

This item is likely protected under Title 17 of the U.S. Copyright Law. Unless on a Creative Commons license, for uses protected by Copyright Law, contact the copyright holder or the author.

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

ON QUANTUM ANNEALING WITHOUT A PHYSICAL QUANTUM ANNEALER

Ameya Bhawe^{1*}, Ajinkya Borle¹

¹CSEE Department, University of Maryland Baltimore County, Baltimore, USA

*E-mail: ameyab1@umbc.edu

Keywords: QUANTUM ANNEALING, QUANTUM COMPUTING, SIMULATED ANNEALING, COMBINATORIAL OPTIMISATION, ISING MODEL

Abstract

Quantum annealing is an emerging metaheuristic used for solving combinatorial optimisation problems. However, hardware based physical quantum annealers are primarily limited to a single vendor. As an alternative, we can discretise the quantum annealing process (discretised quantum annealing or DiQA) and use it on gate-model quantum computers. In this work, we first benchmark DiQA against simulated annealing for a similar number of steps. We then propose and evaluate a hybrid quantum-classical heuristic: Quantum Accelerated Simulated Annealing (QASA), where the traditional classical annealing procedure can be sped up with the use of (relatively) low depth DiQA circuits. This is done by (i) running a partial annealing scheme with a fraction of the depth of the complete circuit (ii) sampling the results from the circuit and fitting a Gibbs distribution on it and (iii) Using the inverse temperature of the Gibbs distribution and the best sample to initialise simulated annealing (SA). Our simulation results show QASA performing comparably to SA but for a reduced amount of steps. With the promising results of our work, we hope to generate interest for potential future work in the area of fixed-parameter quantum optimisation.

1 Introduction

Combinatorial optimisation is one of the most promising use-cases for quantum computing and there have been many heuristics and approximate algorithms proposed for both near and far term quantum hardware. Among them, quantum annealing (QA) [1] is one of the most popular and mature metaheuristics for solving combinatorial optimisation problems with proposed applications in many domains from life-sciences, logistics, telecommunications and many more [2].

However, while classical Monte-Carlo simulations of quantum annealing is a popular and well-studied topic, the only vendor that successfully implemented this metaheuristic at the hardware level is D-waveTM whose physical quantum annealers are analogue computers that implement QA as a physical process. In 2023, their quantum annealer was shown to be faster than simulated quantum annealing (SQA) for crystal dynamics [3].

Since the gate-model of quantum computing is universal in nature [4], an alternative to hardware-based quantum annealers could be to implement quantum annealing on gate-model quantum computers [5][6][7]. Broadly, we can call them discretised quantum annealing (DiQA) schemes since the model of computation considered here is not analogue. It should be noted that while different schemes to do the above have been proposed, the primary method for optimisation on gate-model quantum computers (QCs) have been variational quantum algorithms, such as the quantum approximate optimisation algorithm [8].

While we acknowledge that a physical quantum annealer would probably outperform a DiQA scheme on gate-model QCs for most of the scenarios (due to the maturity, longer coherence times and application-specificity of the former's

hardware); it is important to study DiQA as an alternative especially since the former is exclusive to only one commercial vendor. In this work, our aim is to do an exploration of DiQA to get insights on how it may be leveraged in the future to solve non-trivial problems. This means one of the focus areas is on using relatively low depth circuits* as a subroutine in a larger optimisation heuristic. Our results can be divided into two parts.

First, we benchmark DiQA and simulated annealing for randomly generated Ising problems for variable sizes ranging from $N = 10$ to $N = 22$. The results show DiQA outperforming simulated annealing for a discretised annealing schedule of 200 steps each. These would be the baseline results against which we compare other results in our work.

We then propose and evaluate a hybrid quantum-classical heuristic for optimisation. It involves running a fraction of a DiQA circuit and sampling from it in the computational basis. Then we fit a Gibbs distribution on the observed bitstrings to obtain the inverse temperature β . Finally, the temperature β and the best solution obtained from the distribution are supplied as initial parameters for simulated annealing. Initial evaluation against the results from the runs of non-hybrid SA suggests how such heuristics may be able to speed up classical techniques (with the help of a low depth quantum subroutine).

The rest of this paper is structured as follows: section 2 covers the necessary background and related work, section 3 covers the results of the set of initial experiments on DiQA

*With respect to the depth of circuit calculated if the annealing schedule of a standard run of D-wave was to be properly converted.

and simulated annealing, section 4 is on our hybrid quantum-classical heuristic. We briefly discuss different possible directions that can be taken in the future in section 5 and we finish the paper with section 6: the conclusion.

2 Background and Related Work

2.1 Background

Here, we describe the necessary background and notations for this work.

2.1.1 Quantum Computing: Quantum Computing can be defined as leveraging quantum mechanical phenomena like superposition, entanglement, and quantum measurement for computational purposes [9]. The smallest unit of information is a two level quantum state known as a qubit, which can be represented as

$$|\psi\rangle = \mu|0\rangle + \nu|1\rangle \quad (1)$$

$$\text{where } |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (2)$$

Where μ and ν are the amplitudes associated with $|0\rangle$ and $|1\rangle$ respectively, with $|\mu|^2 + |\nu|^2 = 1$. One of the most popular models of quantum computing is gate-model quantum computing, where quantum computing is represented in terms of discrete steps that manipulate a set of qubits (leveraging the above mentioned quantum phenomenon). We recommend the text by Nielsen and Chuang for a more detailed introduction of the subject [9].

2.1.2 The (classical) Ising model: This is a model from statistical mechanics, primarily used for modelling ferromagnetic materials, such as iron [10]. Computationally, it is an NP-Hard problem [11] and in the recent years, the objective function associated with it has been used to represent hard problems from other domains * [13, 14].

