

# Optimizing the spin reversal transform on the D-Wave 2000Q

Elijah Pelofske      Georg Hahn      Hristo Djidjev

Los Alamos National Laboratory

## Abstract

Commercial quantum annealers from D-Wave Systems make it possible to obtain approximate solutions of high quality for certain NP-hard problems in nearly constant time. Before solving a problem on D-Wave, several preprocessing methods can be applied, one of them being the so-called spin reversal or gauge transform. The spin reversal transform flips the sign of selected variables and coefficients of the Ising or QUBO (quadratic unconstrained binary optimization) representation of the problem that D-Wave minimizes. The spin reversal transform leaves the ground state of the Ising model invariant, but can average out the biases induced through analog and systematic errors on the device, thus improving the quality of the solution that D-Wave returns. This work investigates the effectiveness of the spin reversal transform for D-Wave 2000Q. We consider two important NP-hard problems, the Maximum Clique and the Minimum Vertex Cover problems, and show on a variety of input problem graphs that using the spin reversal transform can yield substantial improvements in solution quality. In contrast to the native spin reversal built into D-Wave, we consider more general ways to reverse individual spins and we investigate the dependence on the problem type, on the spin reversal probability, and possible advantages of carrying out reversals on the qubit instead of the chain level. Most importantly, for a given individual problem, we use our findings to optimize the spin reversal transform using a genetic optimization algorithm.

## 1 Introduction

Commercial quantum computers from D-Wave Systems Inc. D-Wave Systems (2000) are designed to approximately solve NP-hard optimization problems that can be expressed as the minimization of a QUBO (quadratic unconstrained binary optimization) or an Ising problem, given by

$$H(x_1, \dots, x_n) = \sum_{i=1}^n a_i x_i + \sum_{i < j} a_{ij} x_i x_j, \quad (1)$$

using a process called *quantum annealing*. In (1), the coefficients  $a_i \in \mathbb{R}$  are the linear weights, and  $a_{ij} \in \mathbb{R}$  are the quadratic couplers defining the problem for  $i, j \in \{1, \dots, n\}$ . If  $x_i \in \{0, 1\}$ , (1) is called a QUBO problem. If  $x_i \in \{-1, +1\}$ , (1) is called an Ising problem. Both the QUBO and Ising formulations are equivalent Djidjev et al. (2016). Many important NP-hard problems can be expressed as the minimization of a quadratic function of the form (1), see Lucas (2014). We denote the number of variables in the Ising model as  $n$  throughout the remainder of the article.

When submitting a problem of the form (1) to the D-Wave annealer, it is preprocessed in two ways. First, the qubits are arranged on the quantum annealer in a particular graph structure, called the *Chimera* graph, consisting of a lattice of bipartite cells on the physical chip Chapuis et al. (2017). However, the connectivity structure of a QUBO or Ising model, that is its nonzero couplers in (1), does not necessarily match the Chimera graph. In order to alleviate this issue, a

*minor embedding* of the QUBO or Ising connectivity to the Chimera graph can be computed. In such an embedding, several physical qubits are identified to act as one logical qubit in (1), often severely limiting the number of available qubits. The set of physical qubits on the chip representing a logical qubit (problem variable) is called a *chain*. Second, although  $a_i, a_{ij} \in \mathbb{R}$  can be arbitrary, they are rescaled to  $a_i \in [-2, 2]$  and  $a_{ij} \in [-1, 1]$  and subsequently converted to analog currents on the chip using an 8-bit digital-to-analog converter.

During the annealing, *leakage* on the physical chip from the coupler  $a_{ij}$  ( $i, j \in \{1, \dots, n\}$ ) can alter the linear weights  $a_i$  and  $a_j$  Systems (2019). This effect is reported to be often more serious for chained qubits. Moreover, during the process of digital-to-analog conversion, the linear weights  $a_i$  will not be perfectly mapped to currents on the chip, and thus biased to the above or below.

The so-called *spin reversal* or *gauge* transform is a simple way to alleviate this issue for Ising problems. The spin reversal is based on the observation that, although theoretically quantum annealing is invariant under a gauge transformation, the calibration of the D-Wave device is not perfect and breaks the gauge symmetry. This implies that, indeed, spin reversed Ising systems realize slightly different systems on the annealer, yielding different results which can be averaged. To apply the spin reversal, we flip the sign of an arbitrary number of variables and coefficients of the Ising problem, thus resulting in a reinterpretation of an *up* as a *down* spin and vice versa. This leaves the ground state of (1) invariant, but has the potential to reduce analog and systematic errors on the device as described earlier (by averaging them out), thus improving the quality of the solution.

In particular, to transform the qubit  $x_i$  from  $-1$  to  $+1$ , we define a new function  $H'$  with  $a'_i \rightarrow -a_i$  as well as  $a'_{ij} \rightarrow -a_{ij}$  and  $a'_{ji} \rightarrow -a_{ji}$  for all  $j \in \{1, \dots, n\}$ . We observe that the ground state energies of  $H$  and  $H'$  are identical, and that the minimum of  $H'$  is the minimum of  $H$  with the  $i$ -th variable having a flipped sign. As reported in Systems (2019), reversing too few spins leaves the Ising model almost unchanged, whereas applying the spin reversal transform to too many qubits likely results in many pairs of connected qubits being transformed, thus effectively leaving the corresponding quadratic couplers unchanged. In both cases, the spin reversal transform might only have little effect.

