


Ferromagnetically Shifting the Power of Pausing

Zoe Gonzalez Izquierdo^{1,2,*}, Shon Grabbe,¹ Stuart Hadfield^{1,2}, Jeffrey Marshall,^{1,2}
Zhihui Wang,^{1,2} and Eleanor Rieffel¹

¹*QuAIL, NASA Ames Research Center, Moffett Field, California 94035, USA*

²*USRA Research Institute for Advanced Computer Science, Mountain View, California 94043, USA*

 (Received 22 July 2020; revised 20 January 2021; accepted 4 March 2021; published 6 April 2021)

We study the interplay between quantum annealing parameters in embedded problems, providing both deeper insights into the physics of these devices and pragmatic recommendations to improve performance on optimization problems. We choose as our test case the class of bounded-degree minimum-spanning-tree problems. Through runs on a D-Wave quantum annealer, we demonstrate that pausing in a specific time window in the anneal provides improvement in the probability of success and in the time to solution for these problems. The time window is consistent across problem instances and its location is within the region suggested by prior theory and seen in previous results on native problems. An approach to enable gauge transformations for problems with the qubit coupling strength J in an asymmetric range is presented and shown to significantly improve performance. We also confirm that the optimal pause location exhibits a shift with the magnitude of the ferromagnetic coupling, $|J_F|$, between physical qubits representing the same logical one. We extend the theoretical picture for pausing and thermalization in quantum annealing to the embedded case. This picture, along with perturbation-theory analysis and exact numerical results on small problems, confirms that the effective pause region moves earlier in the anneal as $|J_F|$ increases. It also suggests why pausing, while still providing significant benefit, has a less pronounced effect on embedded problems.

DOI: [10.1103/PhysRevApplied.15.044013](https://doi.org/10.1103/PhysRevApplied.15.044013)

I. INTRODUCTION

Quantum computing provides novel mechanisms for efficient computing but the extent of its impact is as yet undetermined. A tantalizing area of application is combinatorial optimization, where challenging instances are currently attacked by a variety of classical heuristics and where quantum heuristics have the potential to outperform these classical approaches. Here, we advance the understanding of one such heuristic, quantum annealing [1–6], deepening the theoretical picture of the roles that thermalization, adiabatic processes, and diabatic process play in quantum annealing and demonstrating the impact of annealing schedules and the interplay between quantum annealing parameters on performance, particularly on application-related problems that require embedding.

Our work builds on the theoretical picture of Marshall *et al.* [7] that explains why pausing in an appropriate time window during the anneal enables the system to thermalize better, improving the fit of the output distribution with a Boltzmann distribution and increasing the probability of success by orders of magnitude. Because

quantum annealing happens at nonzero temperature, temperature plays a significant role, along with quantum dynamics induced by varying the Hamiltonian, particularly near where the temperature and the minimal energy gap between the ground state and the first excited state are commensurate. This effect has also been studied in simulations [8] and, recently, rigorous sufficient conditions under which pausing helps have been identified in Ref. [9]. Here, we build on the above understanding, beyond the native problems studied in Ref. [7], to embedded problems.

It is well known that most problem instances—in particular, those related to applications—will not have a structure that matches that of the hardware, in which case the problems must be embedded. Embedded problems use multiple physical qubits to represent each logical qubit, with these physical qubits coupled via ferromagnetic couplings $J_F < 0$. In the embedded problems we study, we confirm an improvement in probability of success and that for this class of problem, as is found for native problems, there is a time window in which a pause reliably improves the performance across problem instances. We extend the theoretical picture of Refs. [7,9] to embedded problems, including a perturbative analysis on the effect of $|J_F|$ on the minimal energy gap between the ground and the first excited states. Our gap analysis and numerical

*zgonzalez@usra.edu

simulations on small systems show that as $|J_F|$ increases, the minimal gap shifts earlier and the gap size decreases. This extended picture explains why one would expect a shift of the optimal pause location to earlier in the anneal with increasing $|J_F|$ and also a somewhat less pronounced improvement from pausing on embedded problems than on native problems.

We choose as our test case the class of bounded-degree minimal-spanning-tree (BD-MST) problems, seen in a variety of application areas such as a broad spectrum of network-related problems. We demonstrate that small instances of these problems can be embedded and successfully solved by state-of-the-art quantum annealers and confirm the results predicted by our theoretical picture. We demonstrate that for the best parameters, pausing improves not only the probability of success but also the time to solution (T_S). To obtain these results, we use the newly added extended range feature of the D-Wave 2000Q to enable the use of stronger ferromagnetic couplings relative to the problem-instance couplings. Because of the asymmetry in the extended range, we cannot use the standard gauge approach to randomize the effect of qubit biases in the D-Wave 2000Q on the annealing runs. We develop a partial gauge approach that enables us to obtain much cleaner results and substantially better probabilities of success than running without partial gauges.

The rest of the paper is organized as follows. In Sec. II, we review background information on spanning-tree problems and on quantum annealing. In Sec. III, we describe the specifics related to the hardware, the instances, and the parameters for our runs, and the metrics we use to evaluate them. Section IV is devoted to results on the annealer. Results for annealing without pause are shown in Sec. A, how pausing can be helpful is demonstrated in Sec. B and how pausing shifts with $|J_F|$ in Sec. C. The technical treatment that enables the conclusive results, partial gauges, is discussed in Sec. D. We provide theoretical analysis and a physical picture for the shifting of optimal pause location with $|J_F|$ in Sec. V. In Sec. VI we summarize the results and discuss future work.

II. BACKGROUND

We review background material on spanning-tree problems and on quantum annealing.

A. Spanning-tree problem classes

Definition A.1. *A spanning tree for a graph G is a subgraph of G that is a tree and contains all vertices of G .*

Spanning trees are important for several reasons. They play a critical role in designing efficient routing algorithms. Some computationally hard problems, such as the

Steiner-tree problem and the traveling-salesperson problem, can be solved approximately using spanning trees [10]. Spanning-tree problems also find broad applications in network design, bioinformatics, etc.

One flavor of the spanning-tree problem is the weighted spanning-tree problem: given a connected undirected graph $G = (V, E)$ and a set of weights w_{uv} for each edge $(uv) \in E$, we seek a spanning tree $T \subset E$ such that the tree weight $\sum_{(uv) \in T} w_{uv}$ is minimized.

For general graphs, the determination of whether there exists a spanning tree of weight W can be carried out in polynomial time and different efficient algorithms exist to find a minimum-weight tree; for example, Kruskal's algorithm requires time $O(|E| \log |V|)$ [11]. (Special classes of graphs can be solved even faster.) On the other hand, with the additional constraint that the maximum vertex degree of the spanning tree found is at most Δ , even deciding whether there exists such a spanning tree becomes nondeterministic polynomial-time (NP) complete for fixed $\Delta \geq 2$ [12]. In this work, we focus on the bounded-degree maximum-spanning-tree (BD-MST) problem.

1. The BD-MST problem

Given an integer $\Delta \geq 2$ and graph $G = (V, E)$ with edge weights w_{uv} , $(uv) \in E$, find a minimum-weight spanning tree of maximum degree at most Δ .

We refer interested readers to Appendix B and the references therein for approximation complexity theory related to the BD-MST problem.

B. Solving on a quantum annealer

Quantum annealing is a quantum metaheuristic for optimization. Quantum annealers are quantum hardware that is designed to run this metaheuristic. Any classical cost function $C(x)$ that is a polynomial over binary variables $x \in \{0, 1\}^n$ can, with the addition of auxiliary variables, be turned into a quadratic cost function. Problems with quadratic cost functions over a binary variable without additional constraints are called quadratic unconstrained binary optimization (QUBO) problems. Quantum annealing is carried out by evolving the system under a time-dependent Hamiltonian $H(s) = A(s)H_D + B(s)H_C$, where H_D is a driver Hamiltonian, most commonly $H_X = -\sum_i X_i$, and H_C is an Ising Hamiltonian derived from a classical cost function. There is a straightforward mapping between QUBO and Ising problems. The parameter s is a dimensionless time parameter that ranges from 0 to 1, with $A(s)$ and $B(s)$ determining the form of the anneal schedule. As we will see, many different schedules $s(t)$ are possible. More information about quantum annealing generally, including mappings of problems to QUBO, can be found in Refs. [13–15].

For most application problems, on hardware with a restricted qubit connectivity, the resulting QUBO problem must further be embedded to conform with the hardware connectivity; graph minor embedding enables coupling between logical qubits in the QUBO graph by representing each logical qubit by a set of physical qubits ferromagnetically coupled with a magnitude of $|J_F|$ among them to promote collective behavior (J_F is always negative, so we typically refer to its magnitude $|J_F|$). Following standard terminology in graph theory, each such set of physical qubits is called a *vertex model* for its corresponding logical qubit. When embedding, we use the same coupling strength $|J_F|$ for all the couplings within a vertex model. Problems that do not require embedding because their structure matches that of the hardware are called *native problems* for that hardware.

While $|J_F|$ can be set to a large value such that the embedded problem preserves the ground state of the logical problem, and analytical bounds on this value can be obtained [16], too large a $|J_F|$ can reduce quantum annealing performance. Physically, there is an energy limit on the Hamiltonian as a whole and too large a $|J_F|$ relative to other parameters would mean that all of the problem parameters could reduce performance due to precision issues and noise in implementation. Furthermore, the energy spectrum throughout the anneal varies with the value of $|J_F|$ and its effect on the annealing often requires careful case-by-case consideration [13,14,17,18]. Thus, optimally setting the ferromagnetic coupling $|J_F|$ is a challenging task. Prior work has shown that there is a sweet spot for this value. Physically, this makes sense because a stronger $|J_F|$ makes it less likely for individual qubits within a vertex model to flip, which helps to avoid breaking the vertex model, but too large a $|J_F|$ makes it increasingly costly for the vertex model qubit values to flip together, potentially preventing the system from leaving a nonoptimal configuration.

To boost the probability of success, $|J_F|$ must strike the right balance, leading to better chances of arriving at—and staying in—the correct configuration. The D-Wave 2000Q allows asymmetric extension of the pairwise qubit coupling strengths $J_{ij} \in [-2, 1]$ (in addition to the canonical symmetric option $J_{ij} \in [-1, 1]$). One usage of this extension is to set $|J_F|$ in the extended range. We show how the extended values improve the probability of success of our problems.

The schedule $s(t)$ can significantly affect performance. Of particular interest to us are schedules that include a pause where, for some subinterval, $s(t)$ is constant (i.e., H is constant for a specified time). Marshall *et al.* [7] have observed on an ensemble of native problems that, strikingly, a pause at a location (generally) insensitive to the instance specifics boosts the probability of finding the ground state—the probability of success—by orders of magnitude. The physical picture underlying

such a universal effect is reviewed and expanded in Sec. V.

III. METHODS

Here, we discuss the specifics of the problem instances, annealing schedules and parameters, and metrics used to obtain our results.

A. Problem instances

Each BD-MST problem instance consists of a weighted graph $G = (V, E)$ and a degree bound Δ . The underlying graphs are chosen by exhausting all connected graphs with $n = |V| = 5$, which have $m = |E|$ ranging from 4 to 10. The weight sets are uniformly drawn from 1 to 7. Graphs and weight sets are combined to yield a large number of unique instances. The results are averaged over ensembles of instances. The size of the ensemble is specified for each result in Sec. IV. The complete list of graphs and weight sets can be found in Tables III and IV, respectively, of Appendix C.