In the two-dimensional case, the energy or objective function of the classical Ising model can be described as follows

$$E(\sigma) = \sum_i h_i \sigma_i + \sum_{i < j} J_{ij} \sigma_i \sigma_j \quad (3)$$

Where σ is a set of N bipolar variables and $\sigma_i \in \{-1, +1\}$, $i = 1, 2, \dots, N$. The primary goal here is to encode a combinatorial optimisation problem of interest as a minimisation or maximisation problem of the Ising objective function (mainly the former), send it to a quantum optimiser like the D-waveTM's physical quantum annealer, and then decode the results back to the original problem.

An equivalent objective function to Eqn(3) is the quadratic unconstrained binary optimisation (QUBO) objective function, where the variables have binary values. Since our work

deals only with Ising problems, we mention the above for the sake of completeness. But readers interested in this topic are recommended to read the book by Crama and Hammer [15].

2.1.3 Quantum annealing (QA): Quantum annealing is a meta-heuristic that attempts to find the groundstate (or the global minima) of a problem Hamiltonian*. The most well known problem Hamiltonian is the one for two-dimensional classical Ising Hamiltonian \hat{H}_P .

$$\hat{H}_P = \sum_i h_i \hat{\sigma}_i^{(z)} + \sum_{i < j} J_{ij} \hat{\sigma}_i^{(z)} \hat{\sigma}_j^{(z)} \quad (4)$$

$$\text{where } \hat{\sigma}_i^{(z)} = (\otimes_1^{i-1} \hat{I}) \otimes (\hat{\sigma}^{(z)}) \otimes (\otimes_{i+1}^N \hat{I}) \quad (5)$$

$$\text{and } \hat{\sigma}^{(z)} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (6)$$

Eqn(4) is the Hamiltonian for the Ising energy function described in Eqn(3). The process of quantum annealing at any given time can best be described by the following Hamiltonian

$$\hat{H}(s) = -\frac{A(s)}{2} \hat{H}_M + \frac{B(s)}{2} \hat{H}_P \quad (7)$$

$$\text{where } s \in [0, 1] \quad (8)$$

Here, s is the normalised time parameter. $A(s)$ and $B(s)$ are functions known together as the quantum annealing schedule. \hat{H}_M is the starting Hamiltonian for which we can easily prepare the ground state. The most popular choice for this Hamiltonian is

$$\hat{H}_M = \sum_i \hat{\sigma}_i^{(x)} \quad (9)$$

$$\text{where } \sigma^{(x)} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (10)$$

At $s = 0$, all N variables involved have an equal probability of being either +1 or -1, $A(s)$ is very high and $B(s) \sim 0$. As s goes from 0 to 1, $A(s)$ decreases and $B(s)$ increases, allowing the quantum state to go from the ground state of \hat{H}_M to the ground state of \hat{H}_P as long as the process is done slowly[†]. Interested readers are recommended the paper by Kadowaki and Nishimori to understand this topic in depth [1]. This process is implemented either as a physical quantum annealer [17] or as classical simulations [18] with the former recently showing a speedup against the latter for crystal dynamics [3].

2.1.4 Quantum approximate optimisation algorithm (QAOA): The most popular algorithm for combinatorial optimisation on the gate-model computer is the quantum approximate optimisation algorithm (QAOA) [8]. It involves the Hamiltonian

*The matrix that describes the total energy of a physical system

[†]Governed by the principle of adiabatic evolution [16].

*Not all NP-hard problems can be reduced to the Ising model. But all problems in NP can be reduced to the Ising model (some of them which are also NP-Hard), that too in polynomial time [12].

simulations of \hat{H}_M and \hat{H}_P as

$$U(\hat{H}_M, \delta) = e^{-i\delta\hat{H}_M} = \prod_i e^{-i\delta\sigma_i^{(x)}} \quad (11)$$

$$U(\hat{H}_P, \gamma) = e^{-i\gamma\hat{H}_P} = \prod_i e^{-i\gamma h_i \sigma_i^{(z)}} \prod_{i<j} e^{-i\gamma J_{ij} \sigma_i^{(z)} \sigma_j^{(z)}} \quad (12)$$

where $\delta \in [0, \pi]$ and $\gamma \in [0, 2\pi]$ are angles* applied in simulating the Hamiltonian. These Hamiltonian simulations can be performed by a combination of rotation and CNOT gates described below briefly

$$R_x(\omega) = e^{-i\frac{\omega}{2}\hat{\sigma}^{(x)}} = \begin{pmatrix} \cos \omega/2 & -i \sin \omega/2 \\ -i \sin \omega/2 & \cos \omega/2 \end{pmatrix} \quad (13)$$

$$R_z(\omega) = e^{-i\frac{\omega}{2}\hat{\sigma}^{(z)}} = \begin{pmatrix} e^{-i\omega/2} & 0 \\ 0 & e^{i\omega/2} \end{pmatrix} \quad (14)$$

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (15)$$

For this algorithm, the system also begins in an equal superposition of all possible states

$$|\psi_0\rangle = \frac{1}{\sqrt{2^N}}(|0\rangle + |1\rangle)^{\otimes N} \quad (16)$$

A set of p pairs of alternating Unitary operators are applied (based on equations 11 and 12) for angles $\Delta = (\delta_1, \delta_2, \dots, \delta_p)$ and $\Gamma = (\gamma_1, \gamma_2, \dots, \gamma_p)$

$$|\psi_p\rangle = U(\hat{H}_M, \delta_p)U(\hat{H}_P, \gamma_p)\dots U(\hat{H}_M, \delta_1)U(\hat{H}_P, \gamma_1)|\psi_0\rangle \quad (17)$$

$$\text{or } |\psi_p\rangle = U_p U_{p-1} \dots U_2 U_1 |\psi_0\rangle \quad (18)$$

$$\text{where } U_l = U(\hat{H}_M, \delta_l)U(\hat{H}_P, \gamma_l), \quad (19)$$

Here $l = 1, 2, \dots, p$. This parameterised circuit is known as the QAOA-ansatz and p is known as QAOA-depth[†]. The task is to figure out the optimal values of these p angles that would maximise the probability of the ground state(s) to be measured[‡]. This is typically done in a variational manner where the expectation value of the quantum state is calculated and the are parameters adjusted accordingly [19]. In our work, we do not variationally update the parameters to find the best values. Instead, we use the procedure described next.