It is important to note that the spin reversal transform can be applied on two different levels: After embedding the Ising model to be solved onto the D-Wave architecture, the actual embedded problem that D-Wave solves is read and the spin reversal is applied to any qubit independently – this is referred to in the remainder of the article as *spin reversal on the qubit level*. Second, we can apply the spin reversal in such a way that the physical qubits in a chain (representing one logical qubit) are all either spin reversed or all left unchanged – we refer to this technique in the remainder of the article as *spin reversal on the chain level*.

The SAPI interface of D-Wave allows us to apply a built-in (simple) form of the spin reversal transform. It is controlled through the parameter `num_spin_reversal_transforms` ( $= N_s$ ) in the function `solve_ising`. In connection with the parameter `num_reads` ( $= N_r$ ) which specifies the total number of anneal readouts, D-Wave will generate  $N_s$  spin reversed Ising problems and obtain  $N_r/N_s$  readouts for each. The spin reserved Ising models are obtained by flipping each qubit independently with probability 0.5.

In this work, we aim to assess the effectiveness of the spin reversal transform in a more general way. The main contribution of this article is twofold. First, we evaluate the performance of the spin reversal transform: In particular, we apply it to both the raw Ising formulation and the embedded problem. We are furthermore interested in its performance as a function of the probability of flipping a single qubit. Lastly, we assess its performance on two different NP-hard graph problems, the Maximum Clique problem and the Minimum Vertex Cover problem, see Lucas (2014), which will be introduced later.

Second, we aim to use our findings to optimize the spin reversal transform in practice. For this we employ a genetic optimization algorithm which, for a given problem instance given as an Ising model, finds the set of qubits on which the spin reversal transform is most effective. For this we consider a separate binary indicator for each qubit (reversed or not reversed), and optimize over all  $n$  indicators to find the best configuration.

In this article, we provide a rigorous assessment of the effectiveness of the spin reversal transform, which to the best of our knowledge has not been presented in the literature previously. Existing work published in the literature does employ spin reversals, yet only as a tool to possibly enhance solutions, and only using the in-built D-Wave implementation. In King et al. (2015) the authors define a new metric, the *time-to-target* metric, as the time needed by classical solvers to match the results of a quantum annealer: for their experiments, the authors employ D-Wave’s native gauge transform, but it is left unclear what the contribution of the transform to the solution quality is. In Pudenz (2016), the author primarily evaluates several techniques to allocate weights to chains of qubits, as well as two ways of determining the final value of a chain. The effect of four spin reversals is also considered briefly, however no further statement is made on the spin reversal transform apart from the fact that it shows a slight performance gain on certain systems. In King and McGeoch (2014), the authors conclude that gauge transformations are more effective on difficult problems. However, in Boixo et al. (2014), the authors demonstrate that significant correlations exist between different gauge transforms.

The article is structured as follows. Section 2 describes the spin reversal transform in detail, and presents a genetic algorithm to attempt to solve the optimization over all possible spin reversal transformations. Section 3 presents simulation results for the Erdős-Rényi graph family, both as a function of the input graph density as well as of the probability of the spin reversal transform, and for two NP-hard graph problems. We also present results highlighting the dependence of the genetic optimization algorithm on its parameters, and show how the effectiveness of the spin reversal transform can be considerably increased in comparison to the D-Wave transformation over the course of only a few ‘genetic’ generations. The article concludes with a discussion in Section 4.

## 2 The Gauge Transform

This section describes the setup we employ to apply the spin reversal transform.

### 2.1 Spin reversal on the qubit level

Given an input Ising model of type (1), we embed it onto the D-Wave Chimera graph first.

Before starting the annealing process, however, we read the embedded Ising model from the D-Wave connectivity graph. The embedded problem typically consists of more variables, precisely the physical chain qubits representing the logical qubits.

Given the embedded Ising model on the D-Wave chip with  $n$  qubits, we can reverse spins in the following way. Select a set  $I \subseteq \{1, \dots, n\}$  of qubit indices to be switched. In our experiments, we will generate the set  $I$  by adding each index in  $\{1, \dots, n\}$  to it independently with a given probability  $p_s$ .

We then sequentially select one  $i \in I$  at a time, and set  $a'_i := -a_i$ . Furthermore, we set  $a'_{ij} := -a_{ij}$  and  $a_{ji} := -a_{ji}$  for all  $j \in \{1, \dots, n\}$ . After having applied this procedure for all  $i \in I$ , we use the new sets of linear weights  $\{a'_i\}$  and quadratic couplers  $\{a'_{ij}\}$  to form a new Ising problem  $H'$ .

The new  $H'$  is then embedded onto the D-Wave chip (instead of the originally embedded problem) and solved.