A number of mappings of the BD-MST problem to QUBO can be found in Ref. [19]; here, we use the resource-efficient level-based mapping described in Appendix A. For each problem instance, the level-based mapping yields an objective function Hamiltonian H_C . Once mapped to QUBO, our $n = 5$ problems result in 20–74 logical variables, depending on m . Embedding to accommodate the limitations in the architecture of the annealer leads to a final tally of 83–485 physical qubits, with a median vertex model size between 1.5 and 7. More detailed information can be found in Table V. For the degree bound, we generally select $\Delta = 2$, resulting in problems equivalent to Hamiltonian path problems; we also test $\Delta = 3$ and our results hold for this case as well (see Fig. 10 in Appendix F).

B. Annealing parameters and schedules

We run our problems on the D-Wave 2000Q quantum annealer housed at the NASA Ames Research Center, which has 2031 qubits and a chimera graph architecture [20]. To embed the resulting QUBO instances in the D-Wave 2000Q hardware graph, we run D-Wave’s embedding-finding algorithm 30 times and use the smallest size embedding found (the fewest total physical qubits). This procedure finds an embedding for all of the graphs we consider. Detailed information about the typical size of the embedded problems for different graphs can be found in Fig. 9 in Appendix D, including the number of physical qubits and the size of the vertex models. Embedding statistics for a future D-Wave architecture (Pegasus) are also given.

The objective Hamiltonians are scaled so that the coupling strengths are in the range $[-1, 1]$. In the

embedded Hamiltonian, the extended J range is used to couple physical qubits representing the same logical qubits. We choose $|J_F|$ in the range $[1, 2]$, initially exploring all values in that range at 0.1 intervals.

We use the D-Wave 2000Q default $A(s)$ and $B(s)$, exploring two qualitatively different schedules. The first is a standard anneal, with time parameter $s(t) = t/t_a$, where t_a is the annealing time. Baseline runs are performed with this schedule and several annealing times t_a are initially tested. The shortest time allowed by D-Wave, $t_a = 1 \mu\text{s}$, is found to be optimal in terms of T_S for the instance ensembles, agreeing with previous studies for other problems [18,21,22].

The second type of schedule includes a pause. The beginning and end of the anneal are the same as in the first case but at some intermediate point s_p the Hamiltonian is held constant for some time t_p . The entire range of possible pause locations ($s_p \in [0, 1]$) is initially surveyed. A peak is reliably found (see Sec. B). Although the location of the peak is affected by $|J_F|$, it is always within the range $[0.2, 0.5]$, so further runs are limited to this region of interest, with s_p varied between 0.2 and 0.5 at 0.02 intervals. A range of pause durations t_p are also surveyed. Since shorter pause times are found to yield better T_S (see Sec. B), our runs are performed with pause duration $t_p = 1 \mu\text{s}$ unless otherwise noted. After optimal values for other parameters are found, other t_p values in the range $[0.25, 2] \mu\text{s}$ are explored.

We use the extended range of $J_{ij} \in [-2, 1]$ (in addition to the canonical symmetric option $J_{ij} \in [-1, 1]$). The asymmetry in the range with respect to zero precludes the use of a general strategy, gauge transformation (or, spin-reversal transformation), which has been shown to be very effective in reducing noise effects and obtaining higher-quality output data. This is due to the fact that, to perform a gauge transformation, each coupling J_{ij} is transformed as $J_{ij} \rightarrow \tilde{J}_{ij} = a_i a_j J_{ij}$, where the a_k are randomly chosen from $\{\pm 1\}$. If $a_i a_j = -1$ and $J_{ij} \in [-2, -1]$, the transformed coupling would need to be in $(1, 2]$, which is not available due to the asymmetry in the extended range. We design and implement a strategy, *partial gauge transformation*, that selectively applies the transformation only to couplings in the symmetric range $[-1, 1]$. For the case that only the embedding couplings are in the extended range, this is equivalent to applying a general gauge transformation to the logical problem prior to the embedding and is simple to implement. We find that the partial gauge transformation helps significantly in both boosting the probability of success and reducing the output variance. Only by employing partial gauges can we obtain results clean enough to see various features we report on, such as the positive role of an extended $|J_F|$ in the case of no pausing and the shift of the optimal pause location with $|J_F|$. Partial gauges and their effect are discussed in more detail in Sec. D.

Unless otherwise specified, all runs are performed with $t_a = 1 \mu\text{s}$, 50 000 anneals (or reads) and 100 partial gauges.

C. Metrics

We use the empirical probability of success (p_{success}) and time to solution (T_S) as our figures of merit for determining how likely a problem is to be solved, defined as

$$p_{\text{success}} = \frac{\text{number of anneals with correct solution}}{\text{total number of anneals}}, \quad (1)$$

$$T_S = \frac{\log(1 - 0.99)}{\log(1.0 - p_{\text{success}})} t_{\text{tot}}, \quad (2)$$

where the total time $t_{\text{tot}} = t_a + t_p$ is the time spent on each run, taking into account both the base annealing time t_a and the pause duration t_p .

These two measures are complementary to each other. The T_S figure of merit reports the expected time required to solve the problem with 99% confidence. While p_{success} is directly determined by and hence provides a portal to understand the underlying physical process, T_S gives a more practical measure that is universal across different parameter ranges and different solvers. A higher probability of success does not necessarily mean a lower T_S . For instance, we might get a slightly higher p_{success} by using a longer annealing time $t_a = 100 \mu\text{s}$ than a shorter one $t_a = 1 \mu\text{s}$, yet the chance of finding the solution might be higher by repeating the $t_a = 1 \mu\text{s}$ runs 100 times than doing the $t_a = 100 \mu\text{s}$ anneal once.

Because we compare results from two different schedules (baseline no-pause and pause), we also need metrics that help us examine the benefits that the latter presents over the former. To this end, we define two quantities based on the instance-wise improvement in T_S . The first one is the absolute T_S improvement, defined for each instance i as

$$\Delta T_{S_i} = T_{S_i}(\text{no pause}) - T_{S_i}(\text{pause}), \quad (3)$$

with the two T_S values calculated at their respective optimal $|J_F|$ values ($|J_F^*| = 1.6$ for the no-pause case and 1.8 for the pause case). A positive ΔT_{S_i} indicates that a pause improves upon the baseline results (i.e., reduces T_S) for that particular instance. The second one is the relative T_S improvement, defined as the ratio

$$\Delta T_{S_i}/T_{S_i} = \frac{T_{S_i}(\text{no pause}) - T_{S_i}(\text{pause})}{T_{S_i}(\text{no pause})}. \quad (4)$$

When a valid solution is not found for a specific instance and thus $p_{\text{success}} = 0$ for that instance, its corresponding T_S is infinity. If T_S for both the pause and no-pause results

are infinity, the pause is not improving upon the no-pause results, hence $\Delta T_{S_i} = 0$. When $T_S = \infty$ only for the no-pause case, pausing provides the maximum possible improvement and we set $\Delta T_{S_i} = \infty$ and $\Delta T_{S_i}/T_{S_i} = 1$. Finally, when $T_S = \infty$ only for the pause case, the opposite occurs, with $\Delta T_{S_i} = -\infty$ and $\Delta T_{S_i}/T_{S_i} = -\infty$.

After the embedded problem is run on the D-Wave, outputs with any inconsistent values on physical qubits that represent the same logical qubit—or with violated penalty terms such that the output does not encode a degree-bounded spanning tree—are considered to be invalid answers and are counted as failed runs. The retained valid answers are then verified against the exact solution of the problem, which is obtained through direct enumeration for the small problem sizes we consider. The reported data points correspond to the median, with the error bars marking the 35th and 65th percentiles. For ensembles of instances, 10^5 bootstraps are performed over the instances to obtain those values, where each bootstrap sample is drawn with replacement from the original instance ensemble until it is of the same size as the original ensemble. Median and 35th and 65th percentiles from the bootstrap samples are reported, meaning that the data points correspond to a median of medians. There are a few instances that do not solve with or without pauses; these instances are not excluded from the ensemble in our bootstrap procedure but are given a T_S of ∞ . These $\pm\infty$ values for ΔT_{S_i} and $\Delta T_{S_i}/T_{S_i}$ do not appear in our reported results, as they remain very far from the median (which we report as our data point) and from the 35th and 65th percentiles of the bootstrapped results that we present as error bars.

D-Wave returns the solution with the minimum cost it has found. To ensure the validity of this solution, we first confirm that the resulting graph is in fact a spanning tree that satisfies the degree constraint and also a true optimal solution by comparing with the true minimal cost obtained by an exact classical algorithm. Any other outcome is weighted zero toward p_{success} .

IV. RESULTS

We now present our results on the D-Wave 2000Q, including anneals without a pause (baseline) and the effect of pausing.

A. Annealing without pause, effect of $|J_F|$

We first show that the BD-MST problems we study are successfully solved on the D-Wave 2000Q using a standard annealing schedule, demonstrating the ability of a quantum annealer to solve a new class of optimization problems, and study the effect of the strength of the ferromagnetic coupling on the probability of success.

The *baseline* results are obtained with no pause and $t_a = 1 \mu\text{s}$, which is the shortest that D-Wave allows, and is chosen for consistently yielding the best T_S for ensembles

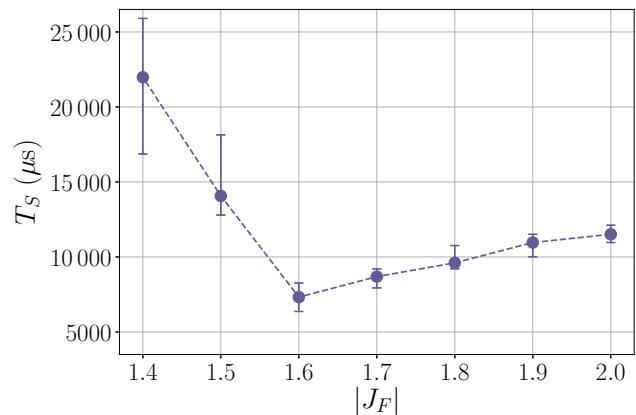


FIG. 1. The optimal $|J_F|$ for the baseline. T_S for an ensemble of 45 instances as $|J_F|$ varies. A 1- μs anneal is used. The best performance is observed at $|J_F^*| = 1.6$.

of problem instances for both this study and previously studied problems [18,21,22].

By exploring the available range of $|J_F|$ values between 1.0 and 2.0, we confirm the advantage of using the extended $|J_F|$ range and identify its optimal value for the base case at $|J_F^*| = 1.6$ with statistical significance, as shown in Fig. 1 where the probability of success is shown for a range of $|J_F|$ for the ensemble of instances.

The results might vary for groups of instances with different n ; the optimal $|J_F|$ for $n = 4$ appears to be lower, around 1.2 or 1.3. This result is obtained from only a limited number of instances and is without statistical significance. A larger number of $n = 4$ instances, and preferably also $n = 6$, would be needed to make any stronger assertions in this regard. Within $n = 5$ instances, the optimal $|J_F|$ does not appear to correlate with the logical or embedded size.

