EQUAL: Improving the Fidelity of Quantum Annealers by Injecting Controlled Perturbations

Ramin Ayanzadeh (ayanzadeh@gatech.edu)
Georgia Institute of Technology
Poulami Das
Georgia Institute of Technology
Swamit Tannu
University of Wisconsin—Madison
Moinuddin Qureshi
Georgia Institute of Technology

Research Article

Keywords:

Posted Date: March 2nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1343360/v1

EQUAL: Improving the Fidelity of Quantum Annealers by Injecting Controlled Perturbations

Ramin Ayanzadeh[1], Poulami Das[1], Swamit S. Tannu[2], and Moinuddin Qureshi[1]

[1]Georgia Institute of Technology, Atlanta, GA 30332, United States
[2]University of Wisconsin—Madison, Madison, WI 53706, United States
*ayanzadeh@gatech.edu

ABSTRACT

Quantum Annealers (QAs) are single-instruction quantum machines that can only sample from the ground state of an energy function, called Hamiltonian. To execute a program, the problem is cast to a Hamiltonian, embedded on the hardware, and a single quantum machine instruction (QMI) is run. Noise and imperfections in hardware result in sub-optimal solutions on QAs even if the QMI is run for thousands of trials. Owing to the limited programmability of QAs, users execute the same QMI for all trials. This subjects all trials to a similar noise profile throughout the execution, resulting in a *systematic bias*. We observe that systematic bias leads to sub-optimal solutions and cannot be alleviated by executing more trials or using existing error-mitigation schemes. To address this challenge, we propose *EQUAL* (Ensemble QUantum AnneaLing). EQUAL generates an ensemble of QMIs by adding controlled perturbations to the program QMI. When executed on the QA, the ensemble of QMIs steers the program away from encountering the same bias during all trials and thus, improves the quality of solutions. Our evaluations using the D-Wave 2000Q machine show that EQUAL bridges the difference between the baseline and the ideal by an average of 14% (and up to 26%), without requiring any additional trials. EQUAL can be combined with existing error mitigation schemes to bridge further the difference between the baseline and ideal by an average of 55% (and up to 68%).

1 Introduction

Quantum computing is an information processing paradigm that leverages quantum mechanical properties of quantum bits (qubits) to store and process information and promises significant computational advantages for many hard problems[1–4]. There exist different models for the physical realization of this computational paradigm, such as gate (or circuit) model quantum computing, adiabatic quantum computing, and measurement based quantum computing[5,6]. *Quantum Annealers (*QAs*)* are a special (or restricted) case of the adiabatic quantum computers. Unlike gate model quantum computers that can be programmed to solve different problem classes, QAs are single-instruction quantum machines that can only solve a specific discrete optimization problem by sampling from the ground state of a physical system energy function, called *Hamiltonian*[6,7]. To solve a problem on a QA, (a) we cast it to a Hamiltonian, (b) embed it to match the topology of the QA device, (c) obtain the resulting single *Quantum Machine Instruction (*QMI*)*, (d) execute the single QMI, and (e) repeatedly run the same QMI multiple times[8]. The outcome with the lowest energy value is deemed as the solution.

QAs available today with 5,000-plus qubits[9–11] are much larger, scale faster , and have the potential to power a wide range of real-world applications—including, but not limited to, planning[12], scheduling[13,14], constraint satisfaction problems[15], Boolean satisfiability (SAT)[16,17], matrix factorization[18], cryptography[19,20], compressive sensing[21,22], control of automated vehicles[23], finance[24], material design[25], and protein folding[26]. Although promising, QA hardware suffers from various drawbacks such as noise, device errors, limited programmability, and confined annealing schedule, which degrade their reliability[6,8,27]. Addressing these limitations requires device-level enhancements that may span generations of QAs. Therefore, leveraging software techniques to improve the reliability of QAs is an important area of research[28–33].

The limited programmability of QAs forces users to run a single QMI for thousands of trials, resulting in a bias. As a user runs a single QMI for all trials, the noise profile is similar throughout execution, resulting in similar quality outcomes due to the inherent bias in the noise profile. We refer to this bias as *Systematic Bias*. In this paper, we propose *Ensemble Quantum Annealing (*EQUAL*)*, an effective scheme for mitigating systematic bias and improving the reliability of QAs by running an ensemble of QMIs with controlled perturbations. EQUAL is based on the insight that running the same QMI for all trials projects QAs to a very similar noise profile and bias. On this basis, EQUAL uses an ensemble of QMIs that subjects the system to different noise profiles and biases. Generating effective ensembles of QMIs is nontrivial, and our design focuses on addressing it. We also propose EQUAL+, which exploits the properties of existing error mitigation schemes for enhanced performance.

Systematic bias in QAs is similar to correlated errors on gate-based quantum computers. To tackle these errors on gate-model quantum computers, recent studies propose the use of an ensemble of mappings that maps a program to different sets of physical qubits and SWAP routes on the same[34] or different machines[35]. This process produces functionally identical copies of the same program but is only executed differently. Leveraging a similar approach for QAs is nontrivial due to the complexities involved in the embedding process, particularly for problems at scale. Obtaining alternate embeddings is nontrivial and may fail or result in inferior quality. Instead, EQUAL uses an ensemble of QMIs by introducing controlled perturbations while minimizing the alterations in the functionality of the original problem Hamiltonian.

Using ensembles in QAs has been investigated at the casting level for two different applications[17,22]. However, as each application uses its own casting algorithm, this approach cannot be generalized. On the other hand, EQUAL avoids such application-specific assumptions and is applicable irrespective of the problem at hand. In another study, Mohseni et al. proposed a multi-level embedding scheme that uses a diverse encoding of qubits to generate ensembles[36]. However, this approach reduces the capacity of the QA significantly. It is also not scalable as it introduces overheads to the embedding step, which can already take several hours for current systems. Moreover, our studies show that finding alternate embeddings frequently fail or result in embeddings of inferior quality for large problems. EQUAL avoids these overheads by introducing diversity post embedding.

Overall, this paper makes the following contributions:

1. We show that there is a systematic bias associated with each QMI, running on QA, that deviates the annealing process from achieving the ground state of the corresponding Hamiltonian and produces sub-optimal solutions.

2. We propose *EQUAL (Ensemble Quantum Annealing)* to mitigate the bias by forming multiple perturbed copies of a given QMI and running each for a subset of trials.

3. We propose an effective method to generate the perturbed copies while retaining the structure of the problem by leveraging the hardware imperfections from limited precision.

4. We propose EQUAL+ that combines EQUAL with existing single-qubit correction (SQC)[27] error mitigation technique to improve the reliability further.

2 Results

Quantum Annealing is a meta-heuristic for solving combinatorial optimization problems that runs on classical computers[7,37–41]. Quantum annealers (QAs) are single-instruction quantum machines that can only sample from the ground state of Hamiltonians. QAs from D-Wave Systems are analog systems that can only sample from the ground state of the following (stoquastic) Hamiltonian:

$$\mathscr{H}_p := f(\mathbf{z}) = \sum_{i=0}^{N-1} \mathbf{h}_i \mathbf{z}_i + \sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} J_{ij} \mathbf{z}_i \mathbf{z}_j, \tag{1}$$

where N is the number of qubits, $\mathbf{h}_i \in \mathbb{R}$ specifies the linear coefficient of qubit i, $J_{i,j} \in \mathbb{R}$ represents the coupler weight between qubits i and j, and $\mathbf{z}_i$ is the variable that can take its value from $\{-1,+1\}$[8,17,27].

