Article

Reverse Quantum Annealing Assisted by Forward Annealing

Manpreet Singh Jattana [ID]

Modular Supercomputing and Quantum Computing, Goethe University Frankfurt, Kettenhofweg 139, 60325 Frankfurt am Main, Germany; jattana@em.uni-frankfurt.de

Abstract: Quantum annealers conventionally use forward annealing to generate heuristic solutions. Reverse annealing can potentially generate better solutions but necessitates an appropriate initial state. Ways to find such states are generally unknown or highly problem dependent, offer limited success, and severely restrict the scope of reverse annealing. We use a general method that improves the overall solution quality and quantity by feeding reverse annealing with low-quality solutions obtained from forward annealing. An experimental demonstration of solving the graph coloring problem using the D-Wave quantum annealers shows that our method is able to convert invalid solutions obtained from forward annealing to at least one valid solution obtained after assisted reverse annealing for 57% of 459 random Erdős–Rényi graphs. Our method significantly outperforms random initial states, obtains more unique solutions on average, and widens the applicability of reverse annealing. Although the average number of valid solutions obtained drops exponentially with the problem size, a scaling analysis for the graph coloring problem shows that our method effectively extends the computational reach of conventional forward annealing using reverse annealing.

Keywords: reverse annealing; forward annealing; graph coloring

1. Introduction

Quantum annealers have made steady progress towards solving prototype real-world problems [1–4] along with gate-based quantum computers [5–7] and their hybrid applications [8–13]. Quantum annealers are based on the principle of quantum annealing, which is based on the adiabatic theorem. Quantum annealing has been described as a metaheuristic for solving combinatorial optimization problems [14–16].

Quantum annealing can be divided into two categories: forward annealing and reverse annealing (RA). The forward or conventional quantum annealing process starts in an initial state that is a uniform superposition of all possible classical states. The advantage of such an initial state is that it is an easy-to-prepare ground state of a known problem. The idea is to exploit the adiabatic theorem, whereby the amplitude of the transverse field is slowly decreased to zero, and the system remains in its instantaneous ground state at all times, thus yielding the solution to the optimization problem at the end of the process.

If we define the total Hamiltonian H as

$$H = A(s)H_p + B(s)H_i, \tag{1}$$

then H_i is the initial Hamiltonian and H_p is the problem Hamiltonian. $A(s)$ and $B(s)$ describe the annealing schedule. In forward annealing, the function $A(s)$ is gradually increased from 0 to 1, i.e., $A(s) : 0 \rightarrow 1$ and $B(s) : 1 \rightarrow 0$, when $s : 0 \rightarrow 1$ within a certain annealing time t. A majority of applications of quantum annealers involve this setup. However, several recent papers have studied or used reverse annealing [14,17–30] for various problems.

The process of reverse annealing, in contrast to forward annealing, does not start in an initial state corresponding to the superposition of all possible classical states. It starts precisely in only one of all possible classical states [14,17]. The amplitude of the transverse field is thereafter first gradually increased to a certain point and then decreased.

Thus, according to Equation (1), reverse annealing undergoes $A(s) : 1 \rightarrow A(s') \rightarrow 1$, $B(s) : 0 \rightarrow B(s') \rightarrow 0$, when $s : 1 \rightarrow s' \rightarrow 1$, where $0 < s' < 1$ is called the reverse distance. Reverse annealing has been shown to produce better results with a dependence on an appropriate choice of the initial state [17–22]. In contrast, an inappropriate choice of the initial state can be generally expected to lead to no improvement in the solution quality. Thus, such initial states play a decisive role in the effectiveness of reverse annealing.

Currently, it is not always straightforward to find the appropriate choice of initial state for a given problem. Previous works have used classical greedy algorithms [19], simulated annealing and approximate message passing algorithm [22], and classical preprocessing or physical intuition [17]. A general strategy is missing that can be put to use for any problem.

In contrast to forward annealing, which can be termed as a global search routine aimed at finding the global minimum, reverse annealing can be termed as a local search routine that refines existing solutions [24]. The combination of both can potentially use the broad exploration capability of forward annealing and then perform local search refinement using reverse annealing [31]. However, it is unclear how to effectively combine these two in practice.

In this work, we propose a method that uses forward annealing to provide good initial states for reverse annealing. To demonstrate the idea, we first use forward annealing to solve the graph coloring problem and increase the problem size until no valid solutions are sampled. We then show that reverse annealing can find valid solutions when invalid solutions obtained from forward annealing are used as the initial state. It also obtains more unique solutions. Additionally, using forward annealing to provide the initial state outperforms guessing it randomly. Our approach can be seen as improving the overall metaheuristic of quantum annealing with general applicability.

The rest of the paper is structured as follows. In Section 2, we give a brief overview of the problem that we will solve in this work. In Section 3, we introduce the theory of RA and mention the relevant parameters at play in the process. In Section 4, we show several results of performing reverse annealing in four different cases, undertake a scaling analysis by increasing the problem sizes, compare our strategy against random initializations, and demonstrate the device independence of our idea. We discuss and conclude in Section 5.

2. Graph Coloring

Given an undirected graph $G = (V, E)$, where V represents the set of vertices and E represents the set of edges connecting these vertices, the graph coloring problem involves finding an order of coloring such that no two connected vertices receive the same color and all vertices are colored. The graph coloring problem has two sub-branches: the minimum-coloring problem and k-coloring problem.

The minimum-coloring problem seeks to find the smallest number of colors k to properly color G. This k is also called the chromatic number of G. The k-coloring problem seeks to color G using no more than k colors. The former is NP-hard and the latter NP-complete for $k \geq 3$ [32,33]. In this work, we use the quantum annealer to solve the k-coloring problem. We create random graphs according to the Erdős–Rényi model using the *networkx* package in Python [34]. We set the value of k equal to the number of colors needed by the 'largest first' greedy algorithm.

