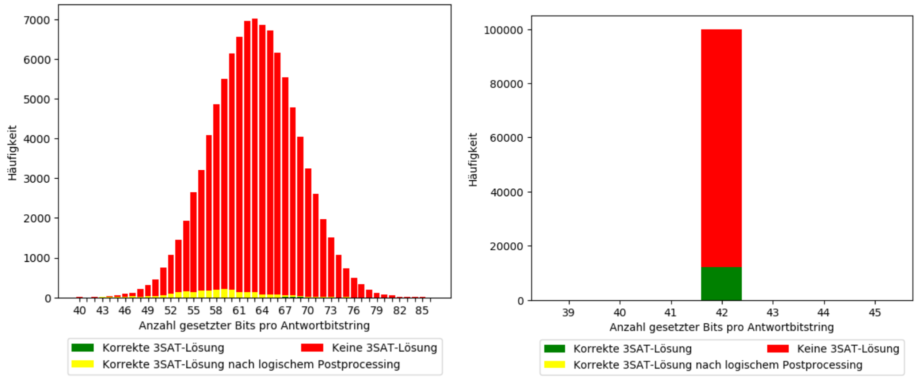

Fig. 5. Distribution of correct (green) and incorrect (red) answers returned by two different random generators. Answers that can trivially be transformed into valid answers using logical postprocessing are marked in yellow. Both plots shows 100,000 answers in total for 100 different hard 3SAT instances ($\alpha \approx 0.2$).

6 Conclusion

1 Discussion & Conclusion (7.5.3)

References

1. Albash, T., Lidar, D.A.: Adiabatic quantum computing. arXiv preprint arXiv:1611.04471 (2016)
2. Amara, P., Hsu, D., Straub, J.E.: Global energy minimum searches using an approximate solution of the imaginary time schrödinger equation. The Journal of Physical Chemistry **97**(25), 6715–6721 (1993)
3. Apolloni, B., Carvalho, C., De Falco, D.: Quantum stochastic optimization. Stochastic Processes and their Applications **33**(2), 233–244 (1989)
4. Apolloni, B., De Falco, D., Cesa-Bianchi, N.: A numerical implementation of "quantum annealing". Tech. rep. (1988)
5. Benjamin, S.C., Zhao, L., Fitzsimons, J.F.: Measurement-driven quantum computing: Performance of a 3-sat solver. arXiv preprint arXiv:1711.02687 (2017)
6. Bernstein, E., Vazirani, U.: Quantum complexity theory. SIAM Journal on Computing **26**(5), 1411–1473 (1997)
7. Cheeseman, P.C., Kanefsky, B., Taylor, W.M.: Where the really hard problems are. In: IJCAI. vol. 91, pp. 331–340 (1991)
8. Chen, L., Aihara, K.: Chaotic simulated annealing by a neural network model with transient chaos. Neural networks **8**(6), 915–930 (1995)
9. Choi, V.: Adiabatic quantum algorithms for the np-complete maximum-weight independent set, exact cover and 3sat problems. arXiv preprint arXiv:1004.2226 (2010)
10. Choi, V.: Different adiabatic quantum optimization algorithms for the np-complete exact cover and 3sat problems. Quantum Information & Computation **11**(7-8), 638–648 (2011)
11. Cook, S.A.: The complexity of theorem-proving procedures. In: Proceedings of the third annual ACM symposium on Theory of computing. pp. 151–158. ACM (1971)
12. Farhi, E., Goldstone, J., Gosset, D., Gutmann, S., Meyer, H.B., Shor, P.: Quantum adiabatic algorithms, small gaps, and different paths. arXiv preprint arXiv:0909.4766 (2009)
13. Farhi, E., Goldstone, J., Gutmann, S.: A numerical study of the performance of a quantum adiabatic evolution algorithm for satisfiability. arXiv preprint quant-ph/0007071 (2000)
14. Farhi, E., Goldstone, J., Gutmann, S., Lapan, J., Lundgren, A., Preda, D.: A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. Science **292**(5516), 472–475 (2001)
15. Farhi, E., Goldstone, J., Gutmann, S., Sipser, M.: Quantum computation by adiabatic evolution. arXiv preprint quant-ph/0001106 (2000)
16. Feld, S., Roch, C., Gabor, T., Seidel, C., Neukart, F., Galter, I., Mauerer, W., Linnhoff-Popien, C.: A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. arXiv preprint arXiv:1811.07403 (2018)
17. Finnila, A., Gomez, M., Sebenik, C., Stenson, C., Doll, J.: Quantum annealing: a new method for minimizing multidimensional functions. Chemical physics letters **219**(5-6), 343–348 (1994)
18. Gendreau, M., Hertz, A., Laporte, G.: A tabu search heuristic for the vehicle routing problem. Management science **40**(10), 1276–1290 (1994)
19. Glover, F., Laguna, M.: Tabu search*. In: Handbook of combinatorial optimization, pp. 3261–3362. Springer (2013)
20. Heim, B., Brown, E.W., Wecker, D., Troyer, M.: Designing adiabatic quantum optimization: A case study for the traveling salesman problem. arXiv preprint arXiv:1702.06248 (2017)

21. Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse ising model. Physical Review E **58**(5), 5355 (1998)
22. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. science **220**(4598), 671–680 (1983)
23. Kirkpatrick, S., Selman, B.: Critical behavior in the satisfiability of random boolean expressions. Science **264**(5163), 1297–1301 (1994)
24. Klauck, H.: The complexity of quantum disjointness. In: LIPIcs-Leibniz International Proceedings in Informatics. vol. 83. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2017)
25. Lucas, A.: Ising formulations of many np problems. Frontiers in Physics **2**, 5 (2014)
26. Mandra, S., Zhu, Z., Katzgraber, H.G.: Exponentially biased ground-state sampling of quantum annealing machines with transverse-field driving hamiltonians. Physical review letters **118**(7), 070502 (2017)
27. Marriott, C., Watrous, J.: Quantum arthur–merlin games. Computational Complexity **14**(2), 122–152 (2005)
28. McGeoch, C.C.: Adiabatic quantum computation and quantum annealing: Theory and practice. Synthesis Lectures on Quantum Computing **5**(2), 1–93 (2014)
29. Mézard, M., Zecchina, R.: Random k-satisfiability problem: From an analytic solution to an efficient algorithm. Physical Review E **66**(5), 056126 (2002)
30. Monasson, R., Zecchina, R.: Entropy of the k-satisfiability problem. Physical review letters **76**(21), 3881 (1996)
31. Morimae, T., Nishimura, H.: Merlinization of complexity classes above bqp. arXiv preprint arXiv:1704.01514 (2017)
32. Moylett, D.J., Linden, N., Montanaro, A.: Quantum speedup of the traveling-salesman problem for bounded-degree graphs. Physical Review A **95**(3), 032323 (2017)
33. Murty, K.G., Kabadi, S.N.: Some np-complete problems in quadratic and nonlinear programming. Mathematical programming **39**(2), 117–129 (1987)
34. 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)
35. Systems, D.W.: Postprocessing Methods on D-Wave Systems (2016)
36. Van Dam, W., Mosca, M., Vazirani, U.: How powerful is adiabatic quantum computation? In: Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on. pp. 279–287. IEEE (2001)
37. Warren, R.H.: Small traveling salesman problems. Journal of Advances in Applied Mathematics **2**(2) (2017)