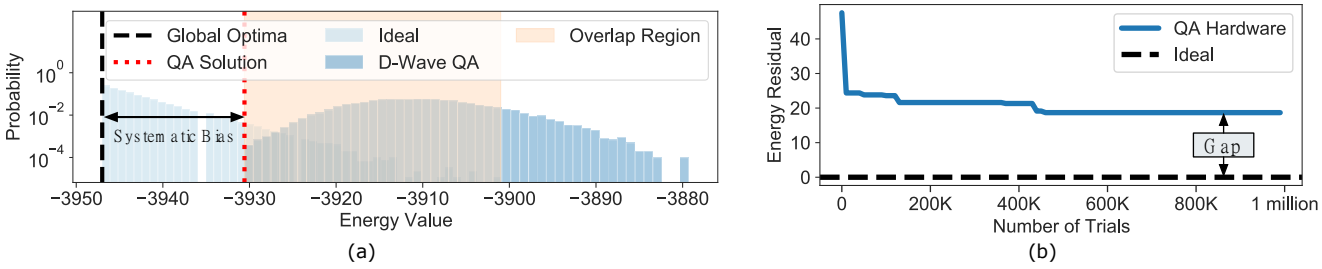

Figure 1. (a) Energy histogram of a 2000-qubit optimization benchmark executed on D-Wave QA (on logscale). The QA can quickly identify the region of the ground state energy (overlapping region), but the solution is far from the global optima due to systematic bias. (b) Energy Residual (ER) of an optimization problem on D-Wave QA with an increasing number of trials.

Figure 1(a) shows the energy histogram of a random benchmark problem (on the log scale) on a D-Wave 2000Q machine. We can think of QA as a machine that samples from a Boltzmann distribution such that samples with lower energy values, according to $f(\mathbf{z})$, are exponentially more likely to be observed[27,42]. In theory, therefore, QA can find the optimal solution

with a very high probability[41]. However, Figure 1(a) depicts that the energy histogram of the drawn samples by a physical QA follows the Gaussian distribution (expected to follow the Boltzmann distribution), and the resulting distribution has been shifted away from the ideal distribution—we refer to it as *systematic bias*. As users run only a single QMI, the program is subjected to a similar noise profile for all trials, resulting in a systematic bias. Figure 1(b) shows the *Energy Residual* (ER)[43,44] for an optimization problem on D-Wave QA. ER compares the gap between the energy of the solution from a noisy QA and the global minima. The energy of the best solution from a noisy QA remains far from the global optima even after running 1 million trials. This non-zero ER occurs due to systematic bias and is particularly severe for large problems.

2.1 EQUAL: Ensemble Quantum Annealing

The vulnerability of a program to systematic bias results from limited programmability and the current execution model of QAs where the same QMI is executed for thousands of trials. This subjects each trial to a similar noise profile on the QA, and the entire execution suffers from the same inherent bias. Our proposed solution *EQUAL* takes a different approach. Instead of sampling from the ground state of a single Hamiltonian, EQUAL generates an ensemble of Hamiltonians and executes their corresponding QMIs that subject the program execution to different noise profiles and, therefore, different systematic biases. If we aggregate these (ensembled) Hamiltonians (to form an aggregated Hamiltonian) and then run the resulting QMI, we will again face a similar systematic bias. However, when results (i.e., samples by executing all QMIs on a physical QA) are aggregated, ensembles enable us to mitigate the systematic bias and improve the quality of solutions.

2.1.1 Challenges in Generating Ensembles in EQUAL

There is a potential to generate ensembles during any one of the three phases that a problem goes through before execution on a physical QA hardware: (1) casting, (2) embedding, and (3) QMI generation. Generating ensembles during the casting was previously studied in the context of Boolean satisfiability (SAT)[17] and binary compressive sensing[22] problems on QAs. Unfortunately, these methods exploit the features of the application-specific casting algorithms. Therefore, this approach has limited applicability and is hard to generalize for QAs. The other alternative approach is to use an ensemble of embeddings for a given problem. However, this approach also has its limitations. Firstly, finding the best embedding is an NP-hard problem in itself[28,29,32,45]. Secondly, current embedding schemes for QAs use several approximations and may or may not be able to determine an ensemble of embeddings of similar quality[28,29,32,45]. Our studies show that existing embedding algorithms often fail to find an adequate number of embeddings, particularly for problems at scale that require 2,000-+ qubits. Thirdly, even if it is possible to find multiple embeddings, they are often of inferior quality and require larger chains of physical qubits to represent a program qubit with higher connectivity. This makes the embedding significantly more vulnerable to noise compared to the best embedding. Thus, generating ensembles at the embedding step is nontrivial. Instead, EQUAL focuses on generating ensembles at the instruction-level and produces multiple QMIs.

2.1.2 Overview of Design

Figure 2 shows an overview of EQUAL. It relies on adding controlled perturbations to the original QMI. For each ensemble, EQUAL generates a *Perturbation Hamiltonian* (i.e., a constant Hamiltonian), denoted by δ. Each of these Perturbation Hamiltonians creates a new QMI when added to the original Hamiltonian. For example, if EQUAL generates m ensembles of QMIs, it generates m perturbation Hamiltonians, namely $\delta_1, \delta_2, \ldots, \delta_m$. The ensemble QMIs—$QMI_1$, QMI_2 to QMI_m—are obtained by adding the original Hamiltonian (say $\mathscr{H}$) and the respective perturbation Hamiltonians. In other words, the ensemble of QMIs now corresponds to the perturbed versions of the original Hamiltonian.

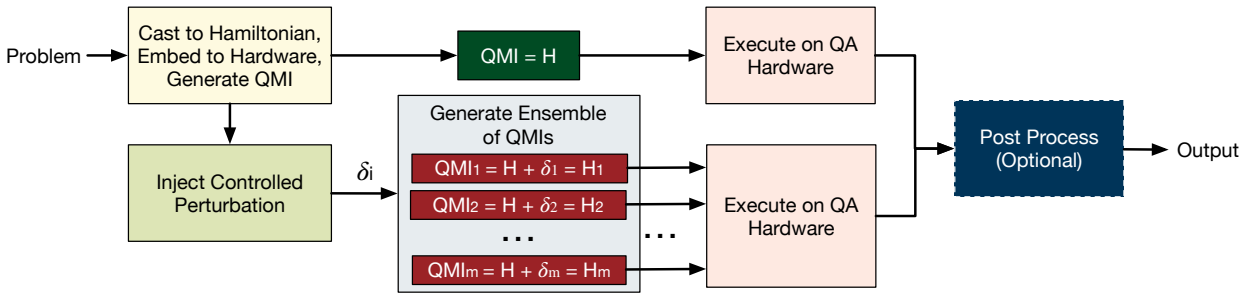

Figure 2. Overview of EQUAL. EQUAL creates an ensemble of QMIs by adding controlled perturbations to the original QMI. It executes the original QMI as well as the ensemble of QMIs separately on the QA hardware and returns the outcome with the lowest energy value. EQUAL can also optionally leverage existing postprocessing error mitigation schemes (EQUAL+).