*Typically represented as β and γ , our change in notation is to avoid ambiguity with another value also represented by β .

[†]Not to be confused with the related concept of gate depth. Each Unitary matrix from U_1 up to U_p would have a fixed number of gates in it, depending on the problem and the QPU architecture.

[‡]For converting the measured qubit into bipolar values, $|0\rangle \rightarrow +1$ and $|1\rangle \rightarrow -1$ (essentially the eigenstates and eigenvalues of a $\sigma^{(z)}$ matrix).

2.1.5 Discretised Quantum Annealing (DiQA): Like mentioned before, this is a general approach to express an analogue computational process like QA to convert it into discrete unitary operations. Our version of DiQA makes use of the second-order Suzuki-Trotter decomposition mentioned in the work by Willsch et al. [7]. Essentially, given a quantum schedule defined by the tuple of functions $(A(s), B(s))$, we can discretise the QA process into p steps of the QAOA-ansatz by the following equations:

$$\delta_l = \frac{-\tau(A(s_{l+1}) + A(s_l))}{2}, \quad l = 1, 2, \dots, p-1 \quad (20)$$

$$\delta_p = \frac{-\tau A(s_p)}{2} \quad (21)$$

$$\gamma_l = B(s_l), \quad l = 1, 2, \dots, p \quad (22)$$

Here, the normalised time parameter is also discretized into p steps. τ represents the amount of time taken per step. We use the annealing schedule for dwave_2000Q_6 machine [20] in our experiments. These schedules are measured in Ghz* and so, for correctly converting these frequencies into angles, we need to multiply $A(s)$ and $B(s)$ with a factor of $1/4\pi$ each[†]. In the context of the D-waveTM's annealing schedule, τ is measured in nanoseconds. The value for τ we use for all our experiments down below is 0.8ns.

2.1.6 Simulated Annealing (SA): Simulated Annealing is a family of heuristics that aim to optimise a cost function by randomly sampling from the solution space of an objective function and then accepting or rejecting a new sample based on the current sample and a temperature parameter [21]. In the context of our work, the objective is to minimise Eqn(3). The process starts with a inverse temperature parameter $\beta = \beta \sim 0$ and ends with $\beta = \beta_{max} \gg 0$. Specifically, for all our experiments, we use $\beta_0 = 0.01$ and $\beta_{max} = 100$ with an exponential schedule given by the set T_{inv} .

$$T_{inv} = \{\beta_i | \beta_i = \beta_0 \left(\frac{\beta_{max}}{\beta_0}\right)^{i/b} \text{ and } 0 \leq i \leq b-1, i \in \mathbf{Z}\} \quad (23)$$

where b is the number of steps for which the schedule is defined. At each step of the annealing process, a number of monte carlo steps of sampling can be taken, governed by the *sweeps* parameter. Further details are presented in Algorithm 1.

2.1.7 Gibbs distribution: A Gibbs or Boltzmann distribution is a probability distribution that is defined using an energy

*the original energies were divided by the Plancks constant h

[†] $1/2\pi$ for converting frequency into radians/sec and an additional $1/2$ because of Eqn(7).

Algorithm 1 Basic Simulated Annealing(pseudocode)

```
1: procedure MAIN( $h, J, sweeps, T_{inv}$ )
2:   Randomly initialise all Ising spins  $\sigma_i \in \{-1, +1\}, 1 \leq i \leq N$ 
3:   repeat
4:     Get next  $\beta$  from  $T_{inv}$ 
5:     for  $k$  from 1 to  $sweeps$  do
6:       Select a variable  $i$  randomly
7:       Update  $\sigma \leftarrow \text{MCSTEP}(\sigma, i, h, J, \beta)$ 
8:     end for
9:   until all  $\beta$ s in  $T_{inv}$  are covered
10:  return  $\sigma$ 
11: end procedure
12: procedure MCSTEP( $\sigma, i, h, J, \beta$ )
13:  Calculate current state energy  $E_{curr}$  with Eqn(3)
14:  Flip sign  $\sigma_i \leftarrow -\sigma_i$  and calculate the proposed state energy  $E_{prop}$ 
15:  Calculate  $E_{diff} = E_{prop} - E_{curr}$ 
16:  if  $E_{diff} < 0$  then
17:    return  $\sigma$  ▷ Accept move
18:  else
19:    Randomly sample a value  $r$  from  $U(0, 1)$ 
20:    if  $r < e^{-\beta E_{diff}}$  then
21:      return  $\sigma$  ▷ Accept this move as well
22:    else
23:      Flip  $\sigma_i \leftarrow -\sigma_i$  ▷ Reject move, revert back
24:      return  $\sigma$  ▷ Return old  $\sigma$ 
25:    end if
26:  end if
27: end procedure
```

function and the inverse temperature parameter β .

$$P(\sigma) = \frac{e^{-\beta E(\sigma)}}{Z} \quad (24)$$

$$\text{where } Z = \sum_{\{\sigma\}} e^{-\beta E(\sigma)} \quad (25)$$

Where Eqn(25) describes the partition function, which is intractable to calculate in most cases. In our case, the variables are bipolar and the energy function is defined by Eqn(3).

2.2 Related work

As mentioned above, there have been various works that have discretised quantum annealing for circuit based quantum computers, predominantly for the QAOA-ansatz [5][6][7]. In our work, we use DiQA (i) to assess its performance against simulated annealing and (ii) as a subroutine to speed up simulated annealing.