---

**Algorithm 1:** Genetic algorithm for spin reversal tuning

---

**input** :  $H, N, p_{\text{spin}}, p_{\text{mat}}, p_{\text{mut}}, R, N_a$ ;  
**output**: final population of spin reversal vectors;

- 1  $n \leftarrow$  number of variables in  $H$ ;
- 2  $S \leftarrow \{s_1, \dots, s_N : s_i \in \mathbb{B}_n^{p_{\text{spin}}}\}$ ;
- 3 **repeat**  $R$  **times**
- 4     **for**  $s \in S$  **do**
- 5          $H' \leftarrow$  reverse spins  $s$  in  $H$ ;
- 6         Request  $N_a$  anneals for  $H'$  on D-Wave and store minimal energy among those in  $e_s$ ;
- 7     **end**
- 8      $E \leftarrow \{e_s : s \in S\}$ ;
- 9      $E_0 \leftarrow$  proportion  $p_{\text{mat}}$  of lowest energies in  $E$ ;
- 10     $S_0 \leftarrow \{s \in S : e_s \in E_0\}$ ;
- 11     $S_1 \leftarrow \emptyset$ ;
- 12    **repeat**  $N$  **times**
- 13         Draw two random  $s_1, s_2 \in S_0$  and combine bits randomly with probability 0.5, store result in  $S_1$ ;
- 14    **end**
- 15     $S_2 \leftarrow \emptyset$ ;
- 16    **for**  $s \in S_1$  **do**
- 17         Flip each bit in  $s$  independently with probability  $p_{\text{mut}}$  and store mutated  $s$  in  $S_2$ ;
- 18    **end**
- 19     $S \leftarrow S_2$ ;
- 20 **end**
- 21 **return**  $S$ ;

---

## 2.2 Spin reversal on the chain level

Another way of applying the spin reversal is at the chain level. In contrast to Section 2.1, we read the embedded problem and switch the signs of either all physical qubits in a chain that encode one logical qubit, or of none of them.

## 2.3 Optimizing the spin reversal transform

Algorithm 1 presents the genetic algorithm we use to optimize the spin reversal transform for the D-Wave annealer. Let an input Ising model  $H$  with  $n$  variables be given which we aim to solve on D-Wave. In the following, we characterize a spin reversal through a boolean vector of length  $n$ , where each *True* entry encodes that the corresponding variable is reversed.

Using Algorithm 1, we aim to find the spin reversal transform among all  $2^n$  possible binary vectors that yields the minimal average energy on D-Wave (within a pre-specified number of  $N_a$  anneals).

Algorithm 1 works as follows: We use a genetic algorithm to optimize over generations of boolean vectors (or bitstrings) of length  $n$  which are interpreted as spin reversals in the aforementioned sense. First, in line 2, we draw  $N$  random boolean vectors from  $\mathbb{B}_n^{p_{\text{spin}}}$ , the set of all bitstrings of length  $n$  generated with a probability of  $p_{\text{spin}}$  for an entry *True*. Those are stored in a set  $S$ .

Next, the initial population in  $S$  is evaluated with regards to their *fitness*, which in our case is the quality of the solution. For this, we apply the spin reversal to the variables indicated in each

$s \in S$ , resulting in a new Ising model  $H'$ , perform  $N_a$  anneals for  $H'$  on D-Wave, and record the minimal energy  $e_s$  obtained in this way for  $s$ . All those energies are stored in a set  $E$ . Afterwards, we compute a subset  $E_0$  of  $E$  corresponding to the proportion  $p_{\text{mat}}$  of lowest energies (those are the *fittest individuals* of the current population). We store the boolean strings that correspond to the energies in  $E_0$  in a set  $S_0$ .

Next, we carry out a *crossover* operation on the population in line 12. For this, we take two arbitrary  $s_1, s_2 \in S_0$  and combine them into a new boolean vector  $s_3$  by choosing the entries of  $s_3$  independently from  $s_1$  and  $s_2$  with probability 0.5. The new  $s_3$  is added to a set  $S_1$  which was initialized as  $S_1 = \emptyset$ . This is repeated  $N$  times in order to leave the population size invariant in each generation.

Finally, a *mutation* step is applied in line 16. We flip the entries of each  $s \in S_1$  independently with a low probability  $p_{\text{mut}}$  and store the resulting  $s$  in a set  $S_2$  (likewise initialized with  $S_2 = \emptyset$ ). After mutations have been applied to all  $s \in S_1$ , we restart the fitness evaluation on the updated set  $S_2$ .

We create  $R$  generations in this way.

After termination of Algorithm 1, we are left with the last population of boolean vectors which encode different spin reversals. In order to choose the optimal spin reversal for the Ising model  $H$ , we evaluate their fitness one last time: We can apply each to  $H$  (resulting in a new  $H'$  as before), record the lowest minimal energy for  $H'$  in  $N_a$  anneals on D-Wave, and choose the boolean vector among the entire population yielding the minimum energy solution.

### 3 Experiments on D-Wave

#### 3.1 Evaluation of the spin reversal transform

In the following two subsections, we evaluate the spin reversal transform on both the qubit and the chain level.

The test graphs we employ are  $G(|V|, p_G)$  Erdős–Rényi graphs with  $|V| = 65$  vertices and edge probability  $p_G \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ . The reason for using  $|V| = 65$  vertices stems from the fact that this is the largest size of an arbitrary graph that can be embedded onto D-Wave 2000Q, see Pelofske et al. (2019a,b).