B. Improvement with a pause

After establishing the baseline with the no-pause schedule, we introduce a midanneal pause. A pause can be placed at any point in the anneal, i.e., $s_p \in [0, 1]$. Our results show that, as for native problems, the probability of success improves significantly when pausing within a specific region that is consistent across problem instances. The optimal pause location is between $0.3 \sim 0.4$, in the same range as the optimal location for the native problems studied in Ref. [7]. Figure 2 shows this improvement for an ensemble of 45 instances, with a pause of length $t_p = 100 \mu\text{s}$ and $|J_F| = 1.6$ (the optimal $|J_F|$ for the no-pause case). With the introduction of the 100- μs pause, the total time increases significantly. Our first goal is to confirm that p_{success} improves with the introduction of the pause and to identify the region where the pause is beneficial, which is shown in the top panel of Fig. 2. But

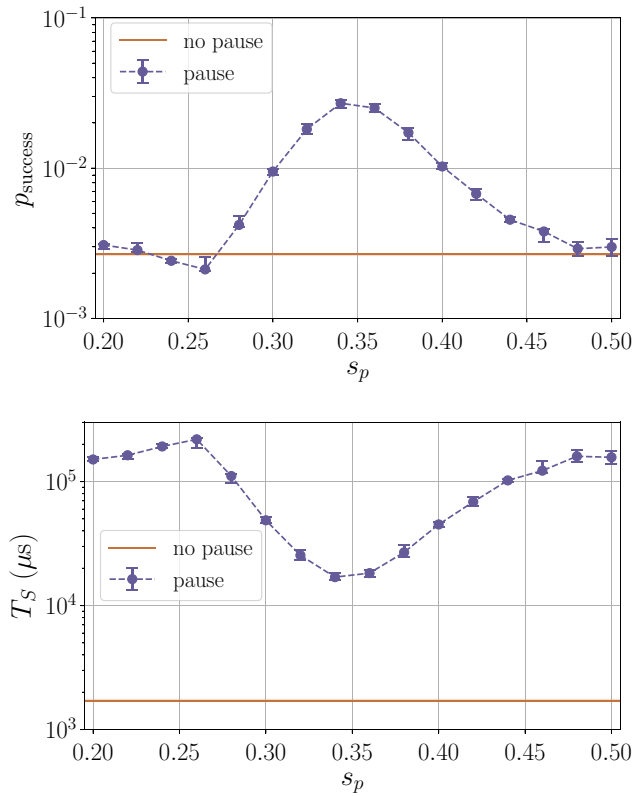


FIG. 2. The improvement of p_{success} with a pause. Top: The probability of success for an ensemble of 42 instances. $|J_F| = 1.6$ and a 1- μs anneal are used. The pause duration is 100 μs . The horizontal line shows the baseline, i.e., the no-pause results. Each data point represents the results when introducing a pause at location s_p . At the optimal pause locations, an improvement of about an order of magnitude in p_{success} is obtained. Bottom: The improvement in p_{success} due to the 101- μs anneal time (a 1- μs anneal plus a 100- μs pause) is not sufficient to overcome the extra time incurred when it comes to T_S , which becomes worse in this case. To improve T_S , we need to optimize the pause duration and location.

p_{success} alone does not necessarily provide a fair comparison between the two schedules (pause and no pause) when it comes to time. As explained in Sec. C, the improvement in the probability of success does not necessarily translate to an improvement in the time to solution (T_S); if instead of using the additional time needed for a pause it is used to repeat the anneal, that procedure may result in a higher T_S . For this reason, we also evaluate the time to solution (T_S), shown in the bottom panel of Fig. 2. In this case, due to the length of the pause, T_S increases with respect to the no-pause case. To achieve an improvement in T_S , we need to investigate shorter pause durations and optimize the pause location, an investigation we detail later in this section. The improvement in p_{success} shown in Fig. 2 confirms the physical arguments given in Sec. A.

When we examine how the optimal pause location is affected by $|J_F|$, we see that the peak in p_{success} moves

earlier with increasing $|J_F|$ but remains in this range. We also find that for instances that are unsolved in the baseline (no-pause) runs, a solution is often found after introducing an appropriate pause. For statistics on such cases, see Appendix E. These findings are given theoretical and numerical support in Sec. V. Beyond that, a correlation between the hardness of a problem and the extent of the benefit provided by pausing is not observed.

As in Ref. [7], the probability of success grows monotonically as the pause duration increases in the range $t_p \in [0.25, 100]$ μs (not shown). With respect to the expected T_S , a longer duration can cancel out improvements due to an increased probability of success. We are able to locate a sweet spot in pause duration for the various T_S metrics (Sec. C) with pause durations of $t_p = 0.75$ or $t_p = 1.0$ (Fig. 3) at pause locations $s_p = 0.30$ or $s_p = 0.32$. We now discuss these results in more detail.

The results of Fig. 3 demonstrate that a properly placed pause of a certain duration leads to a statistically significant improvement in the various T_S metrics on our ensemble of BD-MST instances. After sparsely sweeping through a range of parameters (not shown), we find that the parameter ranges $t_p \in [0.25, 2]$ μs , $s_p = 0.3$ and $s_p = 0.32$, and $|J_F| = 1.8$ deserve particular attention. These t_p values match those found in recent work [23] in which a condition on the pause duration that leads to improvement in T_S is obtained. The three panels of Fig. 3 correspond to the three metrics of Sec. C: (1) the median T_S across the ensemble; (2) the instance-wise difference ΔT_S , taking the median of this difference across the ensemble; and (3) the instance-wise *relative* difference $\Delta T_S/T_S$, taking the median of this difference across the ensemble. The “median of the difference” of the two latter metrics can be quite different from the “difference in median.” Since the magnitude of the T_S across our instances ranges over a few orders of magnitude, the instance-wise *relative* difference $\Delta T_S/T_S$ can be quite different from the instance-wise difference ΔT_S . While several of the pause schedules are better than the baseline according to every metric we use, others only do better in some of the metrics. The magnitude of the improvement, as well as the optimal pause location and duration, can vary significantly depending on the metric.

The left panel of Fig. 3 shows T_S for the ensemble in the above narrowed parameter range. Plotted as a horizontal line is the baseline (no-pause) case at its optimal $|J_F| = 1.6$. At both pause locations $s_p = 0.3$ and $s_p = 0.32$, a pause duration $t_p = 1$ μs is optimal on the ensemble of 45 instances. While at $s_p = 0.3$, only the $t_p = 1$ μs case beats the baseline, at $s_p = 0.32$, the T_S for all values of $t_p \in [0.25, 2]$ μs is consistently lower (better) than that of the baseline (for the corresponding p_{success} , see Fig. 14 in Appendix F).

The center panel in Fig. 3 shows the median instance-wise difference ΔT_S for the ensemble of 45 instances.

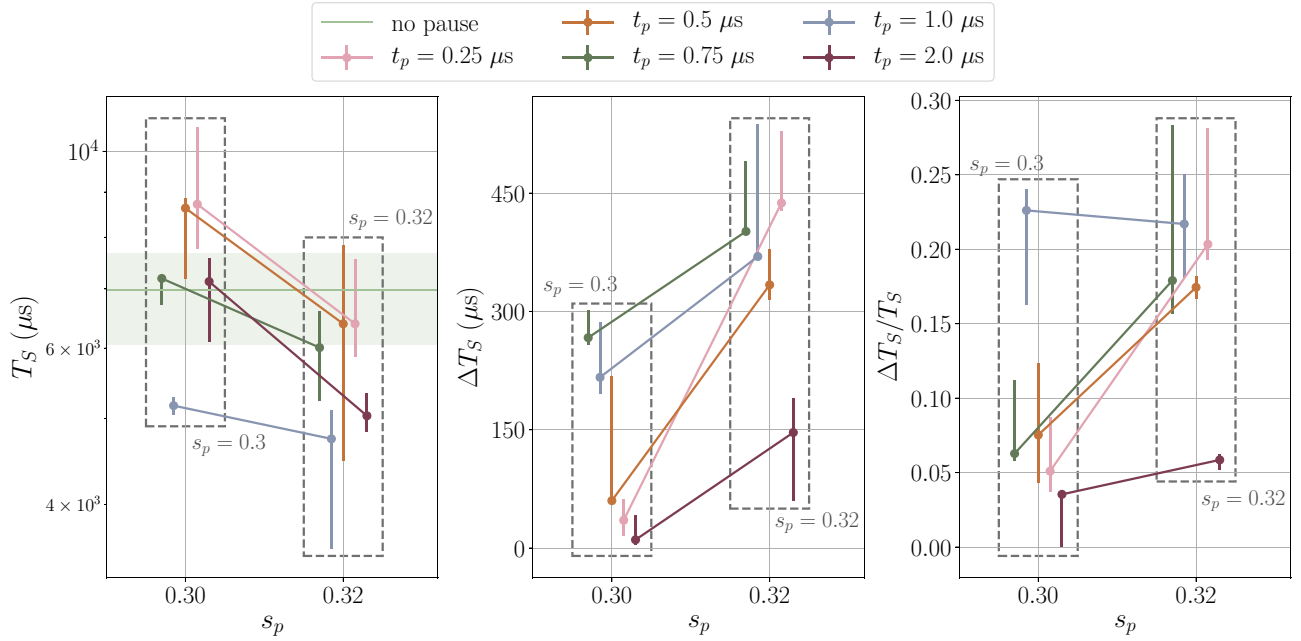


FIG. 3. The effect of the pause duration on T_S . Left: With pause durations of $\{0.25, 0.5, 0.75, 1, 2\}$ μs , and $|J_F|=1.8$, the median T_S for an ensemble of 45 instances is shown for pause locations $s_p = 0.3$ and $s_p = 0.32$. The reference (horizontal line and band for median and 35 and 65 percentiles, respectively) is the no-pausing case with parameters optimal for T_S : $t_a = 1$ μs and $|J_F| = 1.6$. The data points show the median, with error bars at the 35th and 65th percentiles, after performing 10^5 bootstraps over the set of instances. Center: The instance-wise absolute improvement in T_S (in μs). ΔT_S represents the reduction in T_S accomplished by introducing a short pause at an optimal location s_p . A positive ΔT_S indicates that the T_S is reduced (improved) by the introduction of the pause. ΔT_S is calculated by subtracting the T_S for the pause case with $|J_F| = 1.8$ from that of the no-pause case with $|J_F| = 1.6$ (the optimal $|J_F|$ for each case). The data points are the median and the error bars are 35th and 65th percentiles obtained from 10^5 bootstraps over 45 instances. Right: The instance-wise improvement ratio $\Delta T_S/T_S$. The data points are staggered along the s_p axis for readability. The error bars are chosen to showcase where most of the data lie. They represent the 35th and 65th percentiles of the bootstrap samples, instead of the median value of the 35th and 65th percentiles in the instance ensemble.

All the data points and their respective error bars are above zero, indicating that pausing provides a statistically significant improvement when the pause parameters are in the studied range with $s_p = \{0.3, 0.32\}$ and $t_p \in [0.25, 2]$ μs .

For example, while the median T_S of the ensemble is better for the baseline case than for the pause schedule with $s_p = 0.3$, $t_p = 0.25$, this pause schedule does better than the baseline on more than half of the 45 instances, leading to a positive ΔT_S . These two metrics provide different information about the strengths of each method.

The right panel of Fig. 3 represents the instance-wise relative improvement in T_S , that is, each instance-wise improvement is divided by the corresponding baseline no-pause T_S for that particular instance and then the median over the ensemble of instances of this set of values is calculated. We find that for a pause duration of $t_p = 1$ μs , the median relative improvement holds an optimal value of approximately 0.22. This pause duration is not the optimal for the absolute improvement shown in the middle panel, giving somewhat lower values of ΔT_S than a pause duration of 0.75 μs . This “change of order” occurs whenever

the following condition is met:

$$\frac{T_{S_j}(\text{base})}{T_{S_k}(\text{base})} > \frac{\Delta T_{S_j}(t_p, s_p)}{\Delta T_{S_k}(t'_p, s'_p)}, \quad (5)$$

where j is the instance where the median of $\Delta T_S/T_S(t_p, s_p)$ occurs and k is the instance where the median of $\Delta T_S/T_S(t'_p, s'_p)$ occurs. We examine in more detail the four best data points with respect to the $\Delta T_S/T_S$ metric, those with $s_p \in \{0.30, 0.32\}$ and $t_p \in \{0.75, 1.0\}$ μs .

We first look at the absolute improvement. At pause location $s_p = 0.3$, the median improvement for pause durations $t_p = 1$ and 0.75 μs are 216 and 266 μs , respectively. At $s_p = 0.32$, it is 369 and 401 μs , respectively, all the same order of magnitude. Consider the four instances that yield these four values. Their baseline no-pause T_S values vary considerably, being 897, 5003, 2738, and 25 581 μs , respectively. The substantially longer baseline T_S for the instances that are the median in each of the 1 μs cases than those in the 0.75 μs cases ($5\times$ at 0.3 and $10\times$ at 0.32) suggests that the 1- μs pause performs better than the 0.75 μs pause under the relative difference metric. (This

TABLE I. $s_p = 0.3$.