The graph coloring problem is reformulated as a quadratic unconstrained binary optimization (QUBO) problem using a Python package [35], which is based on Ref. [36] and can be solved on quantum annealers [37]. Some previous work has been undertaken that used quantum annealers: as an independent set sampler [38] and using a domain-wall encoding [39]. However, none of the previous works on reverse annealing known to the author [14,17–23,25,26] studied it for the graph coloring problem.

3. Reverse Annealing

Reverse annealing can be categorized into different categories based on either the (a) Hamiltonian and its implementation or (b) the method used to produce the initial state for

the first cycle of RA. In case of (a), RA is of three types: first, adiabatic RA, in which an additional term is added to H given in Equation (1) and can be used to perform analytical treatment of a problem [20,40]; second, repetitive RA, in which no terms are added to Equation (1) and the initial state is always the same for each run [41]; third, iterated RA, in which no terms are added to Equation (1) but the output of a RA cycle is fed as input to sample the output from the next RA cycle. Iterated RA is an important tool because the progress of a previous cycle can be improved even further in a new cycle, in contrast to repetitive RA. If the initial state to iterated RA is a higher-order excited state, then it is possible to progress towards the ground state by reaching lower-order excited states through progressive cycles. Iterated RA is commonly used on D-Wave systems [21,42]. We also use iterated RA for all our experiments.

In case of (b), RA is of at least four types: First, random input RA where an initial state is randomly generated; second, classically assisted RA, in which an algorithm on a classical computer produces an initial state; third, RA in which the initial state is motivated by physical or intuitive reasoning about the problem Hamiltonian; fourth, quantum-assisted RA, in which either the same or a different (type of) quantum device is used to produce the initial state. The novelty of our work lies in successfully combining this last category with iterated RA.

Since we use a quantum annealer to produce the initial state in the forward annealing setting, we call it forward-annealing-assisted RA. The algorithm is illustrated in Figure 1. The problem graph is created on a classical computer and formulated into a QUBO problem. The quantum annealer then tries to solve the problem using forward annealing, and its output is fed to a validator, which checks for valid solutions among the bitstrings. Valid solutions are those that color our graphs in exactly k colors without violating the constraints. If at least one valid solution is found among all the sampled bitstrings, then forward annealing has solved the problem and the algorithm ends. In contrast, if no valid solutions are found, one of the invalid solutions is input as the initial state for the first cycle of RA. One or more samples can be obtained through RA. The validator will check the output bitstring obtained from RA for validity. If an invalid solution is sampled, it is (within iterated RA) taken as an input for the next RA. The algorithm comes to an end if a valid solution is sampled.
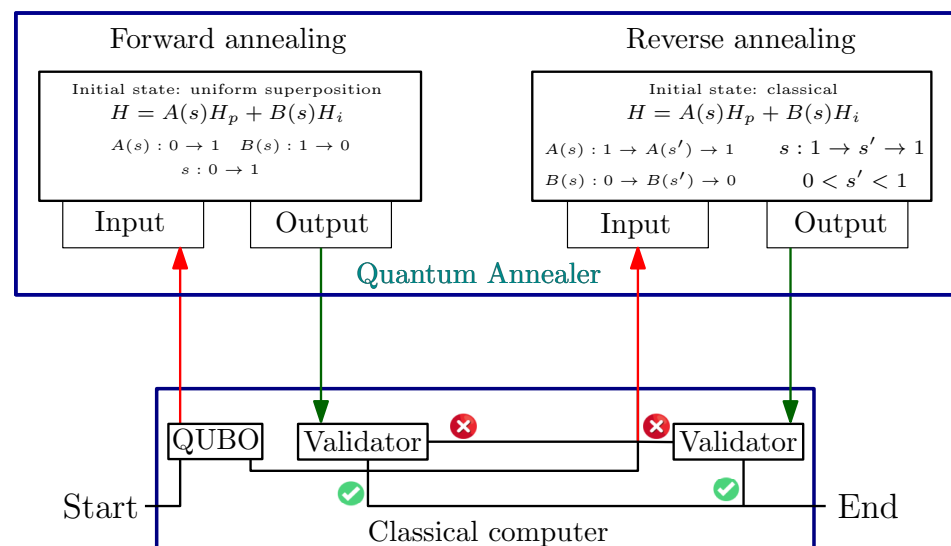

Figure 1. Working of the forward-annealing-assisted reverse-annealing algorithm. The QUBO problem is first solved using forward annealing, and the algorithm halts if the output of the quantum annealer is a valid bitstring (green check mark). Otherwise, the lowest energy bitstring is taken as input for reverse annealing, which can be made to run until a valid solution is obtained or a given sampling limit is reached.

RA Parameters

An important factor in the success of RA is the choice of parameters. These include the reverse distance, annealing schedule, and annealing path. We look at these parameters closely in this section.

In the following, we illustrate using a simple example of how choosing the reverse distance can be relevant and its relationship to different annealing schedules. To do so, we will look at the energy spectrum of H from Equation (1) for one case of the graph coloring problem. Consider a graph with five vertices in the form of a one-dimensional chain without periodic boundaries. This graph can be colored using two colors. To proceed, we first convert the QUBO to its corresponding H_p. A given QUBO variable can be mapped to the corresponding Ising variable using

$$s_i = 1 - 2x_i, \tag{2}$$

where $s \in \{+1, -1\}$ is the i-th spin variable and $x \in \{0, 1\}$ is the i-th binary variable. The Ising Hamiltonian can be written as [43,44]

$$H_p = \sum_{i,j=1}^{N} Q_{ij}(\mathbf{1} - Z_i)(\mathbf{1} - Z_j), \tag{3}$$

where Q is the QUBO matrix and Z is the σ^z Pauli operator. Python packages are available for converting QUBO to Ising and vice versa [37].