At a glance, the work on Quantum-enhanced Markov chain Monte Carlo (QE-MCMC) [22] seems to be similar to our proposed QASA technique. There are however a few differences: (i) QE-MCMC is a generalised framework focused on sampling a distribution whereas with QASA our end goal is to get the best solution for a cost Hamiltonian, (ii) QE-MCMC is

very broad in its scope of potential applications whereas QASA is more focused on solving combinatorial optimisation problems (iii) QE-MCMC is an iterative algorithm that switches between the quantum and the classical step (with a feedback to the quantum step), whereas QASA does not loop between its quantum and classical steps. We recommend interested readers the work on QE-MCMC by Layden et al. [22] as well. Another directly related work to QASA is the work that uses the D-waveTM physical quantum annealer and neural networks for accelerating MCMC on the Ising model [23].

Our QASA heuristic is inspired by the work of França and Patron [24]. In that work, the authors estimate the quality of the results of a quantum optimiser by fitting its outputs to a Gibbs distribution and obtaining the corresponding β (larger the value, the better). In this work, we use a similar idea, but use the β and a bipolar bitstring σ from DiQA with (one of) the lowest energy as the starting parameters for SA.

It is also important to mention the work done on neutral-atom quantum computers that can do analogue or dual analogue/discrete quantum computing. In brief, they have the ability to perform analogue Hamiltonian evolution for applications in combinatorial optimisation [25][26]. Though our demonstration of QASA is with DiQA, the basic procedure would also be able to incorporate analogue quantum computing as well.

3 Preliminary Experiments

3.1 Experiment setup

We generated 100 random Ising problems with a 6-regular graph interactivity* between the variables for sizes $N = \{2i | 5 \leq i \leq 11, i \in \mathbf{Z}\}$. The coefficients are uniformly sampled in the range $h_i \in [-2, 2]$ and $J_{ij} \in [-1, 1]$ rounded to one decimal place. For the purposes of the experiments, we found the groundstate(s) for each problem by brute-forcing every possible combination of variable values. This is important since the probability of successfully measuring (or reaching) a ground state \mathcal{P} is the key output metric for us.

We ran DiQA simulations with a QAOA-ansatz depth of $p = 100$ and $p = 200$ against SA with the same number of steps $b = p$ as above. For our SA runs, we also ran them for two settings (i) one sweep per β and (ii) N sweeps per β for both $b = 100$ and $b = 200$ steps. All quantum simulations in this work were conducted using noiseless simulators with all-to-all qubit connectivity using the QISKIT package for the Python programming language [27] and computers with standard x86-64 processors. The success probability \mathcal{P} was calculated with the help of the resultant statevectors for DiQA. This is done to get an accurate estimate of what \mathcal{P} would be in ideal conditions. For SA however, there is no direct equivalent of a statevector that we can keep track of, thus we need to approximate the success probability. We do this by running each problem 2000 times, counting the number of times we

*The justification being that the first quantum annealers were based on the chimera graph, a structure with an average degree of six [13].

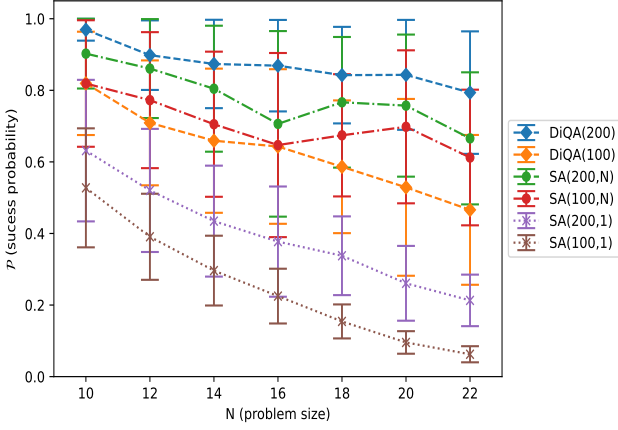


Fig. 1: Success Probability \mathcal{P} with respect to the problem size N for different heuristics and parameters. The parameter denoted for DiQA is the circuit depth p for which it was run. For SA, the first parameter indicates the number of annealing steps taken, and the second indicates the number of sweeps per anneal step.

attain a ground state (at the end of a run) and dividing it by 2000.

3.2 Results and discussion

The results shown in figure 1 show the success probabilities \mathcal{P} as the number of variables N increase. All points on the plot are median values of the 100 problems and the error bars are median absolute deviations (MAD). The general trend of \mathcal{P} reducing as problem size N increases is to be expected, since the heuristics were run with fixed (for DiQA and SA) or linearly increasing resources (for SA only).

The worst performing heuristic here is simulated annealing when its allowed to only take one sweep per β , even though the number of steps taken by it was similar to DiQA. When we increase the number of sweeps to N per β we can see SA with 100 steps and N sweeps perform better than DiQA with $p = 100$. However, we see DiQA with $p = 200$ marginally outperforming SA with $b = 200$ steps and N sweeps. The improvement in performance becomes more significant when considering the run-time resources increase linearly for SA (for N sweeps) but are fixed for DiQA.

The results of these preliminary experiments would be useful for helping us with understanding the performance of our QASA hybrid quantum-classical heuristic. While previous comparisons between quantum annealing and SA have existed [28][29][30], comparing a discretised version of quantum annealing (meant for gate-model quantum computers) with SA is still a useful result for the community. However, it is also important to keep in mind that these results shouldn't be taken as a proof of quantum advantage over classical annealing.

4 Quantum Accelerated Simulated Annealing (QASA)

Our hybrid classical-quantum heuristic is aimed at leveraging a lower depth quantum circuit (relative to the depth of a standard DiQA run) to help with combinatorial optimisation tasks. Figure 2 shows this process pictorially. But a detailed description of the process is given next.

4.1 Description

4.1.1 Partial annealing with DiQA: In the very first step, a QA schedule is discretised into p steps. The first p' steps are chosen ($p' \leq p$) and a quantum circuit is made according to the rules of DiQA. At the end of the circuit we measure the outputs and get the probability distribution of measuring a particular bitstring back.