In order to generate an Ising model to solve for each test graph, we consider two NP-hard graph problems: the Maximum Clique problem, which asks for a maximum fully-connected set of vertices of a graph, and the Minimum Vertex Cover problem, which asks for a minimum set of vertices in a graph such that each edge is incident to at least one vertex from that set. The QUBO/Ising formulations of those two problems in the form of (1) can be found in Lucas (2014).

The spin reversal transform requires the index set  $I$  which specifies the qubits to be spin-reversed, see Section 2.1. In the simulations of this section we generate the index sets with varying probabilities  $p_s \in \{0.01, 0.05, 0.1, 0.2, \dots, 0.8, 0.9, 0.95, 0.99\}$ , that is for each element in  $\{1, \dots, n\}$  we decide with an independent Bernoulli trial of probability  $p_s$  whether it should be included in  $I$  or not (where  $n$  is the number of variables in the Ising model, see (1)).

For each value of  $p_s$ , we generate 50 graphs, apply the two Ising model formulations for the Maximum Clique and Minimum Vertex Cover problems, and solve them using the following three techniques using  $N_a = 1000$  anneals on D-Wave: (a) we solve with D-Wave’s spin reversal parameter using  $N_s = 1$  (see Section 1); we will refer to this technique as *D-Wave’s native spin reversal*, (b) with spin reversal on the qubit level, and (c) with spin reversal on the chain level. For every batch of anneals, we report the best solution found as the mean among the 10% lowest energy solutions

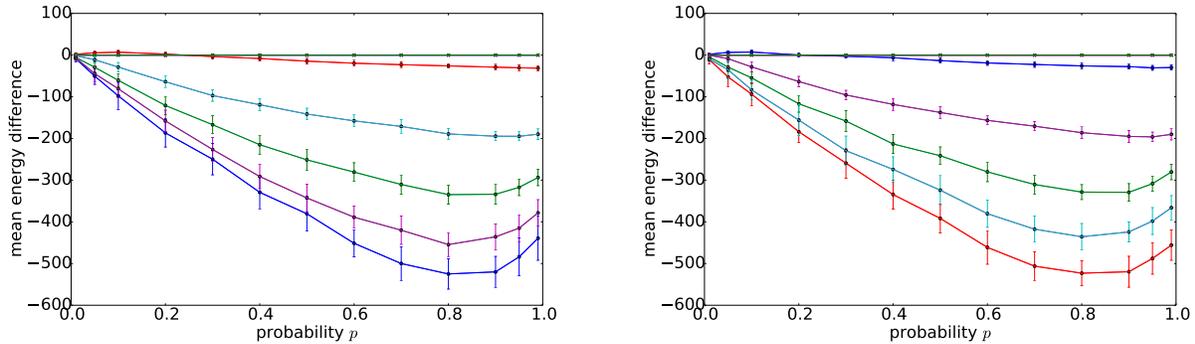


Figure 1: Mean energy difference of the spin reversal on qubit level of Section 2.1 to D-Wave’s native spin reversal (black line) as a function of  $p_s$ . Test graphs generated with edge probabilities 0.1 (blue), 0.3 (green), 0.5 (cyan), 0.7 (magenta) and 0.9 (red). Maximum Clique (left) and Minimum Vertex Cover (right) problems.

on D-Wave. This is to report a more robust measurement than merely recording the best solution found.

In this subsection, all lines in the figures are averages over 50 repetitions. For each repetition, we regenerate both the input graph, the Ising model, and its embedding on D-Wave.

### 3.1.1 Results for spin reversal on the qubit level

Figure 1 shows experimental results for a comparison of D-Wave’s native spin reversal (black line) to the more general approach of Section 2.1 as a function of the spin reversal probability  $p_s$  ( $x$ -axis). We use test graphs of varying density (blue to red) applied to the Maximum Clique (left) and Minimum Vertex Cover (right) problems. The two plots display the mean energy difference to D-Wave’s native spin reversal on the  $y$ -axis. We display error bars of one standard deviation for each measurement.

The figure shows that for both the Maximum Clique and the Minimum Vertex Cover problem, the approach of Section 2.1 almost always yields a solution of higher energy (worse quality) than the one using D-Wave’s native spin reversal. For the Maximum Clique problem, applying the spin reversal transform with a low probability  $p_s$  (roughly  $p_s \in [0.1, 0.5]$ ) yields marginally better solutions for high density graphs only (red line).

For the Minimum Vertex Cover problem, as being related to the complement of the Maximum Clique problem, the situation is reversed: Only for low density graphs (blue line), using a custom spin reversal probability of  $p_s \in [0.1, 0.5]$  marginally outperforms D-Wave’s native spin reversal.

The curves in Figure 1 exhibit a minimum (which happens to occur around  $p = 0.8$ ): Such a minimum is to be expected, since both very low and very high probabilities leave an Ising problem essentially unchanged (see Section 1).