One must decide on an annealing scheme to study the energy spectrum in Equation (1). A linear scheme is one of the simplest. To use this scheme, we set

$$H = sH_p + (1-s)H_i, \tag{4}$$

where $H_i = \sum_{j=1}^{N} \sigma_j^x$, and s increases from 0 to 1 in one hundred equal steps. At each step, we reconstruct and exactly diagonalize H and plot its 15 lowest-lying eigenvalues. We include degenerate cases, if any. The spectrum is shown in Figure 2.
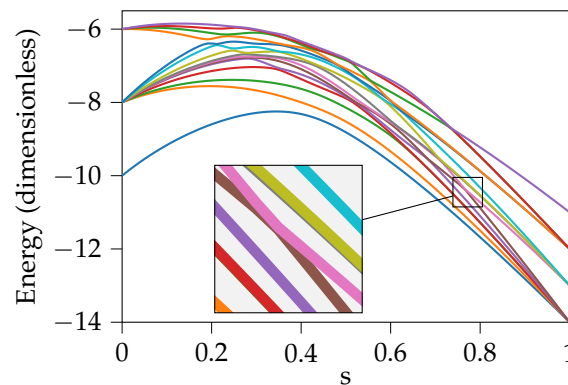

Figure 2. Energy spectrum of a coloring problem with five vertices using a linear annealing schedule. Shown are the 15 lowest-lying eigenvalues as a function of s, the annealing parameter. Inset: an anti-crossing between paths of ground and first excited states, shown in brown and pink, respectively.

The energy spectrum in Figure 2 reveals that the ground state is degenerate. This is understood intuitively by noting that, in general, there are multiple ways to color a graph. It is interesting to note that multiple paths lead to the ground state in our example. Several anti-crossings can be observed. We highlight one of them (in pink) in the inset magnified image, where one starts from the first excited state (at $s = 1$), anneals back up to $s \approx 0.8$, and performs forward annealing; then, if the system shifts to another path (in brown), it leads to the ground state. Such a situation would imply that we started from an invalid solution initial state and obtained a valid solution final state.

Actual quantum hardware, however, rarely follows the relatively simple linear schedule considered above. We plot the data released by D-Wave about their annealing schedule in Figure 3 for the Advantage System 5.4. We observe immediately that this schedule differs substantially from a linear schedule. Using H_p from Equation (3), the D-Wave data are then used in Equation (1) as the annealing schedule. The energy spectrum is shown in Figure 4. Although H is the same, the new spectrum looks visibly different from that of the linear schedule in Figure 2. The anti-crossings have shifted to a different range of s. Thus, unlike in the previous case, annealing back to $s \approx 0.8$ will most likely not work and a different value of $s \approx 0.4$ appears more appropriate.
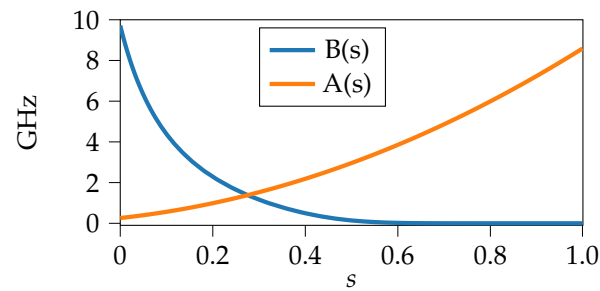

Figure 3. Annealing schedule used by D-Wave quantum annealers. The parameter s goes from 0 to 1 during forward annealing, whereas it goes from 1 to some middle non-zero value and then back to 1 for reverse annealing.
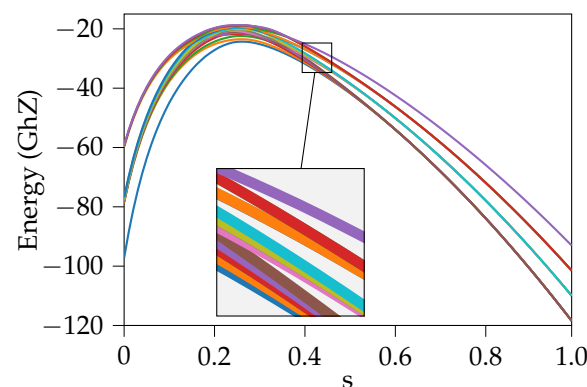

Figure 4. Same as Figure 2 except using the schedule given in Figure 3 rather than a linear schedule given in Equation (4).

In summary, the energy spectrum reveals that some paths that lead to the ground state and excited states have a low minimum gap between them, even for a very small graph coloring problem. Additionally, the value of s, where the minimum gaps come closest, depends strongly on the annealing schedule. Finally, the appropriate choice of the reverse distance combined with the quality of the input initial state can be decisive factors in our ability to reach the ground state.

Although the above example gives us a theoretical understanding, it must be remembered that this spectrum is the ideal case at zero temperature. Temperatures in an actual device, e.g., from D-Wave, can be around 20 mK. Such low but finite temperatures can be expected to induce effects not accounted for in the theoretical zero-temperature case for annealing times considered in this work.

We fix the total annealing time for both forward and reverse annealing to 100 μs. The RA path defines the way in which the lowest reverse distance is reached and scaled back. We include a small pause in the schedule, which has been suggested to lead to better results [45]. Three examples are shown in Figure 5. The D-Wave annealers take a maximum of 12 inputs for the values of s. We use the same anneal path for all our experiments and from now on will only refer to the reverse distance, which is the lowest s on the path.
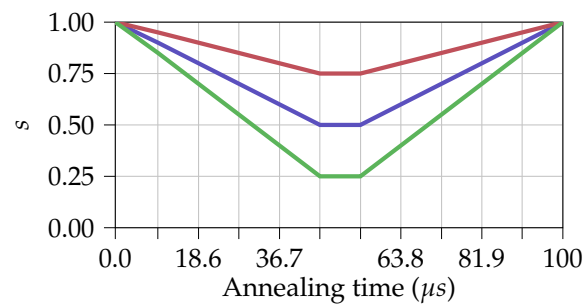
Figure 5. Reverse annealing paths for three different reverse distances: 0.25, 0.50, and 0.75. A maximum of 12 points are allowed to be defined, which are labeled on the x-axis.