$s_p = 0.3$	Number of instances	Median $\Delta T_S/T_S$	35th percentile	65th percentile
$t_p = 1$				
Pause hurts	15	-0.6666	-0.77	-0.42
Pause helps	27	0.3960	0.33	0.59
$t_p = 0.75$				
Pause hurts	14	-1.19	-2.02	-0.61
Pause helps	28	0.2522	0.22	0.41

is not certain, because the median in the two metrics may correspond to different instances.)

We now look at the relative improvement. Compared to how it did with respect to the ΔT_S metric, the 0.75- μ s pause does much worse than expected relative to the other pause durations. For all four cases with parameters $s_p \in \{0.30, 0.32\}$ and $t_p \in \{0.75, 1.0\}$ μ s, the relative performance of many more instances improves more with a pause than is hurt by a pause (see Tables I and II). On the other hand, for pauses at $s_p = 0.3$, the median benefit over the instances for which a pause helps is less than the median amount of harm caused by a pause over the instances in which a pause hurts. This difference is much more pronounced for the $s_p = 0.30$ $t_p = 0.75$ case, with the median harm over 5 times that of the median benefit, compared to the other case, where the ratio is less than 2. At $s_p = 0.32$, the median benefit is larger than or the same as the median harm.

(In all cases, there are three instances that are not solved with or without a pause and hence are not included here.)

When interpreting these results, it is worth keeping in mind that with the exponential dependence of T_S on the probability of solution, long T_S values are subject to much greater statistical fluctuations than shorter ones.

C. Shift in optimal pause location with $|J_F|$

One interesting new avenue that opens up with the study of embedded problems is how the value of $|J_F|$ affects the benefits and effects of pausing. As previously discussed, the p_{success} versus s_p curve typically shows a peak around an optimal pause location and is mostly flat far away from it (as in Fig. 2). We also seen in Fig. 1 that without a pause,

TABLE II. $s_p = 0.32$.

$s_p = 0.32$	Number of instances	Median $\Delta T_S/T_S$	35th percentile	65th percentile
$t_p = 1$				
Pause hurts	12	-0.1993	-0.29	-0.14
Pause helps	30	0.3467	0.32	0.58
$t_p = 0.75$				
Pause hurts	13	-0.3602	-0.59	-0.21
Pause helps	29	0.3879	0.34	0.50

the value of $|J_F|$ affects p_{success} . For the pausing case, when $|J_F|$ increases, not only does the height of the peak change with $|J_F|$ but its position shifts as well, moving earlier in the anneal. The top panel of Fig. 4 shows this shift for a demonstration instance and a wide range of $|J_F|$, with the horizontal axis spanning the range of pause locations where the peak in p_{success} is found.

Such clear shifting is found in many instances and results in a shift in the behavior of the whole instance ensemble, as shown in the bottom panel of Fig. 4. For figure clarity, pausing results for just three values of $|J_F|$ are shown. The shift is consistent over all of the $|J_F|$ values we examine (in [1.2, 2]); see Fig. 12 in Appendix F for additional results.

The probability of success for smaller $|J_F|$ values, such as 1.2 and 1.3, even away from the peak, is clearly lower than for larger $|J_F|$ values (This holds true for the ensemble of instances, but some individual outliers are found, with a high p_{success} for smaller values of $|J_F|$. Figure 11 in the Appendix shows some examples.) The reason is that when the ferromagnetic coupling is not very strong compared to the problem couplings, it is more likely that the low-lying energy states are densely populated by states with an inconsistent vertex model (i.e., when not all the qubits are aligned and hence are no longer acting as a single variable). Accordingly, even when the annealer is doing well at finding the ground state or a low-lying state, such an outcome does not correspond to a valid solution of the original problem. Indeed, by applying simulated annealing to solve the embedded problem (which is too large to diagonalize exactly), we verify that in the range of $|J_F|$ that we use, the ratio of states with broken vertex models in the ground or low-lying states is significantly higher for $|J_F| = 1.2, 1.3$ than that for 1.4 and above.

The mechanism for why the optimal pause location typically shifts toward earlier in the anneal with $|J_F|$ fits our theoretical understanding, which is set out in Sec. B.

D. Help of partial gauges

We develop a *partial gauge transformation* technique that significantly improves the probability of success and enables the confirmation of the peak shift.

Gauge averaging is a technique commonly used to alleviate the effect that intrinsic biases on the local fields and couplers can have on the data obtained from a quantum annealer [24]. It can help improve statistics and lead to less noisy results and improved p_{success} and T_S . A gauge transformation starts with assigning a random sequence $a_j \in \{\pm 1\}$ to redefine the basis for each qubit, $\tilde{Z}_j = a_j Z_j$. If we adjust the local field and couplers accordingly such that

$$\tilde{J}_{ij} = a_i a_j J_{ij}$$

$$\tilde{h}_i = a_i h_i,$$

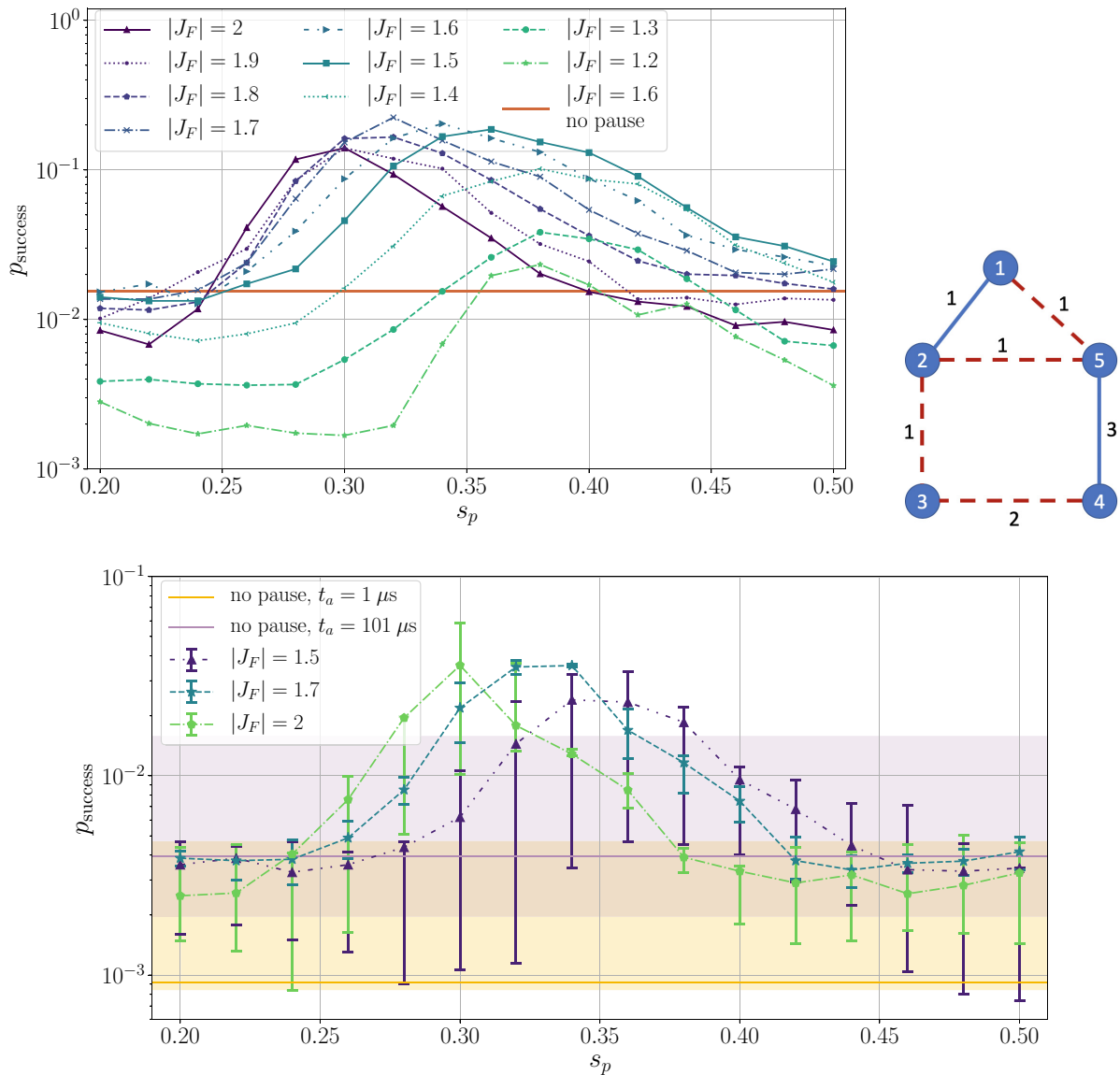


FIG. 4. The shift of the optimal pause location with $|J_F|$. Top left: The probability of success versus the annealing pause location for the demonstration instance. The anneal is performed with a $1\text{-}\mu\text{s}$ anneal time and a $100\text{-}\mu\text{s}$ pause. A monotonic shift in the peak location with $|J_F|$ is observed. The horizontal curve corresponds to no pause, $|J_F| = 1.6$, and an anneal time of $1 \mu\text{s}$. The reason for lower probability of success for $|J_F| = 1.2$ is detailed in Sec. C. Top right: The graph of the demonstration instance. The dashed red edges represent the minimum spanning tree of degree 2. Bottom: The probability of success for an ensemble of nine instances of $n = 5$ with a pause duration $t_p = 100 \mu\text{s}$ and $t_a = 1 \mu\text{s}$. The horizontal lines (for the median) and the bands (for the 35th to 65th percentiles) are baseline results with no pause, $|J_F| = 1.6$: Blue (lower) line or band, $t_a = 1 \mu\text{s}$; orange (lower) line or band, $t_a = 101 \mu\text{s}$.

then the resulting Hamiltonian has the same energy spectrum as the original one. This Hamiltonian is run on the annealer and the output bit string is transformed back using the same a_j 's. By performing multiple gauge transformations and averaging results over them, biases that stem from, for example, a qubit having a slight preference to aligning in one direction over the opposite can be suppressed.

When $J_{ij} \in [-1, 1]$, it is straightforward to apply gauges. For our embedded problems, however, we are making use of D-Wave's extended J range, allowing

$J_{ij} \in [-2, 1]$. The extended range discourages the breaking of vertex models during annealing due to the stronger ferromagnetic couplings between physical qubits representing the same logical variable, but it also impedes the use of standard gauges, since any couplings in the range $[-2, -1)$ cannot change sign.

Our partial gauge method circumvents this issue by only applying the gauge transformation on the couplings within the interval $[-1, 1]$. Because the extended range is exclusively used on the vertex models in our problems, the partial gauge on the embedded problem is equivalent to

applying a general gauge to the logical problem before embedding.

In a previous study [25], the boost in p_{success} by pausing has been observed for a family of embedded problems, but no relation between the optimal pausing location and J_F has been observed. In our study, with the help of partial gauge transformation, the variance in the annealing output is significantly suppressed, resulting in the revelation of the shift of the peak in Sec. C. This improvement of the variance is seen in the top panel (note the log scale) and by comparing the middle and bottom panels of Fig. 5.