### 3.1.2 Results for spin reversal on the chain level

Figure 2 shows results of a comparison of D-Wave’s spin reversal to the one on the chain level described in Section 2.2. As in Section 3.1.1, our approach of Section 2.2 is performing worse than D-Wave’s native spin reversal in almost all cases. The only exception seems to be the application of our spin reversal approach to high density graphs (Maximum Clique problem, left) or low density

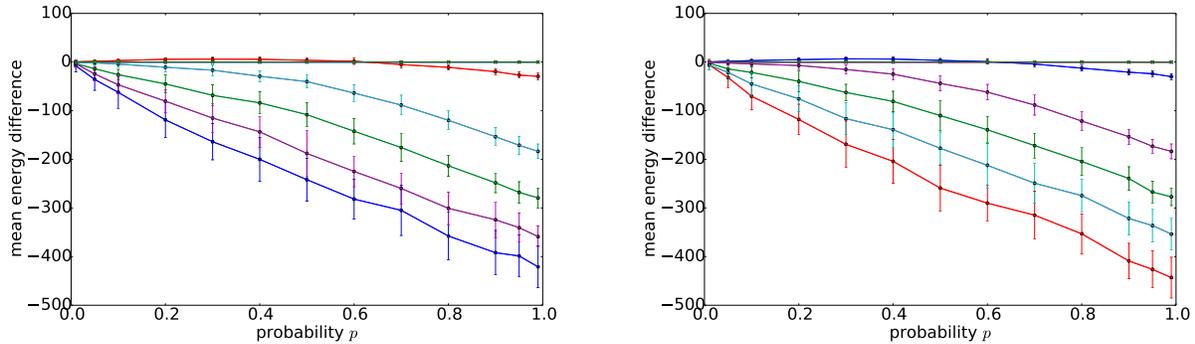


Figure 2: Mean energy difference of the spin reversal on chain level of Section 2.2 to D-Wave’s native spin reversal (black line) as a function of  $p_s$ . Test graphs generated with edge probabilities 0.1 (blue), 0.3 (green), 0.5 (cyan), 0.7 (magenta) and 0.9 (red). Maximum Clique (left) and Minimum Vertex Cover (right) problems.

graphs (Vertex Cover problem, right) in connection with a spin reversal probability  $p_s$  of around 0.1 to 0.6.

### 3.2 Results for our genetic algorithm

This section starts by assessing the dependence of the performance of Algorithm 1 on its parameters. For this, we first define a base set of parameters:  $N = 30$ ,  $p_{\text{spin}} = 0.5$ ,  $p_{\text{mat}} = 0.5$ , and  $p_{\text{mut}} = 0.01$ . When varying each of those four parameters, we keep the others fixed at their base values.

Figure 3 shows results for graphs of density 0.5 applied to the Ising model of the Maximum Clique problem. The generated graph and the embedding of the Maximum Clique Ising model onto the D-Wave architecture were kept fixed. Algorithm 1 was always run over  $R = 50$  generations. The spin reversal was applied at the chain level. We assess the dependence on  $N \in \{10, 30, 50\}$  (top left),  $p_{\text{spin}} \in \{0.1, 0.5, 0.9\}$  (top right),  $p_{\text{mat}} \in \{0.1, 0.5, 0.9\}$  (bottom left) and  $p_{\text{mut}} \in \{0.01, 0.1, 0.5\}$  (bottom right). In all subplots of Figure 3, the green lines consistently indicate the smallest value of the three investigated ones for each parameter, red indicates the medium value and blue the largest one.

Figure 3 shows that, as expected, larger values of  $N$  result in more diverse populations, which increases the probability of creating *fit* population members (in our case, those are the bitstrings indicating which qubits need to be spin reversed) resulting in spin transforms with a low energy. Likewise, the spin reversal probability  $p_{\text{spin}}$  (used to create the initial population) should be rather low, although high values of  $p_{\text{spin}}$  eventually result in the same low energy solution given enough generations are created. The mutation probability should likewise be chosen small, since large values have a tendency to add too much noise to good solutions. The rate  $p_{\text{mat}}$  specifying the proportion of fittest population members that are used in the crossover stage should similarly be small in order to only pass on the best candidates to the next generation.

Note that in Figure 3 (top right) assessing the dependence on the spin reversal probability  $p_{\text{spin}}$ , in contrast to the other three plots the blue, red and green lines do not start in the same point in generation zero. This can be explained as follows: Only the parameters  $p_{\text{spin}}$  and  $N$  appear in the initialization in line 2 of Algorithm 1. Whereas for large enough  $N$ , the lowest energy solution of the initial population stays roughly the same, the spin reversal probability  $p_{\text{spin}}$  affects the population and its overall fitness as a whole in generation zero.

We observe that in all subfigures of Figure 3, our genetic algorithm converges to nearly the same