4. Results

In this section, we thoroughly test RA for a variety of different cases for the graph coloring problem. We show the results obtained when feeding the bitstrings from forward annealing as the initial state of reverse annealing. We obtain 10^3 forward annealing samples for each case in this section. If forward annealing finds one valid solution, this is taken as the initial classical state for the reverse annealing process. If forward annealing finds more than one valid solution, then one of the solutions is randomly chosen. This allows us to test if RA still works if a valid solution is used as an initial state. If forward annealing fails to find a valid solution, the sample with the lowest energy is chosen.

We divide the results into two sections. First, we study the quality of solutions obtained from reverse annealing as a function of the reverse distance. Second, we compare the scaling analysis of obtaining valid solutions using reverse and forward annealing for different problem sizes. We also use another device to illustrate that our proposal is device-independent. We always use the iterated version of reverse annealing where the total number of samples equals the number of iterated cycles.

4.1. Reverse Distance

In order to demonstrate our idea, we create 500 random graphs generated according to the Erdős–Rényi model [46], and the number of vertices is set to 15. The value of *k* is obtained from the greedy algorithm. To test assisted reverse annealing, the reverse distance is decreased from 0.93 to 0.30 in a total of 10 equally spaced steps. For each value of the reverse distance, we obtain 10^3 samples. We analyze the total number of valid solutions as well as the total number of unique valid solutions. Since the initial state of the RA process can itself either be a valid or an invalid solution, depending on whether the forward annealing process was able to find at least one valid solution out of 10^3 samples, we then have four cases:

A.   Valid RA solutions; valid solution initial state. See Figure 6.
B.   Unique RA solutions; valid solution initial state. See Figure 7.
C.   Valid RA solutions; invalid solution initial state. See Figure 8.
D.   Unique RA solutions; invalid solution initial state. See Figure 9.

4.1.1. Case A

Shown in Figure 6 are the total numbers of valid solutions obtained at different values of the reverse distance. Plotted are only 41 out of 500 problems because forward annealing was able to find a valid solution only in these cases. Since we start from a valid solution obtained from forward annealing, we observe that almost all obtained samples are valid solutions at larger values of the reverse distance. This behavior is expected between a reverse distance range of 0.75 and 1.0. One explanation arises from the fact that the function $B(s)$ changes only negligibly between 0.75 and 1.0 in the annealing schedule of the D-Wave annealer shown in Figure 3. Due to this, the *H* in Equation (1) has H_p as the dominant component. Therefore, upon reverse annealing, no major contribution is expected from the mixing term H_i and the annealer is expected to stay in the provided initial state.
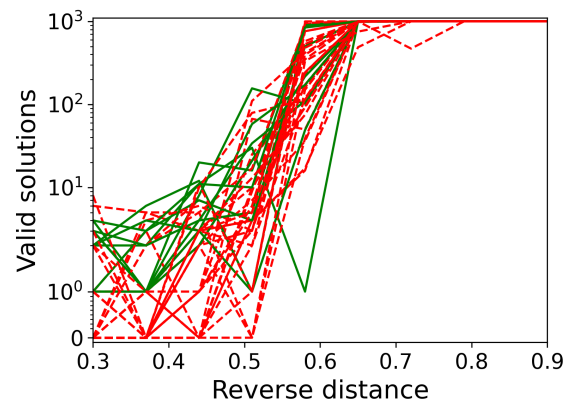
Figure 6. Total number of valid solutions obtained as a function of the reverse distance. A valid solution from forward annealing was used as an initial state to the RA process. The graph represents a total of 41 out of 500 problems. Green (Red) lines represent that there was at least one valid solution at all values (at least one value) of the reverse distance.
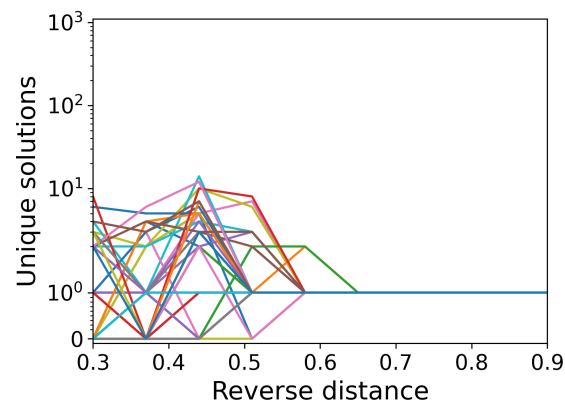

Figure 7. Same as Figure 6 except the y-axis shows total number of unique valid solutions. Each problem is given by a random color.
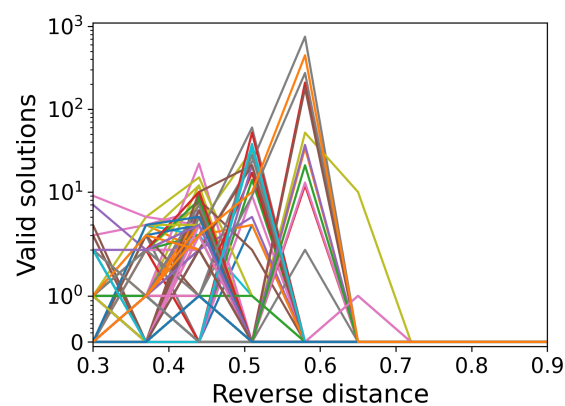