Another benefit is a remarkable increase in p_{success} for hard problems. Usually, we do not expect p_{success} to change significantly from gauge averaging, because solving the problem without gauge transformations amounts to just applying one gauge, which is typically near average instead of being an outlier. But the probability of success is lower bounded by zero and when problems such as the ones we are solving here are difficult for the solver, the typical empirical p_{success} is zero or very close to zero. The existence of such a lower bound explains the significant benefit in applying gauges: even if we get a bad gauge, p_{success} cannot go below zero, while a good gauge can yield a much higher p_{success} . In a number of gauges, it is likely to encounter a few good gauges, bringing the average p_{success} up. The top panel of Fig. 5 shows, for a large ensemble of instances, that the improvement in the probability of success with 100 partial gauges is significant: about an order of magnitude higher than the results run without gauge transformation. The number of anneals remains unchanged regardless of the number of gauges used, with the total always being 50 000 anneals. For instance, with 100 gauges, 500 anneals are performed for each gauge. In this way, the use of gauges does not negatively impact the T_S .

The improvement in p_{success} saturates as one increases the number of gauges applied. Figure 13 in the Appendix shows, for an $n = 4$ ensemble, that applying as few as ten gauges yields similar p_{success} to 100 gauges. These results indicate that with ten gauges we are already likely to encounter one or more positive outliers, leading to the large improvement in p_{success} . As the number of gauges increases further, the effect is not as dramatic, indicating that the spread in gauge quality approaches the intrinsic distribution.

The partial gauge transformation therefore enables us to extend the benefits of general gauge averaging to embedded problems.

V. PHYSICAL PICTURE

In this section, we expand the physical picture of Ref. [7] to embedded problems, explaining both the shift of the optimal pausing location with increasing $|J_F|$ and why embedded problems, while benefiting significantly, benefit

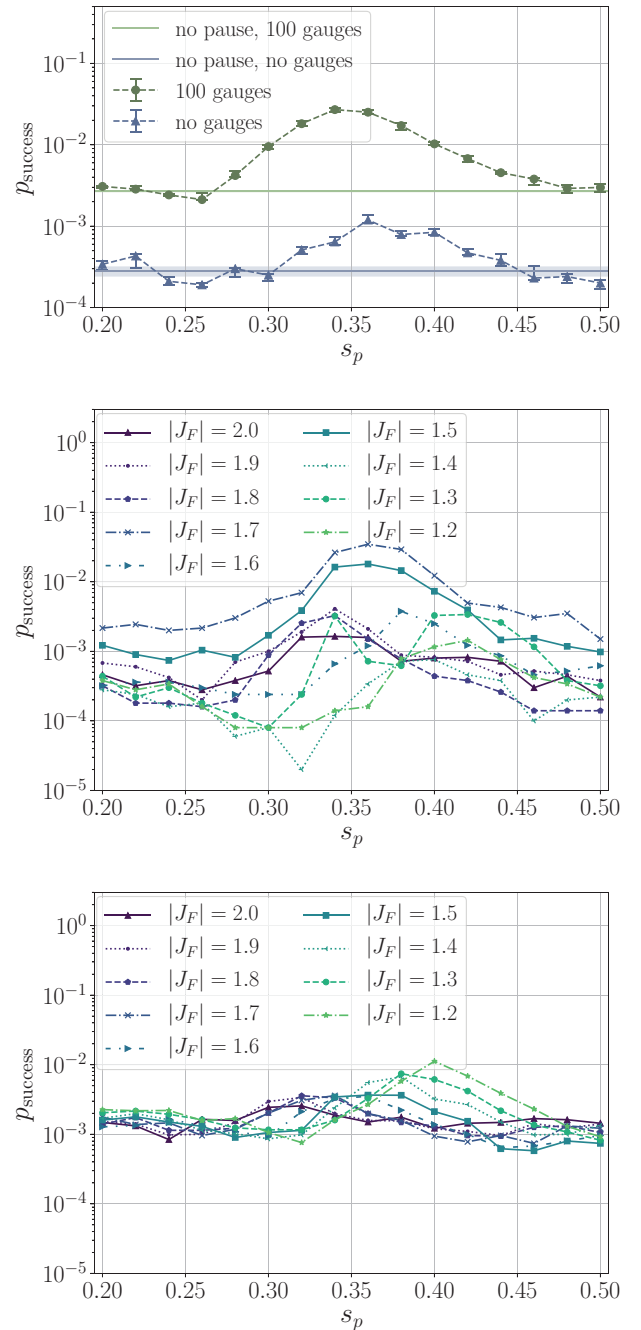


FIG. 5. The effect of partial gauges. Top: Partial gauges help to boost the average probability of success for an ensemble of instances (54 instances without gauges and 42 with gauges). $|J_F| = 1.6$ is used for all of the data shown. The baseline case without pausing is shown for reference: the median is shown as the horizontal lines and the 35th and 65th percentiles are shown as half-transparent bands. The blue (lower) band is for no gauges and orange (upper) band is for 100 gauges applied. Middle: The effect of $|J_F|$ on P_{succ} for a single instance. $t_a = 1 \mu\text{s}$ and a pause of duration $100 \mu\text{s}$ is applied. No gauge transformations are performed. Bottom: The same instance and parameters as in the middle panel, but with 100 partial gauges applied. The partial gauges help to suppress the variance and reveal the peak shift with $|J_F|$.

less from a pause than native problems. We provide a perturbation analysis supporting the picture and numerical evidence on the change in minimal gap location. The picture is far from that of the adiabatic regime. Pausing is effective after—not at—the minimum gap and diabatic and thermal effects play a significant role.

A. How pausing helps

We start with a recap of the physical picture of Ref. [7] that explains the increase in the probability of success by introducing a pause in the middle of the annealing schedule, after the minimal energy gap. Recent work [9] has verified this qualitative picture in numerical simulations and has also provided sufficient conditions under which pausing improves the probability of success. Loosely speaking, so long as shortly after the minimum gap the relaxation time scale is small enough (relative to the pause time), one can expect a pause to boost the probability of success. As discussed above, whether or not this improves T_S is not as obvious.

We use GS and FES to refer to the ground and subspace of the first excited states of the instantaneous quantum Hamiltonian. In the rest of the section, we refer to the *gap* as the energy gap between the GS and the FES.

At very early or late regions in the annealing, only one Hamiltonian—either the driver H_X or the problem H_C —dominates. Since both the problem Hamiltonian and the driving Hamiltonian are classical when acting alone, the dynamics in these regions are almost classical. Because the temperature T is much lower relative to the energy scale, excitations out of the GS are suppressed.

In the middle of the anneal, when the scales of H_X and H_C are comparable, the system dynamics are determined by the interplay of the energy gap, the nonadiabaticity (the annealing speed relative to the gap), and thermalization. In this region, we expect significant population loss from the ground state to excited states. In particular, when the gap is small enough, thermal-excitation channels are expected to become more dominant, populating excited states. This region is also where nonadiabatic transitions are expected to be largest.

We thus distinguish three different regimes in the anneal, as described below and illustrated by the schematic presented in Fig. 6.

Regime I: $\|A(s)H_X\| \gg \|B(s)H_C\|$. The instantaneous Hamiltonian is mainly H_X and its energy scale is much larger than the temperature, T . The system stays in the ground state of H_X .

Regime II: $\|A(s)H_X\| \sim \|B(s)H_C\|$ and their energy scale is comparable to the temperature. Both thermal and quantum dynamics happen and the minimal gap occurs in this region. As opposed to the zero-temperature case of adiabatic evolution, in which all of the population remains

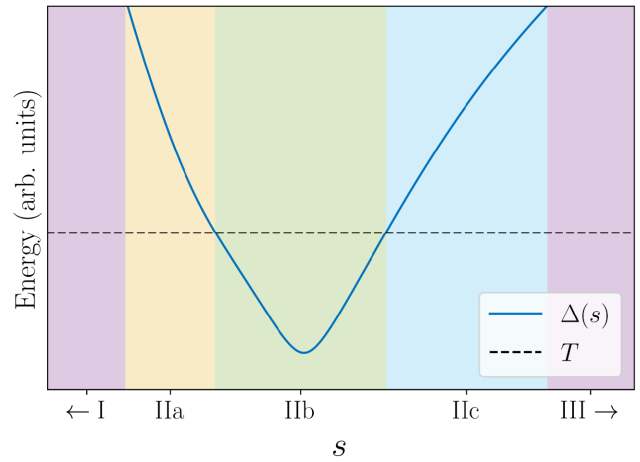


FIG. 6. A sketch of the three regimes. Colored in purple, the left and rightmost regions are the adiabatic regimes I and III (which extend further to the left and right as indicated by the arrows). Regime II is further subdivided into three regions, a, b, and c, as in the main text, which are determined by the instantaneous gap Δ and the temperature T . In region IIa, we expect the system to stop behaving strictly adiabatically and in region IIb approximate instantaneous thermalization may occur if the relaxation time scale is small enough, as well as nonadiabatic transitions. In region IIc, a pause may help to repopulate the GS. This should be thought of as an approximate picture of what occurs, to aid the reader. In reality, the transitions between these regions will, of course, not be sharp and defined by a single point during the evolution.

in the GS, thermal excitations and nonadiabatic transitions can both occur and it is difficult to distinguish between the two.

As the anneal goes on in this regime, it sequentially goes through the following regions:

(a) Gap approaching the temperature, system leaving the adiabatic regime, but transitions (nonadiabatic and thermal) may still be relatively slow compared to the system evolution.

(b) Gap near its minimum and much smaller than the temperature—thermalization effects play a dominant role and cause population loss from the GS. If a long enough pause is inserted, the system could approach its thermal equilibrium. Quantum nonadiabatic effects could be strong enough to increase the population of the FES beyond its magnitude at thermal equilibrium.

(c) Gap larger than the temperature, nonadiabaticity is weak. The system may still approximately equilibrate if given enough time (e.g., a pause) but will likely be far from the instantaneous Gibbs state during the standard anneal.

As the system enters region II c from II b, pausing can bring significant FES population back to the GS, since relative to the gap, the temperature T is now lower, hence boosting the probability of success.

Regime III: $\|B(s)H_C\| \gg \|A(s)H_X\|, \|B(s)H_C\| \gg T$, dynamics are slow, the system simply picks up phases under H_C , and the population distribution is final. This is also known as the frozen region in the literature.

B. How $|J_F|$ shifts the optimal pausing location earlier

An increase in $|J_F|$ is expected to shift the minimal gap to earlier in the anneal, meaning that region II b occurs earlier in the anneal, and therefore also shifts the optimal pause region II c earlier. This shift of the minimal gap can partly result from the increase in the relative norm of H_C , i.e., decreasing the value of $[A(s)/B(s)](\|H_X\|/\|H_C\|)$. Similar to Ref. [13], this is akin to shifting each point earlier in the anneal. In the case in which the $|J_F|$ are dominant, the result of Ref. [13] can be applied in a straightforward manner and one expects (i) the gap location to shift earlier in the anneal and (ii) the minimum gap size to increase. In our case, however, the $|J_F|$ are not generally dominant, in which case the argument of Ref. [13] cannot be applied directly. Instead, we approach it using perturbation theory. As we show below, while we do generally expect the location of the gap to shift earlier in the anneal, the size of the minimum gap may decrease with $|J_F|$ when taking the problem couplings J_{ij} into account.

In Fig. 7, for a small Ising problem embedded using an additional physical qubit on chimera connectivity—which allows exact diagonalization of the instantaneous Hamiltonian—we show the change of minimal gap with $|J_F|$.