4.1.2 Calculate the inverse temperature $\beta_{p'}$: In this step, the $\beta_{p'}$ value will be calculated by fitting the probabilities from the step before as a Gibbs distribution as described in Eqn(24). However, since the partition function Z is intractable to calculate in most cases, we can eliminate it from the equation by dividing Eqn(24) for two Ising configurations*: $\sigma^{(1)}$ and $\sigma^{(2)\dagger}$

$$\frac{P(\sigma^{(1)})}{P(\sigma^{(2)})} = \frac{e^{-\beta_{p'} E(\sigma^{(1)})}}{e^{-\beta_{p'} E(\sigma^{(2)})}} \quad (26)$$

Taking the natural logarithm on both sides, we get

$$\ln P(\sigma^{(1)}) - \ln P(\sigma^{(2)}) = \ln e^{-\beta_{p'} E(\sigma^{(1)})} - \ln e^{-\beta_{p'} E(\sigma^{(2)})} \quad (27)$$

$$\text{or, } \beta_{p'} = \frac{\ln P(\sigma^{(1)}) - \ln P(\sigma^{(2)})}{E(\sigma^{(2)}) - E(\sigma^{(1)})} \quad (28)$$

Thus, if we have two bitstrings ($\sigma^{(1)}, \sigma^{(2)}$) with significant observable probabilities, we can use them to calculate $\beta_{p'}$. In the ideal scenario, the probability distribution is a native Gibbs distribution and so any two pairs of bitstring could be used to calculate the inverse temperature. However, when this is not the case, different bitstring pairs can produce different β values. In this case, we recommend taking the average or median from all calculated β s. We shall see a demonstration of this in our experiments below.

4.1.3 Run (shortened) simulated annealing: With $\beta_{p'}$ and the bitstring for lowest energy found so far, we can now run SA for a shortened amount of steps up to β_{max} (assuming $\beta_{p'} < \beta_{max}$). The DiQA procedure done prior to this is aimed at reducing the number of steps that SA needs to take (while not incurring a large overhead in itself).

*A trick borrowed from SA itself [21].

[†]Here $\sigma_{(1)}$ is a set of N bipolar values, essentially (one of) the Ising configuration for all N variables. Not to be confused with σ_1 which holds the bipolar value (scalar) of the first variable.

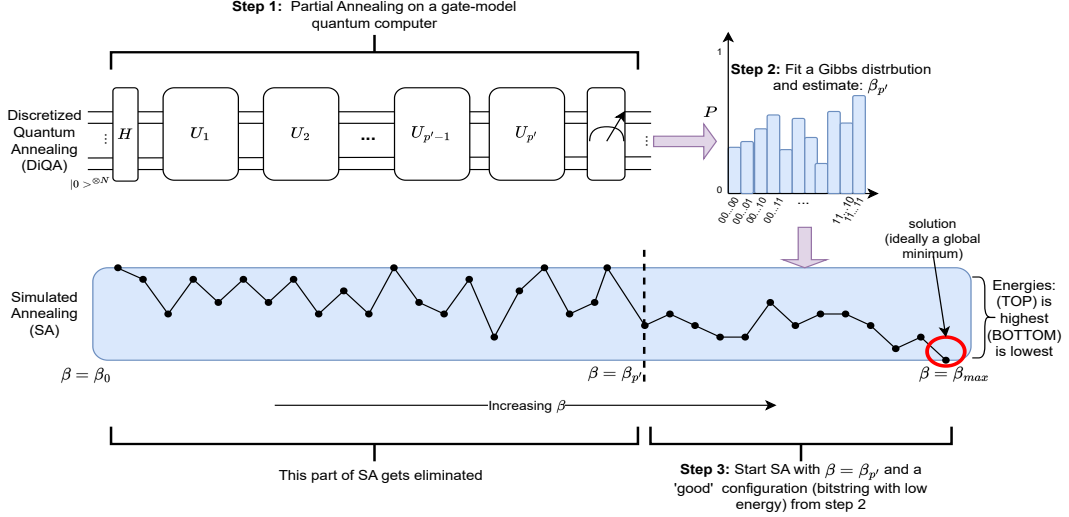


Fig. 2: Block diagram showing the workflow of QASA. The quantum circuit in step 1 starts with applying hadamard gates on individual qubits (all initialised to $|0\rangle$), followed by Unitary operators up to p' depth. The Y axis of the plot (not to scale) for step 2 shows the probability of each measured bit string. The illustration of the SA block is aimed at showing a basic classical annealing procedure that starts at a small inverse temperature β_0 and ends at β_{max} , where $\beta_{p'}$ lies somewhere in between the process. The line inside the SA block is a basic illustration of how the energy values of the system may vary as the heuristic progresses.

4.2 Experiment setup

For experiments to evaluate QASA, the 700 Ising problems described in section 3.1 were used again. With this, we can compare and contrast the results from the two different experiments.

The annealing schedule of `dwave_2000Q_6` was discretised for $p = 200$ steps. From that, we ran DiQA circuits for the first $p' = 50$ and $p' = 100$ steps*.

Since the probability distribution we receive from our partial runs of DiQA may not be native Gibbs distributions. In order to calculate $\beta_{p'}$ for each problem, we use an approximation method. This is done by calculating $m = 50$ bitstrings with the largest probabilities. But since our experiments were done on a small N size, it is generally likely that the largest probability configuration is the ground state. This would make rest of the computation trivial. Thus, for the purposes of this particular experiment, we purposely picked the second-largest probability[†] bitstring as σ^1 and pair it with all the other $m - 1$ bitstrings individually to make $m - 1$ $(\sigma^1, \sigma^{(2)})$ tuples. Finally, we calculate the β_p value for each of these tuples by Eqn(28) and take the median value to represent the $\beta_{p'}$ for a problem.

The SA schedule T_{inv} for $b = 200$ steps is used here. Based on the closest value on the SA annealing schedule to $\beta_{p'}$, we calculate the starting step number b' in the schedule. We use N sweeps per step in this truncated SA procedure.

*We also ran it for $p' = 150$ steps, but only the results from the prior two were used in the full QASA heuristic.