Figure 8. Total number of valid solutions obtained as a function of the reverse distance. An invalid solution with the lowest energy from forward annealing was used as an initial state to the RA process. The graph represents a total of 262 out of 500 problems.
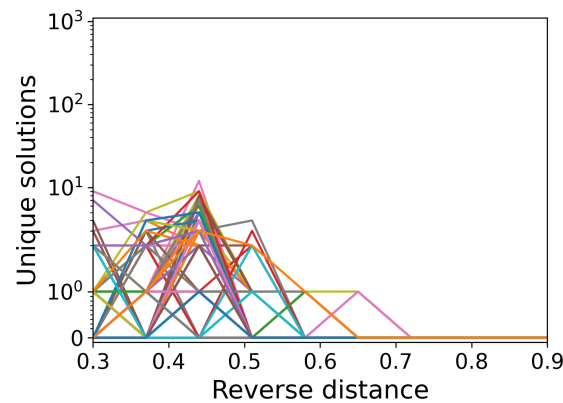
Figure 9. Same as Figure 8 except that the y-axis shows the total number of unique valid solutions.

The number of valid solutions decreases drastically as the reverse distance is decreased further below 0.75. The mixing term H_i also becomes relevant and the annealer is no longer able to sample only the initial state it was provided, leading to a decrease in the number of valid solutions.

We also observe that there are cases in which at least one valid solution was obtained for each value of the reverse distance, shown in green. However, in most cases, not all values of the reverse distance yield at least one valid solution, as shown by the red (dotted) lines. The fact that forward annealing was able to find a valid solution in only 8.2% of the total cases is an indication that the problem size is appropriate to test whether reverse annealing can obtain valid solutions for the rest of the cases.

4.1.2. Case B

Deterministic greedy algorithms find one of all possible valid solutions if the solution space is degenerate. A quantum annealer may offer the advantage that one may obtain different unique solutions by no additional effort than sampling more often than once. We study this case by filtering out the number of unique valid solutions obtained from Case A and plot the results in Figure 7.

We observe that at large reverse distances, all problems converge to the value of one unique solution. The unique solution is clearly separate for each problem. This solution is the same one that we use as the initial state, which was in turn obtained from forward annealing. Thus, from Figures 6 and 7, we discern that we exclusively sample the initial state input to the RA process for larger annealing distances.

The results take an exciting turn at lower reverse distances. We observe a novel pattern where unique solutions start to emerge below a reverse distance of 0.60, reach a peak at 0.44, and then decline again at 0.30. This result is interesting because we showed previously in Figure 4 that the anti-crossings appear close to 0.4, albeit for a smaller problem size. Thus, it is possible to sample more unique solutions simply by choosing an appropriate reverse distance.

4.1.3. Case C

The most interesting test for reverse annealing is when forward annealing finds no valid solutions. This was the case for 459 out of 500 problems. We take the lowest energy bitstring as the initial state and plot the total number of valid solutions obtained as a function of the reverse distance. The results are shown in Figure 8.

In contrast to Case A, we observe no valid solutions for all problems at larger reverse distances. This behavior is expected since, at large values, the annealer outputs the same invalid bitstring input as the initial state. Valid solutions start to appear for reverse distances less than 0.7, and the highest total of these valid solutions tends to decrease as the reverse distance decreases.

At smaller reverse distances, we were able to obtain at least one valid solution for 262 out of 500 problems. Figure 8 only shows these 262 problems; we omit those for which

reverse annealing also found no valid solution. The peak number of total solutions obtained differs for each problem across the reverse distance. From Case C, we conclude that reverse annealing is able to find valid solutions in about 57% of problems where forward annealing does not.

4.1.4. Case D

We plot the unique solutions obtained in Case C in Figure 9. We observe that the number of unique solutions first increases as the reverse distance is decreased, peaks at 0.44, and then decreases when further decreasing the reverse distance. A majority of the problems had the maximum number of unique solutions at a reverse distance of 0.44. In contrast to Case B, there are no valid solutions at larger reverse distances; hence, all lines converge to the value 0.

4.2. Scaling Analysis

In the previous section, we observed that reverse annealing is able to find valid solutions where forward annealing fails. In this section, we perform an analysis across different problem sizes. In addition to the data for the 500 graphs with 15 vertices shown in the previous section, we generate 100 new random graphs, each with 7, 10, 17, and 20 vertices.

Since all the graphs are generated randomly, it is not necessary that all of them having the same node size can be colored using the same number of colors. The number of qubits required is also dependent on the number of colors needed. Hence, there will be a distribution of different number of qubits used across the 100 (or 500) random graphs of a given node size. Therefore, we chose the node sizes such that we sufficiently cover all relevant qubit numbers for our scaling analysis. The software from D-Wave automatically creates chains of physical qubits to form logical qubits to accommodate larger problems. We fix the reverse distance to 0.44 for all problems. This choice is inspired from the results of Case B from Section 4.1.

We calculate the average number of valid solutions obtained for all problems that needed the same logical qubit count. The data are plotted in Figure 10. The number of problems per qubit number differs between different qubit numbers.

We observe that the frequency of success vanishes exponentially in problem size for both forward and reverse annealing. For small problem sizes, given by the corresponding small number of logical qubits required (i.e., <40), we observe that both forward and reverse annealing work almost equally well. In the next region between 40 and 80 qubits, the frequency of finding valid solutions using forward annealing drops further exponentially to below 1. Beyond 80 qubits, forward annealing completely fails to find valid solutions on average. In contrast, reverse annealing is able to find more valid solutions on average in all cases up to 80 qubits. The most interesting cases happen beyond 80 qubits where reverse annealing is able to find at least one valid solution on average in contrast to forward annealing. The average number of valid solutions for reverse annealing drop further until 119 qubits.