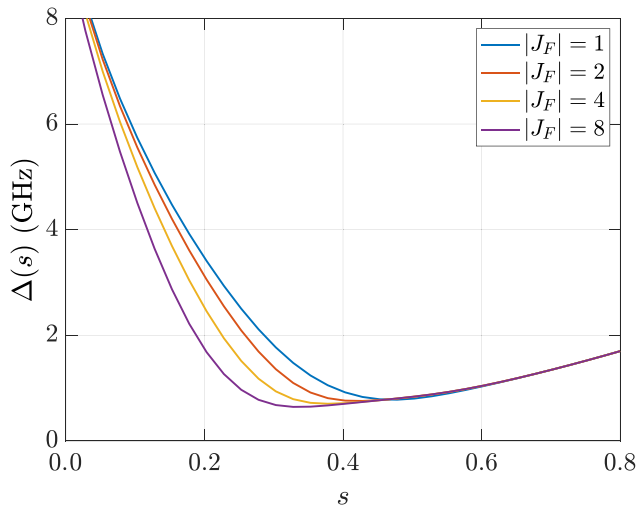


FIG. 7. The shift of the minimal gap with $|J_F|$. The energy gap between the ground and the first excited states for the instantaneous quantum Hamiltonian during annealing for a toy problem. The logical problem is a disordered Ising problem with local fields ($J_{ij}, h_i \in [-1, 1]$) of a complete graph of size 3 (triangle), embedded to four physical qubits (square) on chimera connectivity. The gap is computed exactly by diagonalizing the instantaneous Hamiltonian. As $|J_F|$ increases, the instantaneous gap closes and the minimum gap shifts to earlier in the anneal.

Because the cells in chimera are bipartite graphs, odd cycles are not native to the structure. In this small example, a fully connected triangle graph on three nodes requires minor embedding as a square with four nodes. Below, we provide an argument as to why the minimum gap increases in value with decreasing ferromagnetic strength.

Before providing a proof sketch for the gap increase, we mention another picture that comes into play, which is that an increase in $|J_F|$ can yield “clusters” (physical qubits representing the same logical qubits) with stronger internal couplings. Changing the state in such clusters requires collective flipping of qubits, demanding greater quantum dynamics. Accordingly, the transition from region II b to region II c would happen earlier in the anneal. Such a picture may also be accountable for the less dramatic increase in the success rate compared to the native Ising case: the associated energy barrier may require a much higher relative temperature—while pausing earlier helps, the amount by which it can help is limited (because it is an interplay of the three influences, which are correlated in a given annealing schedule and at a given temperature).

1. Proof sketch of gap scaling under $|J_F|$

We apply first-order nondegenerate perturbation theory. Let $H(s) = H_0(s) + B(s)\lambda H_F$ with $H_0(s) = A(s)H_X + B(s)H_C + B(s)J_F H_F$, where $\lambda > 0$, $J_F < 0$, and H_F is the ferromagnetic Hamiltonian for the vertex model. That is, we are considering the effect of weakening the vertex model infinitesimally by decreasing $|J_F|$. To simplify matters, assume that the only vertex model is a chain of length 2. Then, $H_F = \sigma_{k_1}^z \sigma_{k_2}^z$ for two qubits k_1, k_2 . Write $|E_i(s)\rangle$ as the instantaneous i th eigenstate of $H_0(s)$. For simplicity, we drop the explicit s dependence (i.e., we just consider s fixed at some value). Then we can always decompose our instantaneous eigenstates in the computational basis $|E_i\rangle = \sum_j a_j^{(i)} |z_j^L\rangle + \sum_k b_k^{(i)} |z_k^B\rangle$, where the $|z_j^L\rangle$ are logical states and $|z_k^B\rangle$ has the chain broken. We compute the matrix elements as follows:

$$\langle E_i | H_F | E_i \rangle = \sum_j |a_j^{(i)}|^2 - \sum_k |b_k^{(i)}|^2. \quad (6)$$

Note that, by normalization, $\sum_k |b_k^{(i)}|^2 = 1 - \sum_j |a_j^{(i)}|^2$ and, denoting the logical probability $P_L^{(i)} := \sum_j |a_j^{(i)}|^2$,

$$\langle E_i | H_F | E_i \rangle = 2P_L^{(i)} - 1. \quad (7)$$

This tells us, to first order in $\lambda > 0$, that the low-lying energy levels experience an increase in energy upon decreasing the ferromagnetic strength ($|J_F| \rightarrow |J_F| - \lambda$), i.e., $E'_i = E_i + B\lambda(2P_L^{(i)} - 1) > E_i$, assuming that $P_L^{(i)} > 1/2$. We see consistent behavior with this picture in Fig. 8 (even though this figure is not in the perturbative limit).

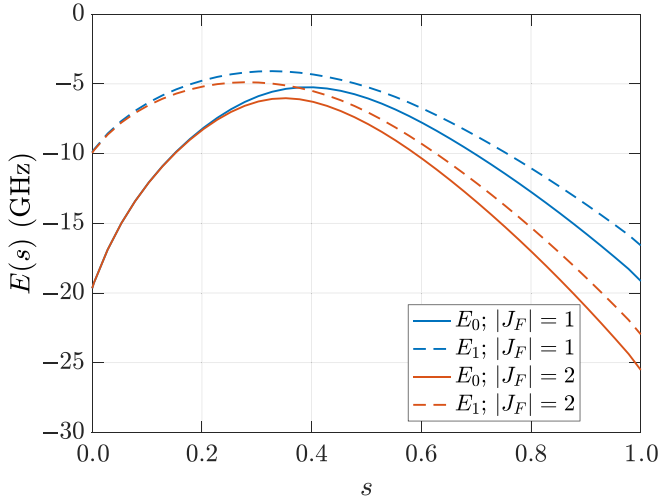


FIG. 8. The energy-level shift with $|J_F|$. The individual energy levels show an increase in energy upon decreasing the ferromagnetic strength (i.e., the case in which $\lambda > 0$ in our perturbation theory) for the problem in Fig. 7.

Now, the gap $\Delta = E_1 - E_0$ changes under λ , to first order, as

$$\begin{aligned} \Delta' &= \Delta + B\lambda(\langle E_1|H_F|E_1\rangle - \langle E_0|H_F|E_0\rangle) \\ &= \Delta + 2B\lambda(P_L^{(1)} - P_L^{(0)}), \end{aligned} \quad (8)$$

which therefore increases in magnitude (at a fixed s) by weakening the ferromagnetic couplings, assuming that $P_L^{(1)} > P_L^{(0)}$.

At the start of the anneal, $|E_0(0)\rangle = |+\rangle^{\otimes N}$ and so $P_L^{(0)} = 1/2$ (in the specific case when the embedding contains just one additional qubit). At $s = 0$, the FESs are linear combinations containing one excitation in the x eigenbasis, i.e., a single $|-\rangle$. Consider the symmetric FES, denoted $|FE_+\rangle$, where the state of the chain is $\frac{1}{\sqrt{2}}(|-+\rangle + |+-\rangle)$ (and the other qubits are all $|+\rangle$). This state is entirely in the logical subspace, due to the canceling out of the $|01\rangle$ and $|10\rangle$ terms. When the transverse field is “strong,” i.e., “near” to $s = 0$ (but where the FES degeneracy is broken), by the perturbation theory we may indeed therefore expect that $\Delta' > \Delta$. We see this in Fig. 7, where the strongest chain, $|J_F| = 8$, has the smallest instantaneous gap. In the case of an arbitrary chain length, following the general expression given in the first line of Eq. (8), a similar argument applies provided that $P_L^{(1)}$ is large enough relative to $P_L^{(0)}$, though the precise dependence is more complicated.

We also know that once the transverse field becomes weak relative to the problem Hamiltonian (e.g., $A/B \lesssim 1$), $P_L^{(1)} - P_L^{(0)} \rightarrow 0$, as both the instantaneous GS and the FES become close to logical states.

By interpolating between the two extremes ($s \approx 0, s \rightarrow 1$), the above argument explains the change in gap size

observed in Fig. 7 and, moreover, why the location of the minimum gap is expected to move earlier in the anneal.

VI. CONCLUSIONS AND FUTURE WORK

We study how midanneal pauses affect performance on embedded problems using the class of degree-bounded minimum-spanning-tree problems. We develop a partial gauge approach that allows us to take advantage of the extended J range while also using gauges (partially), yielding significantly cleaner results and improved performance than without partial gauges, enabling us to confirm the theoretical predictions. Our results confirm that, as for native problems, there is a region, consistent across instances, in which a pause improves the probability of success. We further show that the pause generally improves the time to solution (T_S) for these problems and we evaluate the performance on three T_S -related metrics. We extend the theoretical picture of Ref. [7] to embedded problems, describing the interaction of embedding parameters with annealing parameters, thermalization, and nonadiabatic effects. This picture explains why the optimal pause location moves earlier in the anneal as $|J_F|$ increases and why the benefit provided by pausing, while significant, is not as great as for native problems. It generally provides both deeper insights into the physics of these devices and pragmatic recommendations to improve performance on optimization and sampling problems.

This study suggests a number of avenues for future research. As the connectivity of quantum annealing hardware increases, as is anticipated in D-Wave’s upcoming Pegasus architecture, a lower embedding overhead should translate into greater benefit from pausing. Larger and more connective devices will allow larger problem sizes to be run, enabling scaling analyses. As annealing hardware becomes more flexible, a wider variety of advanced schedules become possible, such as a smooth slowing down rather than a pause or annealing at different rates in different parts of the system depending on the local embedding characteristics or the local problem-instance structure. All of these possibilities should be explored on a variety of optimization problems as well as on the BD-MST problem class investigated here. Embedding affects sampling problems even more than optimization problems [26], so a study of the interplay between embedding parameters and annealing parameters should be done in that context as well. Experiments at other temperatures and with the ability to do quick quenches at arbitrary points in the anneal would give further insight into the underlying physics. Further, given that diabatic behavior is expected to be useful even for devices that could remain adiabatic throughout a run, an intriguing area for both theoretical research and hardware development is the use of engineered dissipation to support cooling in conjunction

with diabatic evolution, enabling much more controlled utilization of thermalization in quantum annealers in the future.

ACKNOWLEDGMENTS

We are grateful for support from the NASA Ames Research Center, particularly the NASA Transformative Aeronautic Concepts Program. We also appreciate support from the Air Force Research Laboratory (AFRL) Information Directorate under Grant No. F4HBKC4162G001 and the Office of the Director of National Intelligence (ODNI) and the Intelligence Advanced Research Projects Activity (IARPA), via IAA 145483. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, the IARPA, the AFRL, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. Z.G.I. was also supported by the Universities Space Research Association (USRA) Feynman Quantum Academy, funded by the NAMS R&D Student Program at the NASA Ames Research Center. Z.G.I., S.H., J.M., and Z.W. are also supported by NASA Academic Mission Services, Contract No. NNA16BD14C. We acknowledge insightful discussions with Davide Venturelli and also with Riccardo Mengoni, particularly preliminary work mapping spanning-tree problems to QUBO [19].

**APPENDIX A: PROBLEM MAPPING:
“LEVEL-BASED”**

Consider a graph $G = (V, E)$ with weights $w(E)$ for each edge, from which we wish to obtain a minimum-weight spanning tree with maximum degree Δ , i.e., find its BD-MST. This involves minimizing the sum of the weights of the tree edges, represented by the cost function

$$C_0 = \sum_{p,v} w_{pv} x_{p,v}, \tag{A1}$$

which we explain below. Several constraints are also imposed to ensure that the graph is in fact a spanning tree and its degree is bounded by Δ .

A root for the tree is picked randomly or based on problem structure—generally, picking a high-degree vertex as the root will result in lower resource costs—and assigned to level 1. Its children will be at level 2, their children at level 3, and so on, leading to the “level-based” designation.

The variables $x_{p,v}$ appearing in Eq. (A1) represent the parent-child relationships in the tree; $x_{p,v} = 1$ if p is the (adjacent) parent of v (and 0 if not). The indices p, v range over $p = 1, \dots, n$ and $v = 2, \dots, n$, restricted to (intersected with) pairs (p, v) or (v, p) that occur in E . Thus there are two variables for every edge not containing the root and one for every root edge, giving $2m - d_r$ total $x_{p,v}$ variables, with m being the number of edges in E and d_r the degree of the root.