2.1.3 Generating Ensembles via Controlled Perturbations

Creating an effective perturbation Hamiltonian is nontrivial. If the perturbations add too little noise, the resulting Hamiltonian will be too close to the problem Hamiltonian and encounter similar bias. Alternately, too large perturbations result in a Hamiltonian significantly different from the problem of interest and can produce infeasible results. Thus, there is a trade-off between the effectiveness of a perturbation Hamiltonian to reduce bias and its ability to alter the problem Hamiltonian. To address this challenge and generate an effective ensemble of QMIs, EQUAL exploits the device-level characteristics of QAs.

Casting a problem to a Hamiltonian can require a double-precision representation of the Hamiltonian coefficients. Unfortunately, real QAs can only support a small range and precision of coefficients due to the limitations imposed by the digital to analog converters (DACs) used on QAs. If the precision of the coefficients is too large, the DACs are too slow, which eventually slows the controlling modules of QAs and is not desirable. To bridge this gap, post the casting step, coefficients of the QMI are truncated to match the precision supported by the hardware. While this is a limitation on QAs, EQUAL leverages it to its advantage and draws the coefficients of the perturbation Hamiltonian randomly at a range that is below the supported precision so that adding the perturbation Hamiltonian only shifts the coefficients of the QMI (post truncation) to one of the neighboring quantization levels and thus, does not significantly alter the problem landscape. More specifically, let b be the number of bits used for representing coefficients of a physical QA. For every ensemble, EQUAL draws a uniform random number as

$$r \in \left[\frac{1}{2^{b+1}}, \frac{1}{2^{b}}\right]$$

and set all coefficients of the Perturbation Hamiltonian to be r.

Unfortunately, the precision of the coefficients supported on real devices is unavailable to programmers. Determining this precision is vital for the performance of EQUAL. Drawing the Perturbation Hamiltonian coefficients far below the supported precision introduces large noise and may alter the Hamiltonian landscape significantly. Alternately, drawing them far above the supported range may not affect post truncation. To address this challenge, EQUAL profiles the QA using random benchmarks to estimate the precision supported by QAs. In this experiment, we truncate all coefficients of the benchmark for $2, 3, \ldots, 16$ bits precision and execute the corresponding QMIs. Figure 3 shows the relative Energy Residual of the truncated QMIs with respect to the original problem (without truncation). Our profiling experiments with multiple benchmarks show that the hardware is likely limited by 7–8 bits of precision. Thus, EQUAL generates coefficients of ensembles as

$$r \in \left[\frac{1}{2^{9}}, \frac{1}{2^{8}}\right].$$

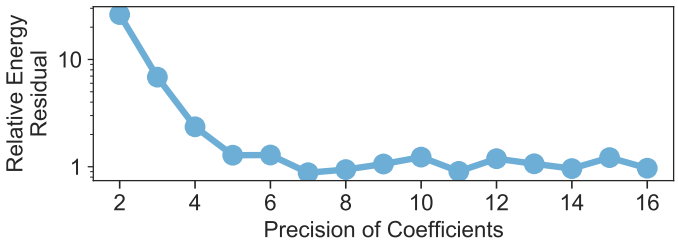

Figure 3. Relative Energy Residual of QMIs with truncated coefficients with respect to the original problem QMI for bits values of precision. (Lower is better)

2.1.4 Execution on QA Hardware

EQUAL splits the trials between the ensemble of QMIs equally, including the original QMI (without perturbation), and executes them separately on the QA hardware. Our default design uses 10 ensembles of QMIs, and allocates 10,000 trials for every ensemble. We do a more rigorous sensitivity analysis for the number of trials and ensembles in Section 2.1.7.

By default, the outcome with the lowest energy is deemed as the solution for problems executed on QAs. In the baseline, this corresponds to the outcome with the lowest energy obtained by executing the original QMI. As EQUAL executes multiple QMIs, the outcome with the lowest energy among all the QMIs is returned as the solution. Also, as EQUAL runs the ensemble of perturbed QMIs in addition to the original program QMI, the final solution is guaranteed not to perform worse than the baseline, assuming there are no sampling errors. Note that the solution with the minimum energy corresponds to an outcome that may come from a single QMI. For the baseline, this corresponds to the original QMI, whereas for EQUAL, it comes from one or more of the QMIs in the ensemble. However, which QMI corresponds to the best solution is unknown a priori, and EQUAL must execute the entire ensemble.

2.1.5 Results for Energy Residual

Energy Residual (ER) computes the gap between the energy obtained from the best outcome on a QA with the global optima. Figure 4 compares the ER of the individual benchmarks for baseline and EQUAL. We observe that the ER quickly saturates in the baseline for all benchmarks, whereas it improves for EQUAL as more QMIs are executed. As the QMIs are generated using random controlled perturbations, some of them may result in higher ER compared to the baseline due to a different noise profile at run time. However, the ensemble overall enables EQUAL to reach a better solution. In the worst case, EQUAL performs similar to baseline as the original program QMI is executed. We observe that the baseline fidelity saturates with more trials, whereas the diversity of EQUAL helps it keep on improving with additional trials.

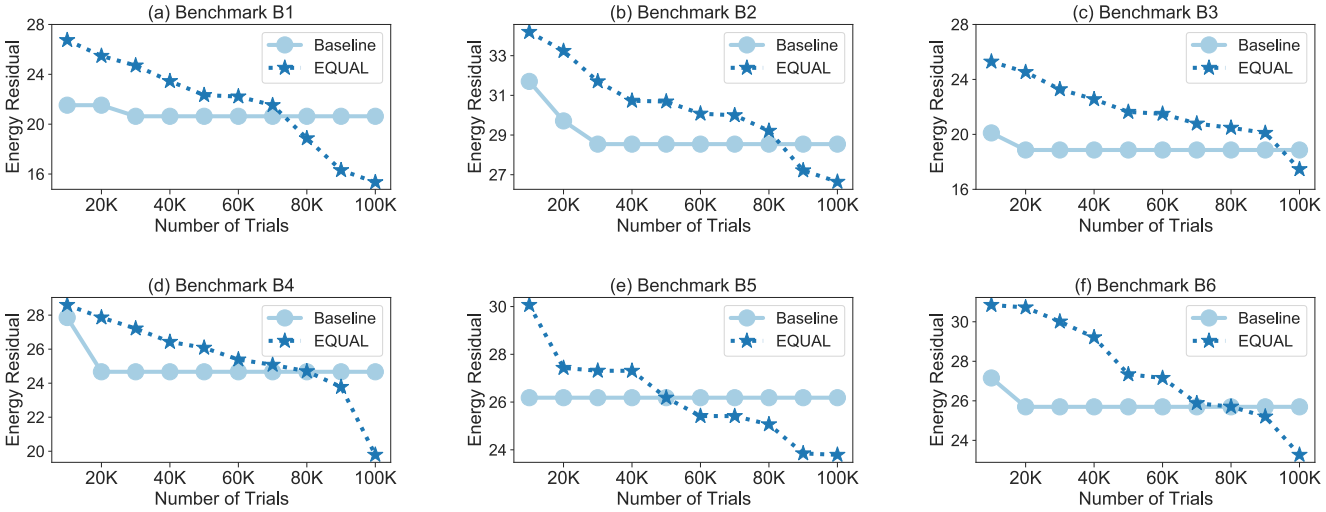

Figure 4. Trends in Energy Residual for the baseline and EQUAL for six random benchmark problems.

Figure 5 shows the ER of EQUAL for our benchmarks executed on the D-Wave 2000Q machine relative to the baseline. We observe that EQUAL bridges the difference between the baseline and the ideal by an average of 14% (and up to 26%).