At 120 qubits, both forward and reverse annealing yield no valid solution for any of the problems. These problems have our largest node size of 20. Our obtained data for 160 and 180 qubits yield also zero valid solutions for all tested problems [47].

We now move to analyzing average unique solutions obtained. The data are shown in Figure 11. We observe that, at first, the number of average unique solutions increases, reaches a peak, and then decreases. This is explained by the fact that for small problem sizes, there does not exist a large number of degenerate solutions and the quantum annealer is able to find all possible unique solutions using both forward and reverse annealing. For large problem sizes, neither forward nor reverse annealing are able to find all unique solutions. Increasing the problem size further then shows an exponential decline in the average number of unique solutions found. Here, the plot exhibits the same features as the

plot in Figure 10. In summary, reverse annealing is able to find more unique solutions on average than forward annealing for the problem sizes we tested.
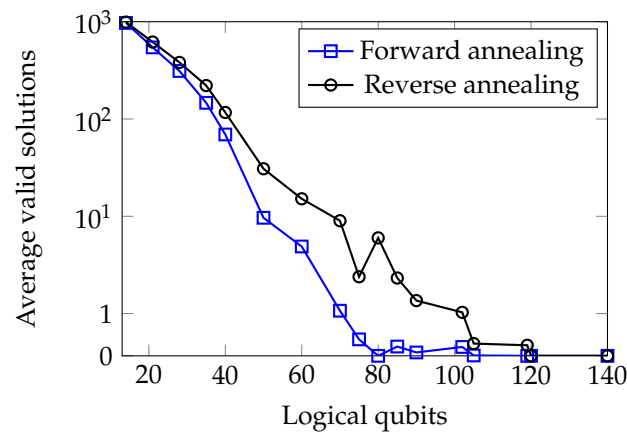

Figure 10. Average valid solutions obtained when using forward and reverse annealing with a reverse distance of 0.44 across 900 randomly generated problems.
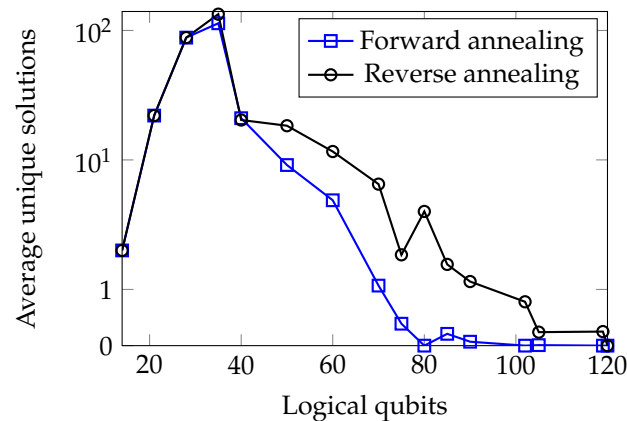

Figure 11. Same as Figure 10 except the y-axis shows average of unique valid solutions.

In conclusion, from Figures 10 and 11, we infer that there exists a range of problem sizes for which one can expect forward annealing to find no valid solutions, but feeding the results to reverse annealing yields valid solutions. It is in this sense that reverse annealing extends the computational reach of quantum annealers for the graph coloring problem.

4.3. Random Initial State

Since reverse annealing accepts any bitstring as initial state, one may question the need for first performing forward annealing as outlined in our algorithm. We now compare our idea of using the lowest-energy bitstring against using a random bitstring as an initial state. Additionally, we repeat our experiments on a different device to confirm that the algorithm is not device dependent.

For each of the 262 problems from Section 4.1.3 Case C, we repeat the experiment only for those values of the reverse distance for which reverse annealing was able to find at least one valid solution with a slight change that the initial state is now a random bitstring. This allows us to save significant computing time on the real hardware. The data are shown in Figure 12. We plot the average number of valid solutions obtained for cases when using a random initial state in red circles. Next, we plot the average number of valid solutions when using the lowest energy (invalid solution) bitstring from forward annealing in green squares. Next, we repeat the green squares case with 3000 samples on a different annealer, Advantage System 6.2.
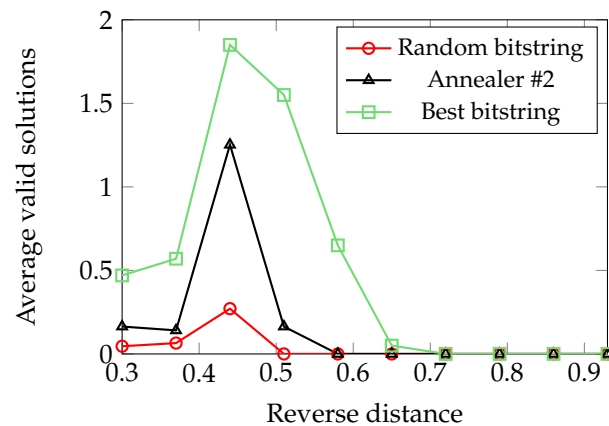
Figure 12. Average valid solutions obtained for 262 problems as a function of the reverse distance. Squares represent the case of using the lowest-energy bitstring from forward annealing as the initial state. Triangles represent the use of another device. Circles represent the case where the initial state was a random bitstring.

We observe that in a small number of cases, starting from a random initial state, the device is still able to find a valid solution for small reverse distances. In contrast, starting from the lowest energy bitstring from forward annealing yields significantly higher averages over a wider region of reverse distances. Additionally, we observe that a different annealer is able to reproduce some results but does not perform better despite a higher number of samples.

The lower performance of a different device can be explained as follows. In solving a given problem, several device parameters are involved, e.g., the annealing time, connectivities between the qubits, embedding, as well as chain strength. One set of such parameters that yields higher frequency of valid solutions on one device need not necessarily yield similar results on another device. We used the same parameters for both devices; thus, there may exist other parameters for the second device that yield a higher frequency of valid solutions. However, finding these parameters is beyond the scope of the current work.