[†]In order to show at least a modicum of robustness in QASA.

4.3 Results and discussion

The results of our QASA experiments in figure 3 show QASA with $p' = 50$ perform similarly to SA[‡] with $b = 200$ in figure 1. For $p' = 100$, we see the results from QASA being the best across all of the tests for the set of general-case problems (created in section 3). While these results are not conclusive on their own, they do show that a technique like QASA can help in optimisation tasks in the near to medium term.

Like mentioned previously, although we ran the complete QASA heuristic for $p' = 50$ and $p' = 100$, we calculated $\beta_{p'}$ for $p' = \{50, 100, 150, 200\}$ to get information about how different p' values affect the inverse temperatures. Figure 4 shows the boxplots of $\beta_{p'}$ and the starting position (or step) b' for SA. We can see that although the median $\beta_{p'}$ for $p' = 50$ and $p' = 100$ is relatively close to zero (≈ 0.1815 and ≈ 1.34812 respectively), the median b' is 64 and 108 due to the exponential SA schedule. Another effect of the exponential schedule can be seen by the outliers for the boxplots of $p' = 150$ and $p' = 200$; for $\beta_{p'}$ they appear on top, but they appear on the bottom in the plot for b' .

It must be mentioned that since SA needs multiple Monte-carlo steps (or sweeps) per anneal step, the improvement for DiQA $p' = 50$ and 100 is much larger than 14 and 8 anneal steps respectively. Especially since DiQA can employ parallelism of quantum gates for implementing much of its Unitary operator U_i per anneal step. An in-depth analysis of the actual advantage in time saved would also depend on a quantum processing unit's (QPU) (i) qubit connectivity and its ability to do (ii) single control multiple target CNOT gates. Likewise, even

[‡]SA is run for N sweeps per step in both cases

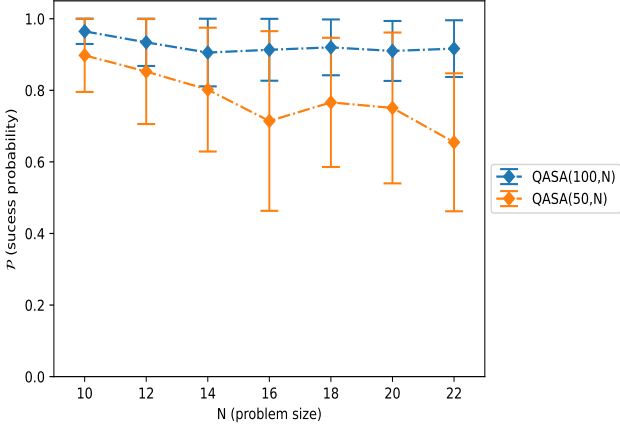


Fig. 3: Success probability \mathcal{P} for QASA when DiQA is run with $p' = 100$ and $p' = 50$, followed by SA for N sweeps per step (labelled as QASA(100, N) and QASA(50, N), respectively). Here the SA process is discretized into $b = 200$ anneal steps but we skip the first $b' - 1$ steps based on $\beta_{p'}$ which is calculated at the end of DiQA.

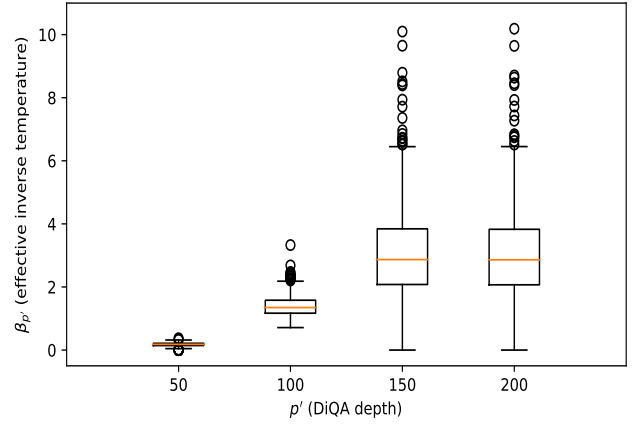
though running the DiQA circuit for $p' = 150$ only gives us a starting position of $b' = 124$, it can still have a time advantage because of how differently an (efficient) anneal step needs to be implemented on a quantum circuit as opposed to a classical computer. It should be noted that at $p' = 200$, figure 4 stops showing any improvement at all (in $\beta_{p'}$ or b'). This can be due to several factors including (i) the highly variable results at higher depths as suggested by the error bars in figure 1, (ii) the limitations of our approximation technique to ascertain $\beta_{p'}$, (iii) the diminishing returns at higher p of a QAOA-ansatz [31], among others.

Now that preliminary testing for this technique has been done in this experiment, it is important to note that the approximation of $\beta_{p'}$ will be more noisy in practise due to many factors, including (i) shot-noise from sampling (ii) noise from the quantum computers. We shall discuss the latter in the next section on the possible future work.

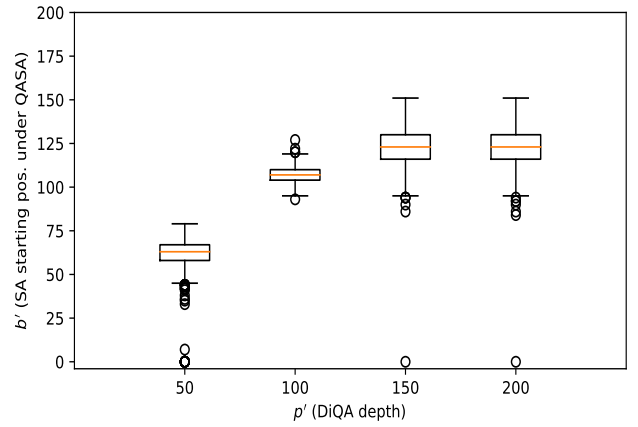
4.4 Feasibility for use on actual quantum devices: a brief comment

One advantage of using the QAOA-ansatz is the prevalence of work that has been done to evaluate its performance on real QPUs and noise-models. However, there still have not been many works that study the ansatz with a larger p .