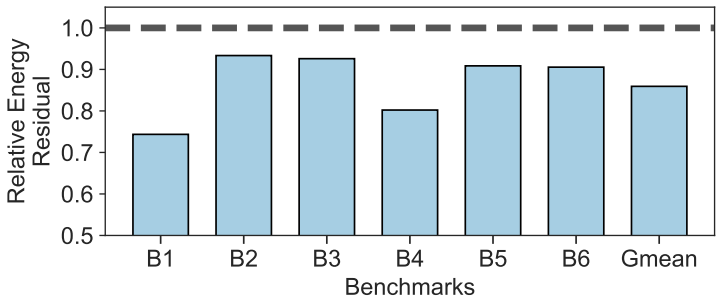

Figure 5. Energy Residual of six random benchmarks on D-Wave QA hardware using EQUAL relative to the baseline.

2.1.6 Results for Validation of Precision Selection

We draw the perturbation Hamiltonian coefficients in the range $\left[\frac{1}{2^9}, \frac{1}{2^8}\right]$ based on profiling across a wide range of values. We confirm that this approach is robust by conducting additional studies at the application level. Figure 6 shows the ER of our benchmarks relative to baseline when the precision range is varied. Here we use 10k trials/shots (1k trials per every ensemble QMI) for every bit-precision setting; however, Figure 5 illustrates the results of 100k trials (10k trials per every ensemble QMI). We confirm that 8 to 9 bits of precision is more robust compared to others and, on average, outperforms the others.

2.1.7 Impact of Number of Ensembles

We study the impact of the number of ensembles on the effectiveness of EQUAL using a single benchmark problem. For a given trial budget of 100K trials, we choose two modes for EQUAL. In the first instance, we use 10 QMIs and run each of them for 10K trials each. In the second instance, we use 100 QMIs and run each of them for 1K trials each. Figure 7 shows the

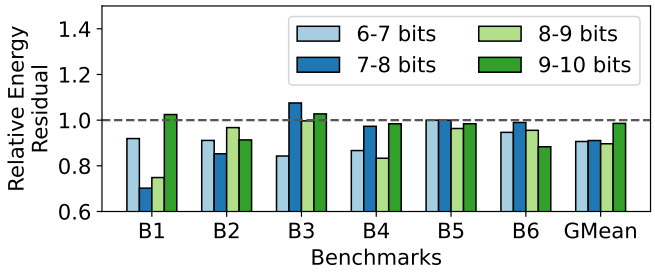

Figure 6. Energy Residual of random benchmarks on D-Wave QA hardware using EQUAL relative to the baseline for different values of bit precision used for generating QMIs.

ER for the baseline and these two instances of EQUAL. Note that we access the QA device through cloud services, and more rigorous sensitivity analysis in terms of QMIs and trials is challenging. We observe that executing more QMIs introduces more randomness and makes them vulnerable to sampling errors. EQUAL with 10 QMIs achieve a sweet spot between the baseline and EQUAL with a large number of ensembles such that we have both diversity as well as sufficient trials for each QMI to reduce sampling errors.

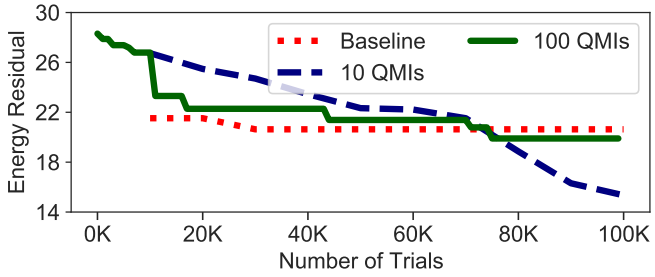

Figure 7. Energy Residual for benchmark (B1). The baseline executes a single QMI for all 100K trials. EQUAL has 10 QMIs for 10K trials each or 100 QMIs executed for 1K trials each.

2.1.8 Case-Study: How EQUAL Reduces Systematic Bias

Figure 8 shows the histograms of the energy values obtained by running benchmark B1 for the baseline and EQUAL. The goal of QAs is to obtain the outcome corresponding to the ground state energy. We observe that the optimal solution is at a distance d_1 from the ground state, and EQUAL produces a solution at a distance d_2 that is closer to the ground state energy ($d_2 < d_1$) by minimizing the impact of bias. We also observe that distributions for both the baseline and EQUAL overlap largely, indicating that the ensemble of QMIs do not largely alter the original Hamiltonian corresponding to our problem. We make similar observations for other benchmarks.

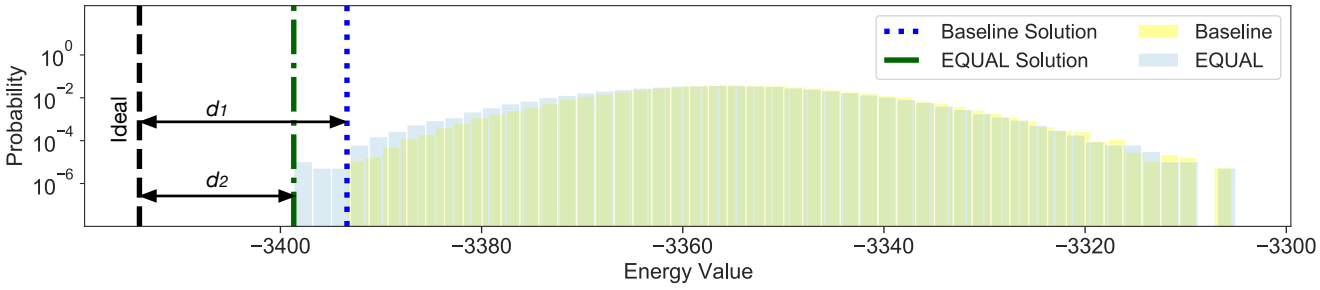

Figure 8. Histogram of energy values from the outcomes on the QA for benchmark B1 using the Baseline and EQUAL. The solution from EQUAL is closer to the ideal solution compared to the baseline solution ($d_2 < d_1$). The histograms for the baseline and EQUAL largely overlaps which indicates that EQUAL does not significantly alter the problem Hamiltonian.

An ideal QA is a machine that samples from the Boltzmann distribution, whereas the distribution obtained from a real QA hardware is different due to noise. The best solution obtained by a QA depends on the overlapping region between the ideal and the noisy distributions. From Figure 8, we observe two potential approaches to get closer to the global optima. First, by *flattening* the energy histogram of the Hamiltonian such that it covers a broader search space. Second, by *shifting* the energy histogram towards the ideal solution. Note that both of these techniques must ensure that the properties of the original program Hamiltonian remain unaltered. EQUAL uses the first approach. The performance of EQUAL can be improved further if we could shift the histogram closer to the ideal solution.

2.2 Combining EQUAL with Error-Mitigation Schemes

We explore combining EQUAL with existing error mitigation schemes to obtain the advantage of both flattening the histogram and shifting the histogram towards the ground state. Ensembles are generated by only adding controlled perturbations to the problem Hamiltonian. Therefore, they have limited capability to shift the noisy distribution from a QA towards the ideal distribution even if a large number of ensembles are used. Alternately, large perturbations may significantly change the landscape of the problem. Instead, we take an orthogonal approach and explore existing error-mitigation schemes that can introduce a shift in the energy histogram.