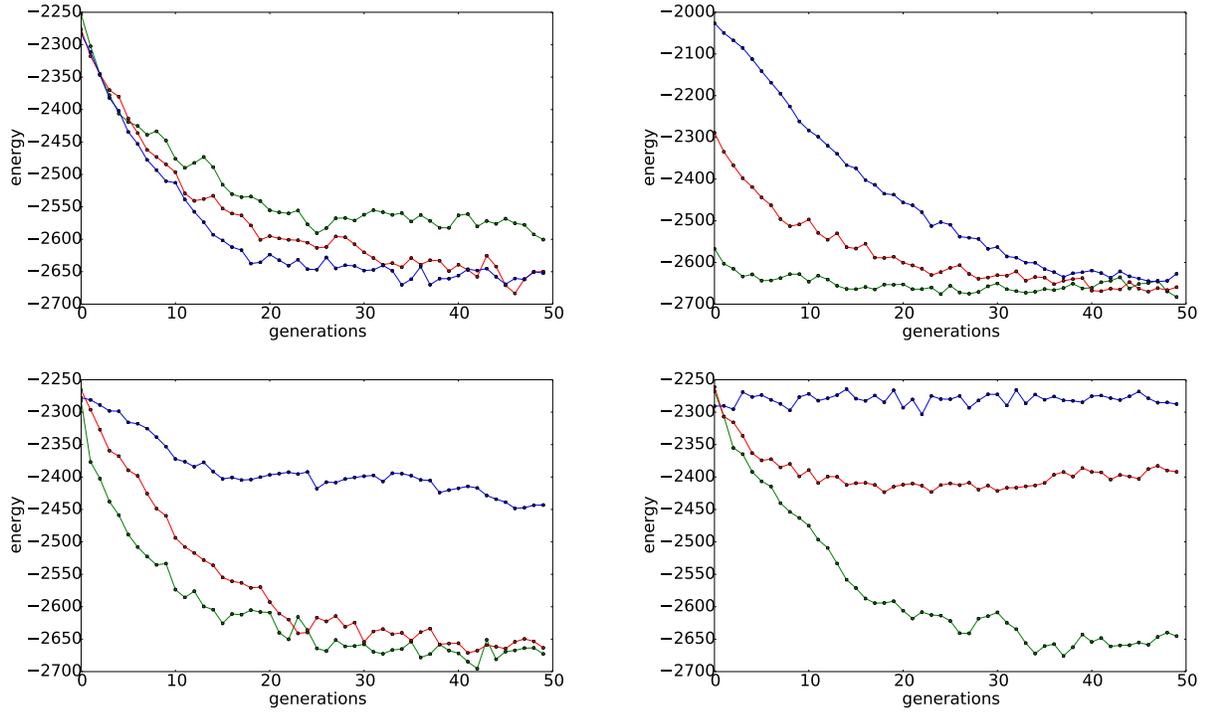


Figure 3: Dependence of Algorithm 1 on its parameters  $N \in \{10, 30, 50\}$  (top left),  $p_{\text{spin}} \in \{0.1, 0.5, 0.9\}$  (top right),  $p_{\text{mat}} \in \{0.1, 0.5, 0.9\}$  (bottom left) and  $p_{\text{mut}} \in \{0.01, 0.1, 0.5\}$  (bottom right) when applied to spin reversal at chain level. We consistently use green for the smallest value, red for the medium value, and blue for the largest parameter value in all subplots.

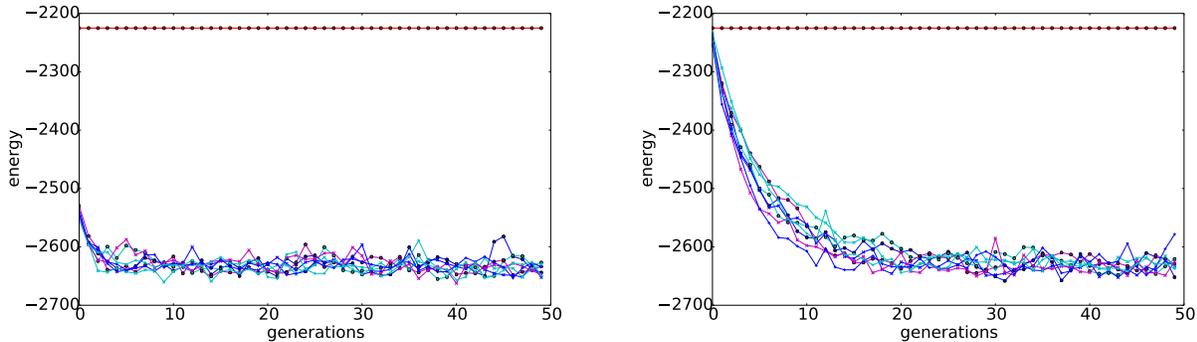


Figure 4: Comparison of D-Wave’s native spin reversal (red line) to Algorithm 1 with parameters given in (2), except for  $p_{\text{spin}} = 0.1$  (left) and  $p_{\text{spin}} = 0.5$  (right). Ising model created by applying the Maximum Clique problem to a graph of 65 vertices and graph density 0.5.

minimum energy for the best parameter choice, which is to be expected for a good optimization scheme.

Based on Figure 3, we will use Algorithm 1 in the remainder of the article with parameters

$$N = 50, p_{\text{spin}} = 0.1, p_{\text{mat}} = 0.1, p_{\text{mut}} = 0.01. \quad (2)$$

When using Algorithm 1 in connection with spin reversal at the qubit level, results are not as clear, see Figure 6 in Appendix A.

Using the parameters given in (2), we are able to employ Algorithm 1 to determine a (reasonably) good set of variables to be spin-reversed (with the aim of creating an equivalent Ising model, which yields lower minimal energies during annealing).

Figure 4 shows simulation results for D-Wave’s native spin reversal (red constant line) and 8 different realizations of Algorithm 1 over 50 generations. For each datapoint of the genetic algorithm, we plot the lowest 10% mean energy obtained when applying the current best set of variables to be spin reversed to the Maximum Clique Ising model.