Since our problem needs to be in QUBO form, the constraints are expressed as penalty terms. The first penalty term enforces that every node (except the root) has exactly

TABLE III. $n = 5$ graphs.

m	Label	Graph name	Edges
4	m4ver1	DhC	(1,2), (2,3), (3,4), (4,5)
5	m5ver1	Dhc	(1,2), (2,3), (3,4), (4,5), (1,5)
5	m5ver2	DiK	(1,2), (2,3), (2,5), (3,4), (4,5)
5	m5ver3	DjC	(1,2), (2,3), (2,4), (3,4), (4,5)
5	m5ver5	DiS	(1, 2), (1, 3), (1,4),(1,5),(4,5)
5	m5ver6	DKs	(1, 2), (2, 3), (3, 4), (4, 5), (3, 5)
6	m6ver1	DyK	(1,2), (1,5), (2,5), (2,3), (3,4), (4,5)
6	m6ver2	DjS	(1,2), (2,3), (2,4), (2,5), (3,4), (4,5)
6	m6ver3	DjK	(1,2), (2,3), (2,5), (3,5), (3,4), (4,5)
6	m6ver4	D{K	(1,2), (1,5), (1,3), (2,3), (3,4), (4,5)
6	m6ver5	D{c	(1, 2), (1, 3), (2, 3), (3, 5), (3, 4), (4, 5)
6	m6ver6	D]o	(1, 2), (2, 3), (3, 4), (1, 4), (2, 5), (4, 5)
7	m7ver1	D]S	(1,2), (1,5), (1,4), (2,5), (2,3), (3,4), (4,5)
7	m7ver2	DzW	(1,2), (1,5), (2,5), (2,3), (2,4), (3,5), (4,5)
7	m7ver3	D]c	(1,2), (1,3), (1,4), (1,5), (2,3), (3,4), (4,5)
7	m7ver4	D ~ C	(1,2), (1,3), (1,4), (2,4), (2,3), (3,4), (4,5)
7	m7ver5	D]w	(1, 2), (2, 3), (3, 4), (1, 4), (4, 5), (2, 5), (3, 5)
8	m8ver1	D]k	(1,2), (1,5), (1,3), (1,4), (2,5), (2,3), (3,4), (4,5)
8	m8ver2	Dz[(1,2), (1,5), (2,5), (2,3), (2,4), (3,4), (3,5), (4,5)
9	m9ver1	D ~ k	(1, 2), (2, 3), (4, 5), (1, 5), (1, 4), (1, 3), (2, 5), (2, 4), (3, 5)
10	m10ver1	D ~ {	(1, 2), (2, 3), (3, 4), (4, 5), (1, 5), (1, 4), (1, 3), (2, 5), (2, 4), (3, 5)

one parent:

$$C_{\text{pen1}} = \sum_{v \in \{2, \dots, n\}} \left(\sum_{p: (pv) \in E} x_{p,v} - 1 \right)^2. \quad (\text{A2})$$

The number of terms in the sum is $2m - d_r$, i.e., equal to the number of variables $x_{p,v}$.

The second penalty term enforces that each vertex exists at exactly one level in the tree:

$$C_{\text{pen2}} = \sum_{v \in \{2, \dots, n\}} \left(\sum_{\ell=2}^n y_{v,\ell} - 1 \right)^2. \quad (\text{A3})$$

It introduces the $y_{v,\ell}$ variables, with $y_{v,\ell} = 1$ if v is at depth ℓ of the tree, $v = 2, \dots, n$, $\ell = 2, \dots, n$. There are $(n-1)^2$ such variables. However, since the number of variables will eventually determine how many logical qubits the problem requires, it is in our interest to reduce it as much as possible. By picking the root smartly, the range of ℓ can be reduced. We also carry out the following preprocessing: taking the original graph $G = (V, E)$, the distance from each node to the one we have selected as the tree root is calculated. Given that it is impossible for a node to be at a level smaller than its distance to the root, we can avoid generating any $y_{v,\ell}$ for which that is the case, further bringing down the total number of $y_{v,\ell}$ variables.

The third penalty term enforces that the tree has degree at most Δ :

$$C_{\text{pen3}} = \sum_{p=2}^v \left(\sum_{v: (pv) \in E} x_{p,v} - \sum_{j=1}^{\Delta-1} z_{p,j} \right)^2 + \left(\sum_{v: (1v) \in E} x_{1,v} - \sum_{j=1}^{\Delta} z_{1,j} \right)^2. \quad (\text{A4})$$

It is separated into two terms to account for the fact that the root can have up to Δ children, while all other nodes cannot have more than $(\Delta-1)$, since they have a parent. To enforce the inequality $\sum_{v: (pv) \in E} x_{p,v} \leq \Delta-1$, integer variable $z_p \in [0, \Delta-1]$ is introduced as a slack variable and the inequality is enforced as equality $\sum_{v: (pv) \in E} x_{p,v} = z_p$. The integer variable is further encoded into binary variables $z_{p,j}$. In general, various encoding methods can be applied to encode an integer into binaries, including binary, unary, and one-hot encodings. While binary encoding is most efficient for integers of value power of 2, we use unary encoding here, which can be applied straightforwardly to arbitrary values of Δ .

The fourth and final penalty term enforces that the tree encoding is consistent, i.e., that if p is the parent of v , then

its level is one less than v 's:

$$C_{\text{pen4}} = \sum_{p,v} \sum_{\ell=3}^n x_{p,v} y_{v,\ell} (1 - y_{p,\ell-1}) + \sum_{v=2}^{d_r} x_{1,v} (1 - y_{v,2}) + \sum_{v=2}^{d_r} y_{v,2} (1 - x_{1,v}), \quad (\text{A5})$$

where the last two sums handle the edges connected to the root and their terms are quadratic, while the first sum deals with the remaining edges and produces cubic terms of the form $x_{p,v} y_{v,\ell} (1 - y_{p,\ell-1})$. While the original number of cubic terms would be

$$(2m - 2d_r) * (n - 2),$$

due to the preprocessing of the $y_{v,\ell}$ variables this number is reduced. Because cubic terms cannot be directly encoded in D-Wave, we introduce an ancilla variable $a_{p,v,\ell}$ to encode $x_{p,v} y_{v,\ell}$ and, accordingly, a penalty function $f(x, y, a) = 3a + xy - 2ax - 2ay$ is added to raise a penalty if $a = xy$ is violated. The term $x_{p,v} y_{v,\ell} (1 - y_{p,\ell-1})$

TABLE IV. Graph weights are uniformly drawn from the above lists.

Label	Weight list
w2	[1, 2, 1, 2, 1, 2, 1, 2, 1, 2]
w3	[1, 1, 2, 1, 1, 2, 1, 1, 2, 1]
w4	[1, 1, 2, 2, 1, 1, 2, 2, 1, 1]
w5	[1, 4, 1, 4, 1, 4, 1, 4, 1, 4]
w6	[1, 3, 6, 1, 3, 6, 1, 3, 6, 1]
w7	[1, 7, 1, 7, 1, 7, 1, 7, 1, 7]
w8	[3, 2, 1, 3, 2, 1, 3, 2, 1, 3]
w9	[4, 3, 2, 1, 4, 3, 2, 1, 4, 3]
w10	[5, 4, 3, 2, 1, 5, 4, 3, 2, 1]
w11	[6, 5, 4, 3, 2, 1, 6, 5, 4, 3]
w12	[7, 6, 5, 4, 3, 2, 1, 7, 6, 5]
w13	[1, 1, 3, 4, 2, 1, 2, 3, 4, 2]
w14	[3, 2, 1, 1, 1, 1, 2, 4, 2, 2]
w15	[2, 1, 2, 1, 4, 1, 1, 3, 3, 2]
w16	[4, 3, 3, 4, 3, 3, 4, 3, 4]
w17	[3, 4, 7, 5, 5, 5, 5]
w18	[2, 1, 4, 1, 2, 1, 2]
w19	[4, 6, 4, 7, 4, 7]
w20	[1, 1, 2, 3, 2, 3]
w21	[4, 5, 4, 5, 5]
w22	[2, 2, 6, 2, 4]
w23	[3, 3, 5, 2, 3, 2, 5, 2, 5]
w24	[4, 3, 2, 2]
w25	[2, 2, 6, 2, 4]
w26	[4, 3, 3, 3]
w27	[3, 4, 7, 5, 5, 5, 5]
w28	[4, 6, 4, 7, 4, 7]
w29	[6, 4, 2, 2]

can then be replaced by quadratic terms:

$$4a - ay_{p,\ell-1} + x_{p,v}y_{v,\ell} - 2ax_{p,v} - 2ay_{v,\ell}. \quad (\text{A6})$$

The total number of variables (and hence, logical qubits) without preprocessing is at most

$$2m - d_r + (n - 1)^2 + n(\Delta - 1) + 1 + (2m - 2d_r)(n - 2) \simeq 2mn + n^2.$$

This would mean, for instance, that the complete graph K_5 with $\Delta = 3$ would require between 86 and 100 logical qubits (depending on d_r). With preprocessing, we are able to bring this number down to 74.

Finally, we can write the overall objective function as

$$C = C_0 + A(C_{\text{pen1}} + C_{\text{pen2}} + C_{\text{pen3}} + C_{\text{pen4}}), \quad (\text{A7})$$

and, accordingly, the cost Hamiltonian H_C . In Eq. (A7), we have defined the minimum penalty weight to be the maximum edge weight:

$$A = w_{\text{max}} = \max_{(uv) \in E} w_{uv}. \quad (\text{A8})$$

In Ref. [19], we provide proof that setting $A = w_{\text{max}} + \epsilon$ with any positive ϵ suffices to guarantee C is minimized by bounded-degree spanning trees that are optimal for C_0

TABLE V. The mapped problem size for $N = 4-6$ using the D-Wave chimera architecture and problem size $N = 4-10$ using the future D-Wave Pegasus architecture. For the Pegasus-architecture entries, embedding is only performed for the complete graphs, which is the reason for the large number of unset entries in the last three columns. For the chimera-architecture-embedding entries, we are unable to embed graphs with $n \geq 7$ using the default embedding parameters, which is the reason for the missing data entries in the last four rows of the table in columns 3–5. Lastly, we are not collecting median-vertex-model-size statistics for some of the early $n = 4$ network communication graphs that we examine early in the study, which is the reason for the missing chimera-architecture entries for the $n = 4$ graphs near the top of the table.

n	m	Chimera architecture			Pegasus architecture		
		Number of logical variables	Number of physical variables	Median vertex model size	Number of logical variables	Number of physical variables	Median vertex model size
4	6	35	108–150	3–4	35	54–71	1–2
4	5	29	65–121
4	4	25	60–116
4	4	23	47–82
4	3	20	40–76
5	4	32	83–140	1.5–3
5	5	42	121–215	2–4
5	5	43	138–169	2–3
5	5	44	149–205	2–4
5	5	47	151–220	2–4
5	5	39	112–179	1.5–3
5	6	50	169–205	2–4
5	6	53	194–255	2–4
5	6	49	181–246	2–4
5	6	46	148–272	2.5–4.5
5	6	50	170–249	3–5
5	6	50	164–217	2.5–4.5
5	7	54	193–247	3–5
5	7	58	229–300	3–4.5
5	7	50	171–260	3–5
5	7	55	226–281	3–5
5	7	56	227–273	2–5
5	7	50	162–224	2.5–5
5	8	58	219–284	3–5
5	8	64	287–362	3–5
5	9	66	299–413	3.5–6
5	10	74	380–485	4–7	74	164–233	2–3
6	15	137	1166–1293	4–6	137	477–544	3
7	21	230	1018–1292	2–4
8	28	359	2046–2712	2–4
9	36	530	3744–4464	2–4
10	45	749	6024–7889	2–4

and correctly encoded. In our runs, for convenience, we set $\epsilon = 0$, which can, in principle, lead to an invalid bit string also minimizing C . The solutions returned from the quantum annealer are each checked for optimality and correct encoding. Though increasing A by any amount would guarantee that the optimal cost of a solution implies its correct encoding, in practice we observe that this is still the case despite having set $\epsilon = 0$. We provide details and further discussion in Ref. [19].