2.2.1 Primer on Error-Mitigation Schemes for QA

Error-mitigation schemes for QAs can be classified into (1) software and (2) hardware schemes. Software schemes refer to optimizations performed during the casting and embedding steps (preprocessing techniques) or modifications on the outcomes obtained from QAs (postprocessing techniques). On the other hand, hardware-based schemes control the device-level parameters on QAs to reduce the impact of errors.

We characterize the impact of these error-mitigation techniques individually and combine them with each other to understand their effectiveness in (1) eliminating the systematic bias on their own and (2) shifting the noisy distribution of the QA towards the ideal distribution. For our analysis, we choose (a) spin reversal transform, (b) longer inter-sample delay, and (c) single-qubit correction. Spin-reversal transform is a representative candidate for a software preprocessing technique. On the other hand, the inter-sample delay is a device-level control available to programmers to reduce the correlation between consecutive trials on a QA. Lastly, Single-Qubit Correction (SQC)[27] is a postprocessing technique that leverages the insight that for a given QMI, QAs can quickly recognize the neighborhood of the ground state even if they fail to get to the ground state[27]. We performed characterization studies for these three error mitigation schemes (see supplementary information) and found that SQC is the most effective scheme, and therefore we use SQC as the error mitigation scheme for our study.

2.2.2 Overview of SQC Post-Processing

SQC is analogous to the gradient descent scheme but only applicable to discrete optimization problems. Instead of computing the gradient for determining the direction of the move in every iteration, SQC uses a greedy approach and moves to a neighbor (i.e., an outcome that is one Hamming distance away from the current solution) with the lowest energy value. Figure 9 illustrates the overview of an iteration of SQC. For each candidate outcome, SQC finds the one Hamming distance away from neighbors and computes their energy values. If any neighbors can obtain a lower energy value than the candidate itself, the neighbor is retained, and the current solution is discarded. When multiple neighbors obtain lower energy values, the best neighbor is retained. The process is repeatedly executed until we cannot find any new neighbor that has better quality.

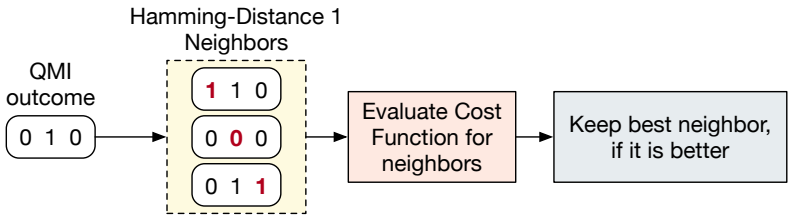

Figure 9. Single Qubit Correction Post-Processing[27]

2.2.3 EQUAL+: Combining EQUAL and SQC

Figure 10 shows an overview of EQUAL+. EQUAL+ applies SQC on the outcomes of each QMI and obtains the best outcome for each QMI. The process is performed for each QMI in parallel. Once applying SQC on each QMI converges, the final output of EQUAL+ is picked as the candidate with the lowest energy among all the individual best candidates from the QMIs. The time to converge depends on several factors such as the size of the problem, number of outcomes, quality of the outcomes.

However, our evaluations show that EQUAL+ converges within a few seconds, even for large benchmarks such as the ones used in our evaluations.

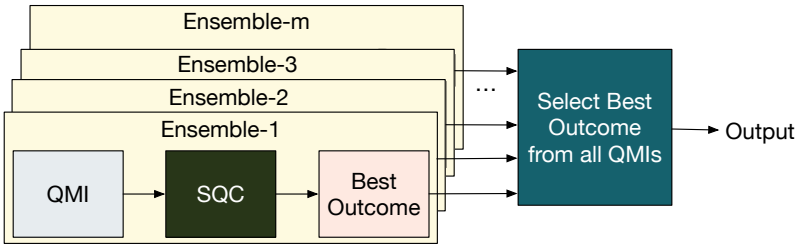

Figure 10. Overview of EQUAL+ design. It applies the SQC postprocessing algorithm on the outcomes from each QMI in parallel. Finally, it selects the best outcome from all the QMIs as the output solution.

Using this greedy approach helps locate neighbors from current outcomes that were originally not produced by the QA. With each neighbor located, EQUAL+ shifts the outcome distribution towards the ideal solution (global optima). Note that although SQC is effective on its own, the diversity of EQUAL+ is essential to improve its search space. The capability of SQC alone to introduce new outcomes is limited by the quality of outcomes from the QMI. In EQUAL+, the ensembles enable us to explore a much larger neighborhood compared to applying SQC alone. In the end, EQUAL+ may discover a solution from one of the weakest outcomes corresponding to one of the weakest QMIs (a sub-optimal outcome that did not correspond to the best solution in any of the QMIs). Note that EQUAL+ is versatile, and any other postprocessing candidate that introduces the desired shifting property in the energy distribution may be used. We use SQC for its performance and low time complexity.

2.2.4 Results for Energy Residual

Figure 11 shows the Energy Residual of EQUAL+ relative to the baseline. We also compare against EQUAL and SQC standalone. We observe that EQUAL+ improves the ER by 0.45 compared to the baseline on average and by up to 0.32. In other words, EQUAL+ improves the quality of solutions by 55% on average and up to 68%.

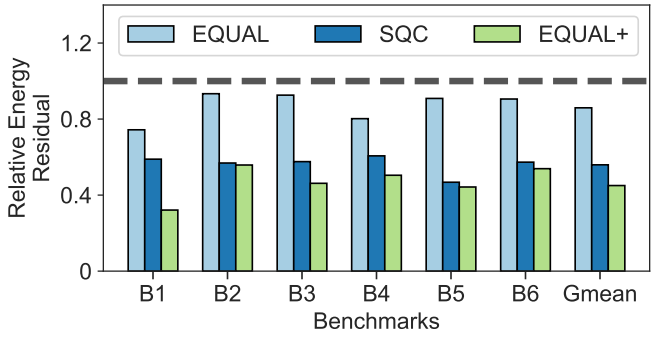

Figure 11. Energy Residual using EQUAL+ relative to the baseline. We also compare with EQUAL and SQC standalone.

2.2.5 Case-Study: How EQUAL+ reduces Systematic Bias

Figure 12 shows the histograms of energy values of benchmark B1 for EQUAL and EQUAL+. We observe that the optimal solution is at a distance d_1 from the ground state energy in EQUAL. EQUAL+ exploits the shifting property of SQC to obtain a solution at distance d_2 and is closer to the ground state energy ($d_2 < d_1$). EQUAL+ shifts the overall histogram towards the ideal solution and achieves the intended goal. As EQUAL+ applies postprocessing on the outcomes from the QMIs, the introduced shift in the histogram does not alter the original problem Hamiltonian.