We observe that for the set of parameters given in (2), after only a handful of generations (around  $R = 5$ ), Algorithm 1 has found a combination of variables in the Ising model, which, after spin reversal has been applied to them, yield a modified (but equivalent) Ising model with substantially lower energies during annealing for  $p_{\text{spin}} = 0.1$ . For  $p_{\text{spin}} = 0.5$ , we observe a slower rate with which our genetic algorithm reaches the minimum. However, both values cause Algorithm 1 to converge to roughly the same minimum.

In the run of Algorithm 1 depicted in Figure 4 (left), the initial population of spin reversal vectors happens to already yield lower energies than D-Wave’s native spin reversal. However, Figure 4 (right) shows that even when using an initial population containing spin reversal vectors which yield energies after annealing that are comparable to D-Wave’s native spin reversal, our genetic algorithm is capable of considerably improving upon D-Wave’s spin reversal within only a few generations.

### 3.3 Spatial correlation of spin reversed qubits on the D-Wave chip

It is of interest to investigate if there is a spatial correlation between those qubits for which Algorithm 1 determines that a spin reversal improves the solution quality.

For the Maximum Clique problem on a graph of 65 vertices and graph density 0.5, we apply Algorithm 1 using the optimized parameters determined in Section 3.2 and  $R = 50$  generations.



Figure 5: Qubit connectivity graph (the *Chimera* graph) of D-Wave 2000Q. Spin reversed qubits in red, non-reversed qubits in blue, and unused qubits in gray for the single current best solution of Algorithm 1 at initialization (left) and after 49 generations (right).

Figure 5 shows the qubit connectivity graph (the Chimera graph) of D-Wave 2000Q, where we color single current best solution of spin reversed qubits with red (qubit is reversed) and blue (qubit is not reversed). The gray qubits are unused.

Figure 5 shows that at initialization (left), the assignment is random. After 49 generations, the distribution of spin reversed qubits has changed: First, we observe that the proportion of spin reversed qubits has increased. Second, we observe that often, spin reversed qubits are adjacent to each other in a cell of the Chimera graph, or in adjacent cells facing each other. Though this finding was expected, the precise reason behind this can only be speculated. One explanation could be that qubits which are physically adjacent on the chip suffer from leakage (see Section 1), thus since the sign of all couplers  $J_{ij}$  related to a particular spin  $i$  are flipped, it is sensible to assume that such a manipulation of an Ising model also affects the other qubit  $j$ .

## 4 Conclusions

This paper investigated the spin reversal on the D-Wave 2000Q quantum annealer. In particular, we applied our own approach of a spin reversal which, in contrast to the native one built into D-Wave 2000Q, allows us to specify the probability with which spins are reversed, and we apply the spin reversal to the qubit and chain level. Importantly, we present a genetic algorithm to select the individual qubits which, if spin reversed, yield a modified Ising model that is easier to minimize (resulting in a minimum of better quality).

We summarize our findings as follows:

1. The native spin reversal on D-Wave 2000Q seems to be highly optimized for the annealer, even though D-Wave claims that its spin reversal is applied at a constant flipping probability of 0.5. In our experiments we did not manage to achieve lower energy solutions in the average

case except for rare instances involving very dense (for the Maximum Clique problem) or very sparse (for the Minimum Vertex Cover problem) graphs and spin reversal probabilities of  $p_s \leq 0.5$ .

2. We show using a genetic algorithm that learns the specific spins to be reversed is much more beneficial, and considerably improves upon D-Wave’s native spin reversal. Importantly, we show that typically only a handful of generations (around  $R = 5$ ) is sufficient to arrive at a combination of individual chained qubits which, if reversed, yield Ising models with solutions of substantially better quality. Therefore, increasing the computational effort (the number of anneals) by a constant factor (around  $R = 5$  generations with a population size of  $N \leq 50$  each) can yield substantially better solutions.
3. We empirically show that those qubits which our algorithm suggests for spin reversal are also physically adjacent on the D-Wave architecture.

The genetic algorithm we employ to optimize the spin reversal transform is a rather basic version, and much more sophisticated approaches exist in the literature. Future work includes the implementation and tuning of such advanced methods, with which we hope to find even better bitstrings of qubits to be spin reversed, and to decrease the required population size and number of generations, thus resulting in a reduced overhead when optimizing the spin transform.

Moreover, we are interested in investigating further if the shape of the curves in Figure 2, which is consistent even though the two Ising models and the input graphs they were applied to are different, can be explained by some intrinsic property of the Ising model.

## References

- Boixo, S., Rønnow, T. F., Isakov, S. V., Wang, Z., Wecker, D., Lidar, D. A., Martinis, J. M., and Troyer, M. (2014). Evidence for quantum annealing with more than one hundred qubits. *Nature Phys*, 10:218–224.
- Chapuis, G., Djidjev, H., Hahn, G., and Rizk, G. (2017). Finding Maximum Cliques on the D-Wave Quantum Annealer. *Proceedings of the 2017 ACM International Conference on Computing Frontiers (CF’17)*, pages 1–8.
- D-Wave Systems (2000). Quantum Computing for the Real World Today.
- Djidjev, H., Chapuis, G., Hahn, G., and Rizk, G. (2016). Efficient Combinatorial Optimization Using Quantum Annealing. *LA-UR-16-27928*. *arXiv:1801.08653*.
- King, A. D. and McGeoch, C. C. (2014). Algorithm engineering for a quantum annealing platform. *CoRR*, abs/1410.2628.
- King, J., Yarkoni, S., Nevisi, M. M., Hilton, J. P., and McGeoch, C. C. (2015). Benchmarking a quantum annealing processor with the time-to-target metric. *arXiv:1508.05087*, pages 1–29.
- Lucas, A. (2014). Ising formulations of many NP problems. *Front Phys*, 2(5):1–27.
- Pelofske, E., Hahn, G., and Djidjev, H. (2019a). Solving large maximum clique problems on a quantum annealer. *LA-UR-18-30973*. *arXiv:1901.07657*.
- Pelofske, E., Hahn, G., and Djidjev, H. (2019b). Solving large minimum vertex cover problems on a quantum annealer. *LA-UR-19-21008*. *arXiv:1904.00051*.