In summary, we learned that there is a significant benefit in first performing forward annealing and choosing the lowest energy bitstring compared to using a bitstring created at random.

5. Conclusions

Finding appropriate initial states for reverse annealing to improve is, in general, an important and difficult task. We proposed a heuristic where the output of forward annealing was used as input for reverse annealing. We found that this setup is particularly useful for the graph coloring problem. We demonstrated our approach by solving randomly generated graph coloring problems. We found that reverse annealing is able to find valid solutions using this technique in about 57% of the cases where forward annealing fails. Thus, one is able to solve many more problems outside the reach of conventional forward annealing with the same quantum annealer.

We performed a scaling analysis where we increased the problem size and used both forward and the new forward-assisted reverse annealing to solve the same problem. Our results showed an exponential decay in the probability to sample valid solutions for both approaches. However, the new approach decayed slower and, thus, is able to extend the computational reach of the overall heuristic of quantum annealing. The new heuristic proposed in this work is general and can be used for any problem that uses forward annealing.

While we showed the working of the new approach, our results do not reflect best effort, which would include finding the most optimal annealing times, reverse distances, chain strengths, embeddings, annealing schedules, and other relevant parameters. These

should help obtain valid solutions in those cases where reverse annealing was also not able to find a valid solution. This can be explored in future works.

References

1. Finnila, A.B.; Gomez, M.A.; Sebenik, C.; Stenson, C.; Doll, J.D. Quantum annealing: A new method for minimizing multidimensional functions. *Chem. Phys. Lett.* **1994**, *219*, 343–348. [CrossRef]
2. Harris, R.; Johnson, M.W.; Lanting, T.; Berkley, A.J.; Johansson, J.; Bunyk, P.; Tolkacheva, E.; Ladizinsky, E.; Ladizinsky, N.; Oh, T.; et al. Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor. *Phys. Rev. B* **2010**, *82*, 024511. [CrossRef]
3. King, A.D.; Nisoli, C.; Dahl, E.D.; Poulin-Lamarre, G.; Lopez-Bezanilla, A. Qubit spin ice. *Science* **2021**, *373*, 576–580. [CrossRef] [PubMed]
4. Ohzeki, M. Breaking limitation of quantum annealer in solving optimization problems under constraints. *Sci. Rep.* **2020**, *10*, 3126. [CrossRef]
5. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.S.L.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [CrossRef] [PubMed]
6. Bruzewicz, C.D.; Chiaverini, J.; McConnell, R.; Sage, J.M. Trapped-ion quantum computing: Progress and challenges. *Appl. Phys. Rev.* **2019**, *6*, 021314. [CrossRef]
7. Gyongyosi, L.; Imre, S. A Survey on quantum computing technology. *Comput. Sci. Rev.* **2019**, *31*, 51–71. [CrossRef]
8. Jattana, M.S. Quantum annealer accelerates the variational quantum eigensolver in a triple-hybrid algorithm. *Physica Scripta* **2024**, *99*, 095117. [CrossRef]
9. McClean, J.R.; Romero, J.; Babbush, R.; Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **2016**, *18*, 23023. [CrossRef]
10. Jattana, M.S.; Jin, F.; De Raedt, H.; Michielsen, K. Improved Variational Quantum Eigensolver Via Quasidynamical Evolution. *Phys. Rev. Appl.* **2023**, *19*, 024047. [CrossRef]
11. Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J.M.; Gambetta, J.M. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **2017**, *549*, 242–246. [CrossRef] [PubMed]
12. Jattana, M.S.; Jin, F.; De Raedt, H.; Michielsen, K. Assessment of the Variational Quantum Eigensolver: Application to the Heisenberg Model. *Front. Phys.* **2022**, *10*, 907160. [CrossRef]
13. Willsch, D.; Jattana, M.S.; Willsch, M.; Schulz, S.; Jin, F.; De Raedt, H.; Michielsen, K. Hybrid Quantum Classical Simulations. *arXiv* **2022**, arXiv:2210.02811.
14. Ohkuwa, M.; Nishimori, H.; Lidar, D.A. Reverse annealing for the fully connected p-spin model. *Phys. Rev. A* **2018**, *98*, 022314. [CrossRef]
15. Kadowaki, T.; Nishimori, H. Quantum annealing in the transverse Ising model. *Phys. Rev. E* **1998**, *58*, 5355–5363. [CrossRef]
16. Santoro, G.E.; Martoňák, R.; Tosatti, E.; Car, R. Theory of Quantum Annealing of an Ising Spin Glass. *Science* **2002**, *295*, 2427–2430. [CrossRef]
17. Perdomo-Ortiz, A.; Venegas-Andraca, S.E.; Aspuru-Guzik, A. A study of heuristic guesses for adiabatic quantum computation. *Quantum Inf. Process.* **2011**, *10*, 33–52. [CrossRef]
18. Pelofske, E.; Hahn, G.; Djidjev, H. Initial state encoding via reverse quantum annealing and h-gain features. *arXiv* **2023**, arXiv:2303.13748.
19. Venturelli, D.; Kondratyev, A. Reverse quantum annealing approach to portfolio optimization problems. *Quantum Mach. Intell.* **2019**, *1*, 17–30. [CrossRef]
20. Yamashiro, Y.; Ohkuwa, M.; Nishimori, H.; Lidar, D.A. Dynamics of reverse annealing for the fully connected p-spin model. *Phys. Rev. A* **2019**, *100*, 052321. [CrossRef]
21. Chancellor, N. Modernizing quantum annealing using local searches. *New J. Phys.* **2017**, *19*, 023024. [CrossRef]
22. Arai, S.; Ohzeki, M.; Tanaka, K. Mean field analysis of reverse annealing for code-division multiple-access multiuser detection. *Phys. Rev. Res.* **2021**, *3*, 033006. [CrossRef]