2.3 Overhead Analysis

We discuss the overheads for both EQUAL and EQUAL+. EQUAL generates the ensemble of QMIs prior to execution on the QA. As the perturbed Hamiltonian only adjusts the coefficients of the original problem QMI, the ensemble does not need to re-perform the casting or embedding step. Although embedding can take up to several hours and may fail for certain Hamiltonians, this overhead and limitation are entirely avoided by EQUAL. EQUAL also requires the programmer to estimate the precision of the hardware using a set of profiling experiments. However, profiling need not be done for each application. As

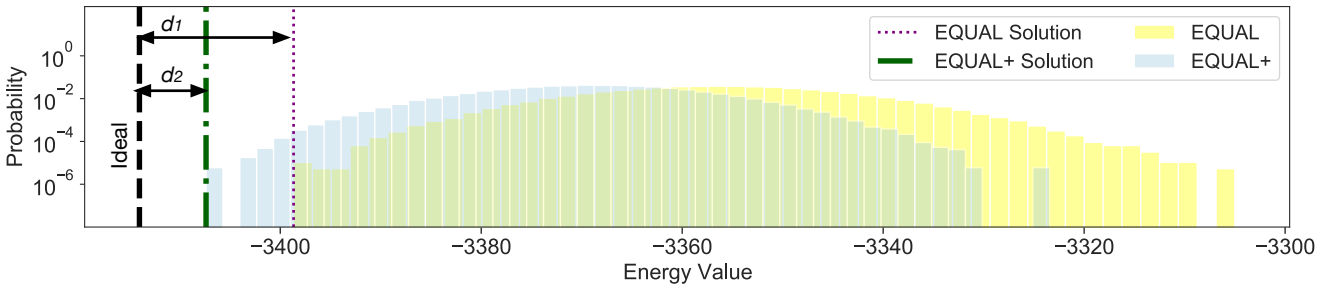

Figure 12. Histogram of energy values from the outcomes on the QA for benchmark B1 using EQUAL and EQUAL+.

the precision supported is only device-specific, profiling once for each QA hardware is enough, and the same information can be re-used for multiple applications. For execution on the QA, EQUAL requires the same number of trials as the baseline and, therefore, does not incur any overhead of additional trials.

EQUAL+ incurs some additional overheads for the postprocessing step as it applies the SQC heuristic algorithm to all the outcomes obtained from all the QMIs. The space complexity of the postprocessing phase in EQUAL+ is linear with the number of qubits[27]. As SQC is iteratively applied on every outcome of a QMI, the time complexity depends on the number of outcomes which is equal to the number of trials in the worst-case (assuming each trial generates a unique outcome). The postprocessing for each QMI is done in parallel. Our studies show that EQUAL+ converges within a few iterations and the postprocessing step for EQUAL+ only takes a few seconds. Therefore, the overheads are acceptable.

3 Discussion

Quantum Annealers (QAs) are single-instruction quantum machines that can only sample from the ground state of a given problem Hamiltonian. Unlike gate model quantum computers that are limited to a few dozen qubits, QAs have thousands of qubits and are promising for a wide range of real-world applications. QAs deal with industry-scale optimization problems where even a minuscule improvement has a tremendous impact in terms of practical advantage such as saving millions of dollars[24,46,47] in the context of scheduling and planning applications or finding better candidates for drug discovery[26] and material science[25]. In theory, we can expect QAs to find the global optima of a given problem with a high probability[6,41]. In reality, however, QAs often fail to find the global optima for large problems due to various hardware drawbacks such as noise, device errors, limited programmability, and a confined annealing process. Addressing these drawbacks requires device-level enhancements that may span generations of QAs. Hence, there is an increasing interest in developing software policies to subside these limitations.

We can view QAs as a (quantum) machine that samples from a Boltzmann distribution where samples/solutions with lower energy/objective values are exponentially more likely to be observed. However, energy values of drawn samples by a physical QA follows the Gaussian distribution, and there is a systematic bias associated with each QMI that shifts the mean of this Gaussian distribution away from the ground state of the corresponding problem Hamiltonian, as shown in Figure 1(a). The probability of finding the ground state of the given problem Hamiltonian mainly depends on the distance between means of these two (expected Boltzmann and observed Gaussian) distributions, as well as the standard deviation of the energy values attained by the QA. Indeed, increasing the number of samples does not alter the mean or standard deviation of the underlying Gaussian distribution, so the fidelity saturates when the number of samples was enough to represent the corresponding energy distribution, as shown in Figure 1(b). In the same manner, employing current error mitigation schemes—e.g., preprocessing schemes like spin-reversal-transforms, hardware-level techniques such as increased inter-sample delay, and local optimization heuristics as postprocessing policies—cannot eliminate this systematic bias. For example, drawing samples by a physical QA results in excited states of the corresponding Hamiltonian that are not necessarily local optimums. Hence, applying local optimization heuristics to drawn samples by a physical QA generally improves the quality of solutions. However, the performance of such techniques mainly depends on the quality of drawn samples that is susceptible to systematic bias.

This paper proposes EQUAL—Ensemble Quantum Annealing—a software framework that creates multiple perturbed copies of an input problem by injecting controlled perturbations to the original problem Hamiltonian. By executing an ensemble of quantum machine instructions (QMIs), EQUAL projects the program to different noise profiles and therefore, different biases. To generate ensembles, EQUAL creates new Hamiltonians, called *Perturbation Hamiltonians*, and adds them to the original problem Hamiltonian. Every perturbation Hamiltonian adds noise to the original Hamiltonian, and the QMI obtained from this process is a perturbed variation of the original QMI. The challenge in this step is that adding extremely small perturbations will have no impact on the systematic bias, whereas adding large perturbations can significantly change the landscape of the

original Hamiltonian. In the worst case, the final perturbed Hamiltonian may correspond to a problem completely different from the one at hand. Thus, there exists a trade-off between the ability to eliminate systematic bias and the correctness of a Hamiltonian. Ideally, we want a perturbed Hamiltonian that can eliminate systematic bias without altering the characteristics of the problem Hamiltonian significantly. To address this challenge, EQUAL exploits the fact that QAs only allow a limited precision of coefficients for a Hamiltonian due to hardware limitations. For every ensemble, EQUAL draws the coefficients of the corresponding perturbation Hamiltonian randomly at a range just below the supported precision so that adding the Perturbation Hamiltonian may only shift the coefficients of the QMI (post truncation) to one of the neighboring quantization levels and not impose significant changes to the landscape of the original Hamiltonian.

We can view a QMI as a random variable that follows a Gaussian distribution—energy values of samples attained by physical QAs follow a Gaussian distribution, as shown in Figure 1(a). Applying controlled perturbations results in an ensemble of perturbed Hamiltonians with (slightly) different mean and standard deviation values. Aggregating samples attained by these perturbed Hamiltonians forms random variables that follow a Gaussian distribution. Since we generate perturbations independently, we can expect the mean of aggregated energy values to approach the mean of these Gaussian distributions. However, as shown in Figure 8, the resulting distribution has a notably higher standard deviation. Moreover, glassy landscapes may include wide energy barriers that can mitigate the odds of quantum tunneling. Adding (controlled) perturbations to the landscape of the problem Hamiltonian can decrease or increase the width of some of the energy barriers. While it is impossible to predict the impact of applying every perturbation individually, we can expect the ensemble of such perturbed Hamiltonians to provide a higher odds of quantum tunneling. Our evaluations using a D-Wave 2000Q quantum machine show that EQUAL bridges the difference between the baseline and the ideal by an average of 14% (and up to 26%), without requiring any additional trials.

We also analyze existing error-mitigation approaches for QAs. Our characterization experiments on D-Wave shows that the SQC[27] postprocessing technique is highly effective for D-Wave QAs. We compare EQUAL with SQC and show that the two schemes can be combined for even more significant benefits. The resulting design, *EQUAL+*, provides significantly better fidelity than EQUAL and SQC standalone. As the SQC postprocessing relies only on classical computations, EQUAL+ does not incur any additional trials compared to EQUAL.