The 2021 work by Harrigan et al. deals with variational QAOA on a superconducting-qubits based device and suggests a drop off in performance after $p = 5$ (for their machine) [19]. However, with the emergence of error-suppression and mitigation techniques, significant improvements in the results have been observed [32]. Another major positive result is the work done by Kim et al. [33], which although is not about QAOA, does deal with implementing a larger number CNOT gates for a significant depth of a circuit successfully.



(a)



(b)

Fig. 4: Boxplots depicting the spread of the $\beta_{p'}$ values (a) and the starting point b' for SA under QASA (b) for $p' = \{50, 100, 150, 200\}$. The values of $\beta_{p'}$ are approximated as mentioned in section 4.2 and the starting positions b' are calculated with respect to the exponential SA schedule.

On trapped-ion QPUs, A simulation of QAOA for 40 qubits showed that that the ‘QAOA performance does not degrade significantly as they scaled up the system size’ [34]. One of the latest works* has implemented advanced mixers for QAOA on a trapped-ion computer for $p > 1$ successfully [35]. This indicates a possibility of reducing DiQA depth by using better mixer Hamiltonians for the annealing evolution.

It appears that current QPUs may not be able to implement DiQA for a non-trivial depth size. But based on the latest work on error suppression and mitigation, DiQA may be feasible for implementation in the intermediate term (within 2 years [36], at the earliest), especially if advance mixers are used (for circuit depth reduction).

Like mentioned in section 2.2, one of the most promising quantum platform for combinatorial optimisation is the neutral-atom quantum computer where the native problem Hamiltonian

*As of the time of writing.

is slightly different to an Ising model [26]. Nonetheless, adapting QASA for neutral-atom computers can be an interesting research direction.

5 Future Work

One of the most important work to be done in the future is to study DiQA and QASA with mixers more advanced than Eqn(9). This will give us an idea about the reduction in p . As mentioned above, A version of QASA for neutral-atom quantum computers can be another interesting research direction.

Another interesting research direction to pursue would be to replace simulated annealing (SA) in QASA with simulated quantum annealing (SQA) [18] that takes the same quantum annealing schedule as DiQA. In this approach, we would not need to calculate β and can directly feed the quantum annealing parameters equivalent to the $p' + 1$ step. Of course, this approach will only work if the mixer Hamiltonian is described by Eqn(9)*.

And finally, for all future experiments, work should focus on better noise-resistant techniques to assess $\beta_{p'}$ in QASA. Performance metrics that allow for approximate answers, like approximation ratios [8] and time-to-solution [37] should also be considered. This is useful since a lot of applications of combinatorial optimisation can often suffice with a solution that may be a local minima with a sufficiently low energy rather than the global minima itself.

6 Conclusion

In this work, we did a preliminary evaluation of the usefulness of discretised quantum annealing (DiQA) on quantum circuits, primarily as an alternative to a hardware based quantum annealer. Our benchmark against simulated annealing showed better results for equal number of discretised anneal steps. Our main contribution was the hybrid quantum-classical heuristic that combined a (relatively) low depth DiQA circuit and simulated annealing with better informed parameters. Our experiments show QASA performing comparably to SA but for a reduced number of total steps (even with the overhead of DiQA).

Techniques like DiQA and its variants are not new, but they are comparatively less studied (as opposed to variational quantum algorithms). Our motivation with this work is to provide new perspectives to the research community, and to encourage more research on fixed-parameter quantum optimisation techniques, and hybrid quantum-classical heuristics that can incorporate them.

*Meaning if the annealing process is described by Stoquatic Hamiltonians.

7 Acknowledgements

The authors would like to thank Dr Charles Nicholas and the DREAM laboratory in the University of Maryland Baltimore County (UMBC), for providing access to their high performance compute server.

8 References

- [1] T. Kadowaki and H. Nishimori, “Quantum annealing in the transverse ising model,” *Physical Review E*, vol. 58, no. 5, p. 5355, 1998.
- [2] D. wave INC, “D-wave list of applications,” 2023.
- [3] A. D. King, J. Raymond, T. Lanting, R. Harris, A. Zucca, F. Altomare, A. J. Berkley, K. Boothby, S. Ejtemaee, C. Enderud, *et al.*, “Quantum critical dynamics in a 5,000-qubit programmable spin glass,” *Nature*, pp. 1–6, 2023.
- [4] D. E. Deutsch, “Quantum computational networks,” *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 425, no. 1868, pp. 73–90, 1989.
- [5] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen, “Benchmarking the quantum approximate optimization algorithm,” *Quantum Information Processing*, vol. 19, pp. 1–24, 2020.
- [6] S. H. Sack and M. Serbyn, “Quantum annealing initialization of the quantum approximate optimization algorithm,” *quantum*, vol. 5, p. 491, 2021.
- [7] D. Willsch, M. Willsch, F. Jin, K. Michielsen, and H. De Raedt, “Gpu-accelerated simulations of quantum annealing and the quantum approximate optimization algorithm,” *Computer Physics Communications*, vol. 278, p. 108411, 2022.
- [8] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” *arXiv preprint arXiv:1411.4028*, 2014.
- [9] M. A. Nielsen and I. Chuang, “Quantum computation and quantum information,” 2002.
- [10] G. Gallavotti, *Statistical mechanics: A short treatise*. Springer Science & Business Media, 1999.
- [11] F. Barahona, “On the computational complexity of ising spin glass models,” *Journal of Physics A: Mathematical and General*, vol. 15, no. 10, p. 3241, 1982.
- [12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2022.
- [13] C. C. McGeoch and C. Wang, “Experimental evaluation of an adiabatic quantum system for combinatorial optimization,” in *Proceedings of the ACM International Conference on Computing Frontiers*, pp. 1–11, 2013.
- [14] A. Lucas, “Ising formulations of many np problems,” *Frontiers in physics*, vol. 2, p. 5, 2014.
- [15] Y. Crama and P. L. Hammer, *Boolean functions: Theory, algorithms, and applications*. Cambridge University Press, 2011.
- [16] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, “Quantum computation by adiabatic evolution,” *arXiv*

preprint quant-ph/0001106, 2000.