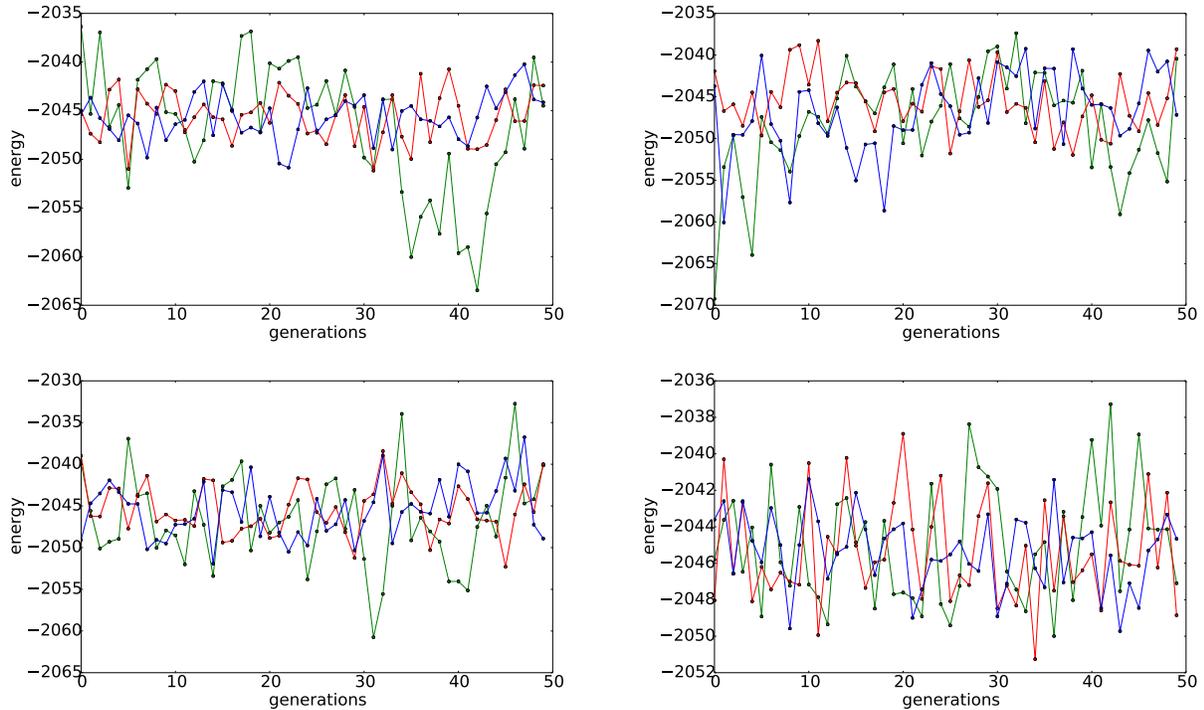


Figure 6: Dependence of Algorithm 1 on its parameters  $N \in \{10, 30, 50\}$  (top left),  $p_{\text{spin}} \in \{0.1, 0.5, 0.9\}$  (top right),  $p_{\text{mat}} \in \{0.1, 0.5, 0.9\}$  (bottom left) and  $p_{\text{mut}} \in \{0.01, 0.1, 0.5\}$  (bottom right) when applied to spin reversal at qubit level. We consistently use green for the smallest value, red for the medium value, and blue for the largest value in all subplots.

Pudenz, K. L. (2016). Parameter setting for quantum annealers. In *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–6.

Systems, D.-W. (2019). D-Wave System Documentation.

## A Further assessments of Algorithm 1 on qubit level

We repeat the assessments of Algorithm 1 in Section 3.2 for the spin reversal applied at qubit level. As in Section 3.2, we evaluate the dependence of the genetic algorithm on its parameters  $N$ ,  $p_{\text{spin}}$ ,  $p_{\text{mat}}$ , and  $p_{\text{mut}}$  while keeping those parameters which are not being varied at their values given in Section 3.2.

In contrast to the assessment at chain level in Section 3.2, Figure 6 shows that our genetic algorithm does not seem to converge after the initial burn-in phase. We explain this with the fact that when applying the spin reversal at qubit level, naturally, chains of qubits are not preserved, even though those represent the same logical qubit. The individual spin reversal within chains could therefore lead to more broken chains after annealing, and thus to an overall reduced solution quality. Additionally, qubit level optimization presents a significantly larger search space which must be optimized over, compared to the smaller quantity of chains which must be optimized.