Unlike classical optimization heuristics (namely the simulated annealing) that are converged to local optimums, samples attained by QAs are not guaranteed to be local/global optimum. Furthermore, due to control errors and diabatic transitions, samples attained by physical QAs are not guaranteed to be an eigenstate of the problem Hamiltonian. Thus, applying local optimization techniques to raw samples attained by QAs is expected to result in new (synthetic) samples with lower energy values. The performance of classical local optimization heuristics significantly depends on the quality of initial states. Problems with glassy landscapes include many excited states whose energy levels are very close to the minimum eigenvalue of the corresponding Hamiltonian. Hence, physical QAs are very likely to observe one of these excited states that surround the ground state of the corresponding Hamiltonian. On this basis, EQUAL runs multiple perturbed Hamiltonians to explore the (glassy) landscapes better and leverages SQC to exploit the neighborhood of attained excited states better. Our evaluations using a D-Wave 2000Q quantum machine show that EQUAL+ bridges the difference between the baseline and the ideal by an average of 55% (and up to 68%).

4 Method

We discuss the evaluation infrastructure used in this paper.

4.1 Quantum Platform and Baseline

For our evaluations, we use the 2041-qubit quantum annealer from D-Wave Systems via Amazon BraKet cloud service[10]. We use the default annealing time (i.e., 20 μseconds) and schedule recommended for this system. For the baseline, we use 100,000 trials for each benchmark. Such a large number of trials reduces sampling errors, and therefore, this serves as a strong baseline. For EQUAL, trials are equally split between QMIs. Thus, EQUAL requires the same number of trials as the baseline.

4.2 Benchmarks

We use *random weighted Max-Cut* problems, similar to Quantum Approximate Optimization Algorithms[48] used on gate-based quantum computers. For the benchmarks, we draw the Hamiltonian coefficients of the QMIs from the standard normal distribution (a mean of 0 and a standard deviation of 1). This approach is a common practice used in prior works related to benchmarking QAs[7,27,31,49,50]. To avoid the impact of embedding on our evaluations, we directly use the connectivity graph of the D-Wave QA. Thus, the number of program qubits in benchmarks is equal to the number of physical qubits on the QA. As the size of benchmarks significantly exceeds the size of existing gate-model quantum computers, we cannot compare our results with them.

Random problems on the Chimera graph have been shown not to have finite temperature spin glass transitions[51]. Therefore, random problems on planar graphs (i.e., Chimera graph or other quantum hardware coupling graphs) are likely not to present hard problems for Monte Carlo type algorithms. Recent studies have shown that the D-Wave QAs generally fail to find the ground state of most randomly generated Ising problems[27]. To compare the performance of the quantum optimization techniques with (best known) classical optimization approaches, we may generate instances of hard Ising models where the global optimum is planted in the problem Hamiltonian[52]. However, our objective in this study is not to demonstrate the supremacy of quantum machines in addressing hard optimization problems. Indeed, our objective is to devise a lightweight and effective scheme for improving the performance of physical QAs. Moreover, to the best of our knowledge, all proposals for generating random hard optimization problems for QAs generate Ising Hamiltonians with integer coefficients[52]. However, physical QAs have limited precision (i.e., about 7–8 bits precision on the D-Wave QAs), and benchmarks with integer coefficients cannot capture this limitation (i.e., the impact of truncating coefficients prior to the annealing)[27].

4.3 Figure-of-Merit

We evaluate the reliability of QA using **Energy Residual (ER)**. The best solution from a QA is the outcome with the minimum energy. ER computes the energy gap between the minimum energy (E_{min}) obtained on a QA with respect to the global energy minimum (E_{global}) of the application as follows:

$$\text{Energy Residual (ER)} = \left| E_{min} - E_{global} \right|. \tag{2}$$

Ideally, when the best solution obtained on a QA corresponds to the ground state of the problem Hamiltonian, ER is zero. Thus, a lower value (closer to zero) for ER is desirable.

The challenge in computing the ER for random large benchmarks spanning 2000+ qubits is that finding the ground state of the Hamiltonian is nontrivial. To overcome this challenge and still enable a fair comparison, we perform intensive classical computations using state-of-the-art tools[53] and approximate the global optimum of our benchmark problems. Recent studies have shown that this algorithm can estimate the ground state of Chimera based Hamiltonians[9,28] (such as the ones considered in our paper) with a very high probability.

Time-to-Solution (TTS) is a standard metric for quantifying the performance of quantum optimization techniques[54]. To apply TTS: (a) we need to know the optimum solution of all benchmarks; and (b) we need to know the required number of samples/trials to observe the ground state of the corresponding Hamiltonian. Finding the ground state of NP-hard problems is infeasible, and current proposals for generating hard benchmark problems with planted solutions are limited to Ising models with integer coefficients. In addition, estimating the probability of success (i.e., required number of samples/trials to visit the ground state of the problem Hamiltonian) is nontrivial in the current model of accessing physical QAs. Furthermore, the Hamiltonian that real QA samples from its ground state can be different from the problem Hamiltonian of interest. Consequently, the ground state of the problem Hamiltonian might be out of reach for physical QAs.

Acknowledgements

We would like to thank Suhas Karthik Vittal, Jiahao Wen, Narges Alavisamani, and Sanjay Kariyappa for constructive criticisms of the manuscript. Ramin Ayanzadeh was supported by the NSF Computing Innovation Fellows (CI-Fellows) program. Poulami Das was supported by the Microsoft Research PhD fellowship. This research was partially supported by the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin–Madison with funding from the Wisconsin Alumni Research Foundation.

Author contributions statement

R.A. conceived of the presented model and carried out the implementations. R.A. and P.D. planned the experiments and performed the analysis. S.T. provided insights into doing experiments and defining the figure of merit. M.Q. supervised the project and discussions. All authors discussed the results and contributed to the final manuscript.

Additional information

Authors declare no competing interests.