- [17] P. I. Bunyk, E. M. Hoskinson, M. W. Johnson, E. Tolka-
cheva, F. Altomare, A. J. Berkley, R. Harris, J. P. Hilton,
T. Lanting, A. J. Przybysz, *et al.*, “Architectural con-
siderations in the design of a superconducting quan-
tum annealing processor,” *IEEE Transactions on Applied
Superconductivity*, vol. 24, no. 4, pp. 1–10, 2014.
- [18] R. Martoňák, G. E. Santoro, and E. Tosatti, “Quantum
annealing by the path-integral monte carlo method: The
two-dimensional random ising model,” *Physical Review
B*, vol. 66, no. 9, p. 094203, 2002.
- [19] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger,
F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends,
S. Boixo, *et al.*, “Quantum approximate optimization of
non-planar graph problems on a planar superconducting
processor,” *Nature Physics*, vol. 17, no. 3, pp. 332–336,
2021.
- [20] D. wave INC, “D-wave systems, qpu-specific anneal
schedules,” 2023.
- [21] D. Bertsimas and J. Tsitsiklis, “Simulated annealing,”
Statistical science, vol. 8, no. 1, pp. 10–15, 1993.
- [22] D. Layden, G. Mazzola, R. V. Mishmash, M. Motta,
P. Wocjan, J.-S. Kim, and S. Sheldon, “Quantum-
enhanced markov chain monte carlo,” *Nature*, vol. 619,
no. 7969, pp. 282–287, 2023.
- [23] G. Scriva, E. Costa, B. McNaughton, and S. Pilati,
“Accelerating equilibrium spin-glass simulations using
quantum annealers via generative deep learning,” *arXiv
preprint arXiv:2210.11288*, 2022.
- [24] D. Stilck França and R. Garcia-Patron, “Limitations
of optimization algorithms on noisy quantum devices,”
Nature Physics, vol. 17, no. 11, pp. 1221–1227, 2021.
- [25] P. Scholl, M. Schuler, H. J. Williams, A. A. Eberharter,
D. Barredo, K.-N. Schymik, V. Lienhard, L.-P. Henry,
T. C. Lang, T. Lahaye, *et al.*, “Quantum simulation of
2d antiferromagnets with hundreds of rydberg atoms,”
Nature, vol. 595, no. 7866, pp. 233–238, 2021.
- [26] M.-T. Nguyen, J.-G. Liu, J. Wurtz, M. D. Lukin, S.-
T. Wang, and H. Pichler, “Quantum optimization with
arbitrary connectivity using rydberg atom arrays,” *PRX
Quantum*, vol. 4, no. 1, p. 010316, 2023.
- [27] G. Aleksandrowicz, T. Alexander, P. Barkoutsos,
L. Bello, Y. Ben-Haim, D. Bucher, F. J. Cabrera-
Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen,
et al., “Qiskit: An open-source framework for quantum
computing,” *Accessed on: Mar*, vol. 16, 2019.
- [28] E. Crosson and A. W. Harrow, “Simulated quantum
annealing can be exponentially faster than classical sim-
ulated annealing,” in *2016 IEEE 57th Annual Symposium
on Foundations of Computer Science (FOCS)*, pp. 714–
723, IEEE, 2016.
- [29] H. N. Djidjev, G. Chapuis, G. Hahn, and G. Rizk, “Effi-
cient combinatorial optimization using quantum anneal-
ing,” *arXiv preprint arXiv:1801.08653*, 2018.
- [30] M. Streif and M. Leib, “Comparison of qaoa with
quantum and simulated annealing,” *arXiv preprint
arXiv:1901.01903*, 2019.
- [31] A. Borle, V. Elfving, and S. J. Lomonaco, “Quantum
approximate optimization for hard problems in linear
algebra,” *SciPost Physics Core*, vol. 4, no. 4, p. 031, 2021.
- [32] P. S. Mundada, A. Barbosa, S. Maity, T. Stace, T. Merkh,
F. Nielson, A. R. Carvalho, M. Hush, M. J. Biercuk, and
Y. Baum, “Experimental benchmarking of an automated
deterministic error suppression workflow for quantum
algorithms,” *arXiv preprint arXiv:2209.06864*, 2022.
- [33] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. Van
Den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel,
K. Temme, *et al.*, “Evidence for the utility of quan-
tum computing before fault tolerance,” *Nature*, vol. 618,
no. 7965, pp. 500–505, 2023.
- [34] G. Pagano, A. Bapat, P. Becker, K. S. Collins, A. De,
P. W. Hess, H. B. Kaplan, A. Kyprianidis, W. L. Tan,
C. Baldwin, *et al.*, “Quantum approximate optimization
of the long-range ising model with a trapped-ion quan-
tum simulator,” *Proceedings of the National Academy of
Sciences*, vol. 117, no. 41, pp. 25396–25401, 2020.
- [35] Y. Zhu, Z. Zhang, B. Sundar, A. M. Green, C. H.
Alderete, N. H. Nguyen, K. R. Hazzard, and N. M. Linke,
“Multi-round qaoa and advanced mixers on a trapped-ion
quantum computer,” *Quantum Science and Technology*,
vol. 8, no. 1, p. 015007, 2022.
- [36] D. Castelvecchi, “Ibm quantum computer passes calcula-
tion milestone,” *Nature*, vol. 618, no. 7966, pp. 656–657,
2023.
- [37] J. King, S. Yarkoni, M. M. Nevisi, J. P. Hilton, and
C. C. McGeoch, “Benchmarking a quantum annealing
processor with the time-to-target metric,” *arXiv preprint
arXiv:1508.05087*, 2015.