23. Pelofske, E.; Bärtschi, A.; Eidenbenz, S. Simulating Heavy-Hex Transverse Field Ising Model Magnetization Dynamics Using Programmable Quantum Annealers. *arXiv* **2023**, arXiv:2311.01657.
24. Golden, J.; O'Malley, D. Reverse annealing for nonnegative/binary matrix factorization. *PLoS ONE* **2021**, *16*, e0244026. [CrossRef]
25. Ikeda, K.; Nakamura, Y.; Humble, T.S. Application of Quantum Annealing to Nurse Scheduling Problem. *Sci. Rep.* **2019**, *9*, 12837. [CrossRef]
26. Imoto, T.; Susa, Y.; Miyazaki, R.; Kadowaki, T.; Matsuzaki, Y. Demonstration of the excited-state search on the D-wave quantum annealer. *arXiv* **2023**, arXiv:2305.15974.
27. Haba, R.; Ohzeki, M.; Tanaka, K. Travel time optimization on multi-AGV routing by reverse annealing. *Sci. Rep.* **2022**, *12*, 17753. [CrossRef] [PubMed]
28. Kim, M.; Singh, A.K.; Venturelli, D.; Kaewell, J.; Jamieson, K. X-ResQ: Reverse Annealing for Quantum MIMO Detection with Flexible Parallelism. *arXiv* **2024**, arXiv:2402.18778.
29. Bando, Y.; Yip, K.W.; Chen, H.; Lidar, D.A.; Nishimori, H. Breakdown of the Weak-Coupling Limit in Quantum Annealing. *Phys. Rev. Appl.* **2022**, *17*, 054033. [CrossRef]
30. Rocutto, L.; Destri, C.; Prati, E. Quantum Semantic Learning by Reverse Annealing of an Adiabatic Quantum Computer. *Adv. Quantum Technol.* **2021**, *4*, 2000133. [CrossRef]
31. King, J.; Mohseni, M.; Bernoudy, W.; Fréchette, A.; Sadeghi, H.; Isakov, S.V.; Neven, H.; Amin, M.H. Quantum-Assisted Genetic Algorithm. *arXiv* **2019**, arXiv:1907.00707.
32. Garey, M.R.; Johnson, D.S.; Stockmeyer, L. Some Simplified NP-Complete Problems. In Proceedings of the STOC'74: Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, 30 April–2 May 1974; Association for Computing Machinery: New York, NY, USA, 1974.
33. Garey, M.R.; Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*; W. H. Freeman & Co.: New York, NY, USA, 1990.
34. NetworkX Developer Team. NetworkX. 2014. Available online: https://networkx.org/ (accessed on 5 May 2023).
35. Tamura, Y.; Sakai, M. Tamuhey/Qubogen. Available online: https://github.com/tamuhey/qubogen (accessed on 5 May 2023).
36. Glover, F.; Kochenberger, G.; Hennig, R.; Du, Y. Quantum bridge analytics I: A tutorial on formulating and using QUBO models. *Ann. Oper. Res.* **2022**, *314*, 141–183. [CrossRef]
37. D-Wave Leap. Available online: https://www.dwavesys.com/take-leap (accessed on 5 May 2023).
38. Kwok, J.; Pudenz, K. Graph Coloring with Quantum Annealing. *arXiv* **2020**, arXiv:2012.04470.
39. Chen, J.; Stollenwerk, T.; Chancellor, N. Performance of Domain Wall Encoding for Quantum Annealing. *IEEE Trans. Quantum Eng.* **2021**, *2*, 1–14. [CrossRef]
40. Seki, Y.; Nishimori, H. Quantum annealing with antiferromagnetic fluctuations. *Phys. Rev. E* **2012**, *85*, 051112. [CrossRef]
41. Pelofske, E.; Hahn, G.; Djidjev, H.N. Advanced anneal paths for improved quantum annealing. In Proceedings of the 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), Denver, CO, USA, 12–16 October 2020; pp. 256–266. [CrossRef]
42. King, A.D.; Carrasquilla, J.; Raymond, J.; Ozfidan, I.; Andriyash, E.; Berkley, A.; Reis, M.; Lanting, T.; Harris, R.; Altomare, F.; et al. Observation of topological phenomena in a programmable lattice of 1,800 qubits. *Nature* **2018**, *560*, 456–460. [CrossRef]
43. Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2014**, *2*, 5. [CrossRef]
44. Tabi, Z.; El-Safty, K.H.; Kallus, Z.; Hága, P.; Kozsik, T.; Glos, A.; Zimborás, Z. Quantum Optimization for the Graph Coloring Problem with Space-Efficient Embedding. In Proceedings of the 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), Denver, CO, USA, 12–16 October 2020; pp. 56–62. [CrossRef]
45. Śmierzchalski, T.; Mzaouali, Z.; Deffner, S.; Gardas, B. Efficiency optimization in quantum computing: Balancing thermodynamics and computational performance. *Sci. Rep.* **2024**, *14*, 4555. [CrossRef] [PubMed]
46. Hagberg, A.A.; Schult, D.A.; Swart, P.J. Exploring Network Structure, Dynamics, and Function using NetworkX. In Proceedings of the 7th Python in Science Conference, Pasadena, CA, USA, 21 August 2008; Varoquaux, G., Vaught, T., Millman, J., Eds.; U.S. Department of Energy: Washington, DC, USA, 2008; pp. 11–15.
47. Jattana, M.S. Modular Supercomputing and Quantum Computing. Goethe University Data Repository. 2024. Available online: https://gude.uni-frankfurt.de/entities/researchdata/57431ed0-f2df-4e4b-aec8-6df63325b34c/details (accessed on 25 July 2024).