Data availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

1. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review* **41**, 303–332 (1999).
2. Grover, L. K. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 212–219 (1996).
3. Feynman, R. Simulating physics with computers. *Int. J. Theor. Phys.* **21** (1982).
4. Lloyd, S. Universal quantum simulators. *Science* 1073–1078 (1996).
5. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge University Press, 2010).
6. Albash, T. & Lidar, D. A. Adiabatic quantum computation. *Rev. Mod. Phys.* **90**, 015002 (2018).
7. Das, A. & Chakrabarti, B. K. Colloquium: Quantum annealing and analog quantum computation. *Rev. Mod. Phys.* **80**, 1061 (2008).
8. McGeoch, C. C. Theory versus practice in annealing-based quantum computing. *Theor. Comput. Sci.* (2020).
9. Inc., D.-W. S. The first and only quantum computer built for business. https://www.dwavesys.com/ (2022). [Online; accessed 22-July-2021].
10. Amazon. Amazon Braket - Explore and experiment with quantum computing:. https://aws.amazon.com/braket/ (2022). [Online; accessed 22-July-2021].
11. McGeoch, C. & Farre, P. The d-wave advantage system: An overview. Tech. Rep., Tech. Rep. (D-Wave Systems Inc, Burnaby, BC, Canada, 2020).
12. Rieffel, E. G. *et al.* A case study in programming a quantum annealer for hard operational planning problems. *Quantum Inf. Process.* **14**, 1–36 (2015).
13. Venturelli, D., Marchand, D. J. & Rojo, G. Quantum annealing implementation of job-shop scheduling. *arXiv preprint arXiv:1506.08479* (2015).
14. Tran, T. T. *et al.* A hybrid quantum-classical approach to solving scheduling problems. In *Ninth annual symposium on combinatorial search* (2016).
15. Bian, Z. *et al.* Mapping constrained optimization problems to quantum annealing with application to fault diagnosis. *Front. ICT* **3**, 14 (2016).
16. Su, J., Tu, T. & He, L. A quantum annealing approach for boolean satisfiability problem. In *Proceedings of the 53rd Annual Design Automation Conference*, 148 (ACM, 2016).
17. Ayanzadeh, R., Halem, M. & Finin, T. Reinforcement quantum annealing: A hybrid quantum learning automata. *Sci. Reports* **10**, 1–11 (2020).
18. O'Malley, D., Vesselinov, V. V., Alexandrov, B. S. & Alexandrov, L. B. Nonnegative/binary matrix factorization with a d-wave quantum annealer. *PloS one* **13**, e0206653 (2018).
19. Peng, W. *et al.* Factoring larger integers with fewer qubits via quantum annealing with optimized parameters. *SCIENCE CHINA Physics, Mech. & Astron.* **62**, 60311 (2019).
20. Hu, F. *et al.* Quantum computing cryptography: Finding cryptographic boolean functions with quantum annealing by a 2000 qubit d-wave quantum computer. *Phys. Lett. A* **384**, 126214 (2020).
21. Ayanzadeh, R., Mousavi, S., Halem, M. & Finin, T. Quantum annealing based binary compressive sensing with matrix uncertainty. *arXiv preprint arXiv:1901.00088* (2019).
22. Ayanzadeh, R., Halem, M. & Finin, T. An ensemble approach for compressive sensing with quantum annealers. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, 3517–3520 (IEEE, 2020).
23. Inoue, D., Okada, A., Matsumori, T., Aihara, K. & Yoshida, H. Traffic signal optimization on a square lattice with quantum annealing. *Sci. reports* **11**, 1–12 (2021).
24. Elsokkary, N., Khan, F. S., La Torre, D., Humble, T. S. & Gottlieb, J. Financial portfolio management using d-wave quantum optimizer: The case of abu dhabi securities exchange. Tech. Rep., Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States) (2017).
25. Kitai, K. *et al.* Designing metamaterials with quantum annealing and factorization machines. *Phys. Rev. Res.* **2**, 013319 (2020).

26. Mulligan, V. K. *et al.* Designing peptides on a quantum computer. *bioRxiv* 752485 (2020).

27. Ayanzadeh, R., Dorband, J., Halem, M. & Finin, T. Multi-qubit correction for quantum annealers. *Sci. Reports* **11** (2021).

28. Cai, J., Macready, W. G. & Roy, A. A practical heuristic for finding graph minors. *arXiv preprint arXiv:1406.2741* (2014).

29. Date, P., Patton, R., Schuman, C. & Potok, T. Efficiently embedding qubo problems on adiabatic quantum computers. *Quantum Inf. Process.* **18**, 1–31 (2019).

30. Golden, J. K. & O'Malley, D. Pre-and post-processing in quantum-computational hydrologic inverse analysis. *Quantum Inf. Process.* **20**, 1–18 (2021).

31. Borle, A. & McCarter, J. On post-processing the results of quantum optimizers. In *International Conference on Theory and Practice of Natural Computing*, 222–233 (Springer, 2019).

32. Goodrich, T. D., Sullivan, B. D. & Humble, T. S. Optimizing adiabatic quantum program compilation using a graph-theoretic framework. *Quantum Inf. Process.* **17**, 1–26 (2018).

33. Okada, S., Ohzeki, M., Terabe, M. & Taguchi, S. Improving solutions by embedding larger subproblems in a d-wave quantum annealer. *Sci. reports* **9**, 1–10 (2019).

34. Tannu, S. S. & Qureshi, M. Ensemble of diverse mappings: Improving reliability of quantum computers by orchestrating dissimilar mistakes. In *Proc. of MICRO*, 253–265 (2019).

35. Patel, T. & Tiwari, D. Veritas: accurately estimating the correct output on noisy intermediate-scale quantum computers. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 1–16 (IEEE, 2020).

36. Mohseni, N. *et al.* Error suppression in adiabatic quantum computing with qubit ensembles. *npj Quantum Inf.* **7**, 1–10 (2021).

37. Amara, P., Hsu, D. & Straub, J. E. Global energy minimum searches using an approximate solution of the imaginary time schrödinger equation. *The J. Phys. Chem.* **97**, 6715–6721 (1993).

38. Finnila, A., Gomez, M., Sebenik, C., Stenson, C. & Doll, J. Quantum annealing: a new method for minimizing multidimensional functions. *Chem. physics letters* **219**, 343–348 (1994).

39. Kadowaki, T. & Nishimori, H. Quantum annealing in the transverse ising model. *Phys. Rev. E* **58**, 5355 (1998).

40. Ohzeki, M. & Nishimori, H. Quantum annealing: An introduction and new developments. *J. Comput. Theor. Nanosci.* **8**, 963–971 (2011).

41. Nishimori, H. & Takada, K. Exponential enhancement of the efficiency of quantum annealing by non-stoquastic hamiltonians. *Front. ICT* **4**, 2 (2017).

42. Vinci, W., Albash, T. & Lidar, D. A. Nested quantum annealing correction. *npj Quantum Inf.* **2**, 1–6 (2016).

43. Karimi, H., Rosenberg, G. & Katzgraber, H. G. Effective optimization using sample persistence: A case study on quantum annealers and various monte carlo optimization methods. *Phys. Rev. E* **96**, 043312 (2017).

44. Karimi, H. & Rosenberg, G. Boosting quantum annealer performance via sample persistence. *Quantum Inf. Process.* **16**, 166 (2017).

45. Boothby, T., King, A. D. & Roy, A. Fast clique minor generation in chimera qubit connectivity graphs. *Quantum Inf. Process.* **15**, 495–508 (2016).

46. Ahuja, R. K., Cunha, C. B. & Şahin, G. Network models in railroad planning and scheduling. In *Emerging theory, methods, and applications*, 54–101 (INFORMS, 2005).

47. Carlson, B. *et al.* Miso unlocks billions in savings through the application of operations research for energy and ancillary services markets. *Interfaces* **42**, 58–73 (2012).

48. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. *arXiv preprint:1411.4028* (2014).

49. Pudenz, K. L., Albash, T. & Lidar, D. A. Quantum annealing correction for random ising problems. *Phys. Rev. A* **91**, 042302 (2015).

50. Ayanzadeh, R., Halem, M., Dorband, J. & Finin, T. Quantum-assisted greedy algorithms. *arXiv:1912.02362* (2019).

51. Katzgraber, H. G., Hamze, F. & Andrist, R. S. Glassy chimeras could be blind to quantum speedup: Designing better benchmarks for quantum annealing machines. *Phys. Rev. X* **4**, 021008 (2014).

52. Marshall, J., Martin-Mayor, V. & Hen, I. Practical engineering of hard spin-glass instances. *Phys. Rev. A* **94** (2016).

53. Ayanzadeh, R., Dorband, J., Halem, M. & Finin, T. Multi qubit correction (mqc) for quantum annealers, DOI: 10.5281/zenodo.5142230 (2021). Python implementation of MQC.

54. Rønnow, T. F. *et al.* Defining and detecting quantum speedup. *science* **345**, 420–424 (2014).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- EQUALsupinfo.pdf