APPENDIX B: APPROXIMATION COMPLEXITY FOR BD-MST PROBLEMS

To find a degree-bounded spanning tree of cost at most r times the optimum remains NP hard for any $r \geq 1$ [27]. Hence, approximation algorithms are often designed to return a low-weight spanning tree with the vertex degree bound Δ slightly relaxed. In Ref. [28], a polynomial time algorithm is given for the unweighted problem that returns a spanning tree of degree at most $\Delta^* + 1$, where Δ^* is the minimal Δ for which such a spanning tree exists. For the weighted case, Ref. [29] shows a polynomial time algorithm that returns a spanning tree with vertex degree at most $\Delta + 2$ —subsequently improved to $\Delta + 1$ in Ref. [30]—and cost at most OPT, where OPT is the optimal spanning-tree weight under the desired bound Δ . Alternatively, heuristics exist that return valid Δ -bounded spanning trees but with suboptimal cost that may be difficult to quantify generally. A wide variety of approaches

have been developed for this problem [31–36], including specific approximations for various special cases (e.g., geometric weights); for an overview, see Ref. [37].

APPENDIX C: BD-MST PROBLEM INSTANCES

All connected graphs of $n = 5$, with $m = |E|$ ranging from 4 to 10, are considered where an BD-MST with $\Delta = 2$ exists. The edges in these graphs are provided in Table III. Additionally, the graph labeled “m5ver5” is included to demonstrate the BD-MST with $\Delta \geq 3$. For each graph, problem instances are generated by assigning a set of weights by sampling from one of the lists of weights appearing in Table IV. The first m weights in each weight list are used to define an instance.

APPENDIX D: EMBEDDING STATISTICS

Table V contains the mapped problem size for each graph and embedding features such as the number of physical qubits, the size of the vertex model, etc. Embedding statistics on a future D-Wave architecture (Pegasus) are also included in this table. For the Pegasus architecture, each qubit can couple to 15 other qubits, as opposed to the chimera architecture, which allows each qubit to connect to at most six additional qubits.

As discussed in Sec. B, Fig. 9 contains detailed information about the typical size of the embedded problems for different graphs, including the number of physical qubits

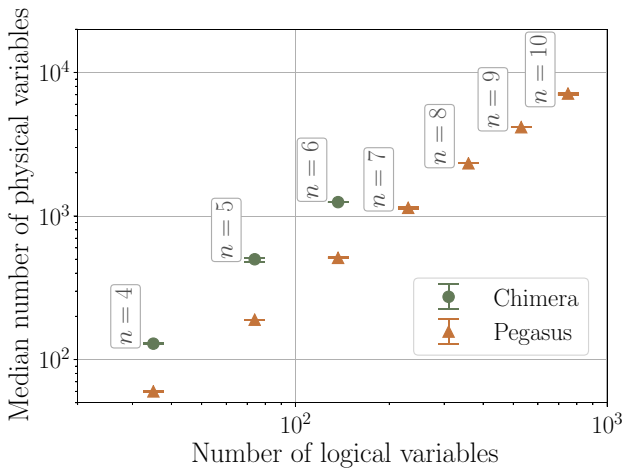


FIG. 9. An embedding comparison between current and future architectures. Embedding for the complete graphs for problem size $n = 4$ –10 with default embedding parameters and 10–20 instances drawn for each graph. Chimera embedding performed with D-Wave’s SAPI2 FIND_EMBEDDING routine with the D-Wave 2000Q hardware adjacency graph. Pegasus embedding performed with the Ocean MINORMINER FIND_EMBEDDING routine. The median number of physical qubits as a function of the number of logical qubits with error bars are at the 35th and 65th percentiles after bootstrapping over the ensemble of instances.

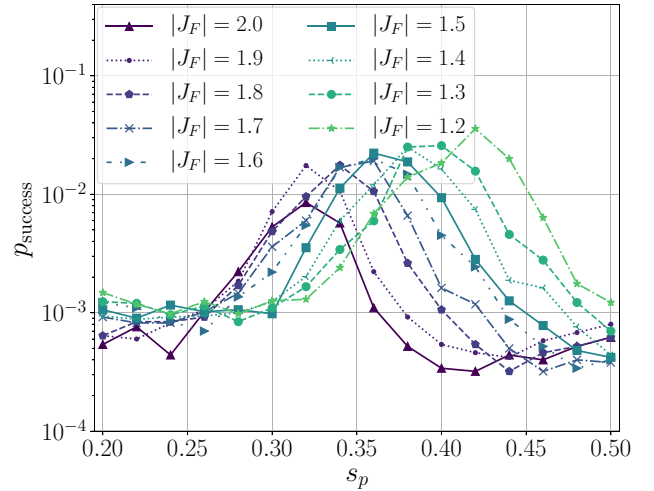


FIG. 10. The shift of the optimal pause location for an instance with $\Delta = 3$ probability of success versus the annealing pause location for the $n = 5$, “m5ver5” [$m = 5$, $K_{1,4} + e$, g_6 : DiS], 14-instance weight set using embedding number 20 with $\Delta = 3$, 1 μ s anneal, 100- μ s pause, 50 000 reads and zero partial gauges. Pause location ranging from 0.2 to 0.5 and J_{ferro} varied from -1.2 to -2.0 . The peak in p_{success} shifts from $s_p = 0.42$ at $J_{\text{ferro}} = -1.2$; to $s_p = 0.36$ for $J_{\text{ferro}} = -1.5$; and $s_p = 0.32$ for $J_{\text{ferro}} = -2.0$. Note that $\Delta = 3$ is the minimum delta that can be used to obtain a minimum-weight spanning tree.

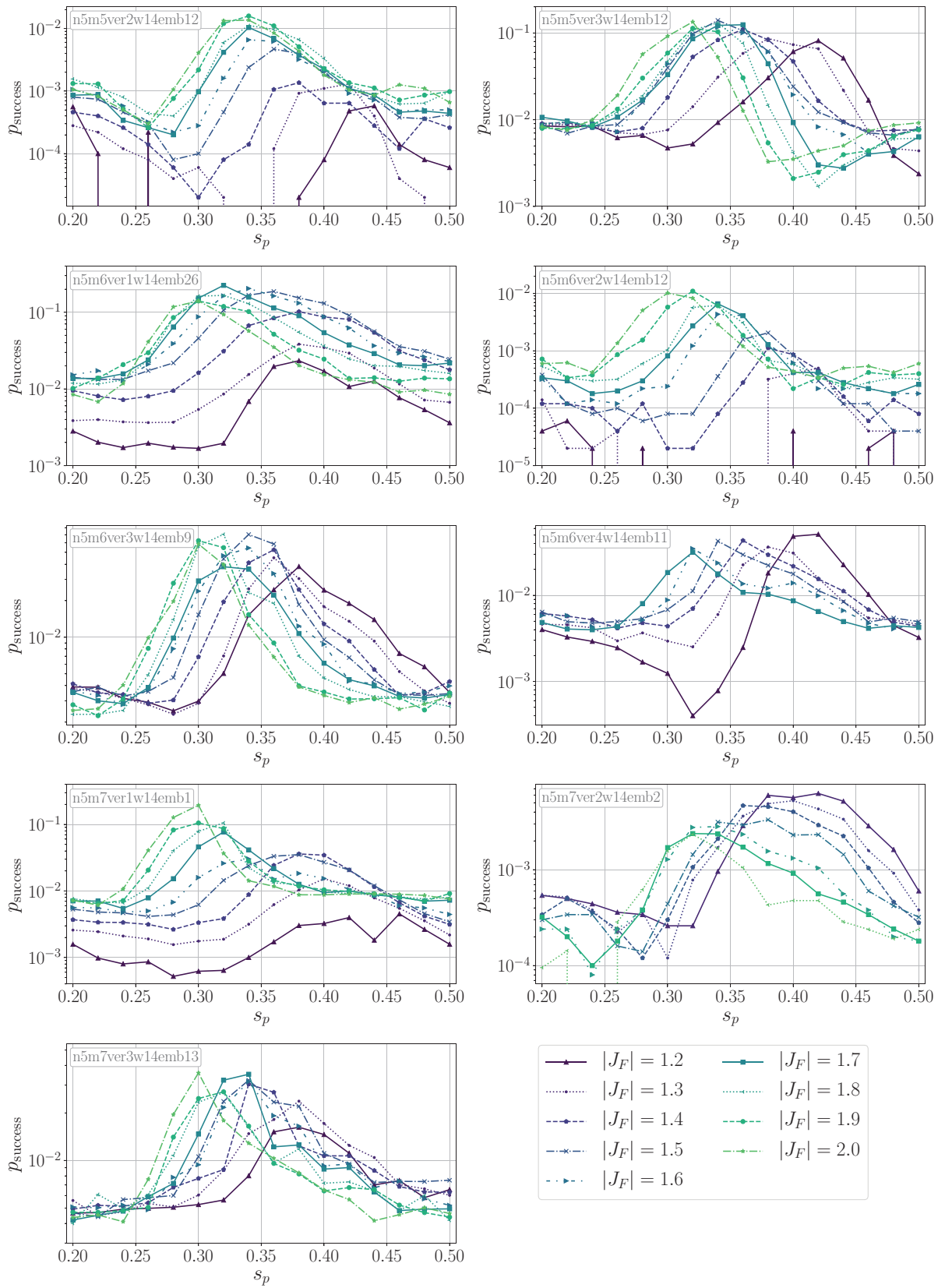


FIG. 11. The shift of the optimal pause location with $|J_F|$ (for multiple instances). The shifting of the optimal pause location with $|J_F|$ for multiple instances with a 100- μ s pause. Note that the scale of the y axis is different across instances.

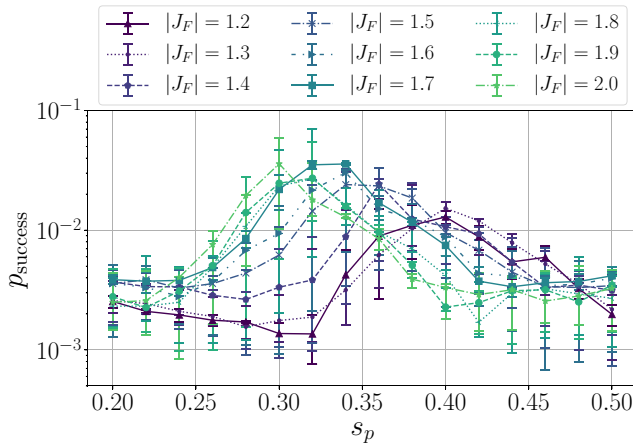


FIG. 12. The shift of the optimal pause location with $|J_F|$ (ensemble). The probability of success for an ensemble of nine instances of $n = 5$ with a pause duration $t_p = 100 \mu\text{s}$, and $t_a = 1 \mu\text{s}$.

and the size of the vertex models. Embedding statistics for a future D-Wave architecture (Pegasus) are also given.

APPENDIX E: DETAILS ON UNSOLVED INSTANCES

As a special case of the improvement in T_S , we find that for certain problems, the no-pause annealing fails to find a solution even after 50 000 reads, while the annealing with an appropriate pause is able to find one. In particular, out of the 45 instances tested, the no-pause annealing fails to solve seven of them. Of those seven, there are three that remain unsolved by any of the pause runs (we are considering a total of ten pause runs, resulting from two pause locations $s_p = \{0.3, 0.32\}$ and five pause durations $t_p = \{0.25, 0.5, 0.75, 1, 2\} \mu\text{s}$), while the other four

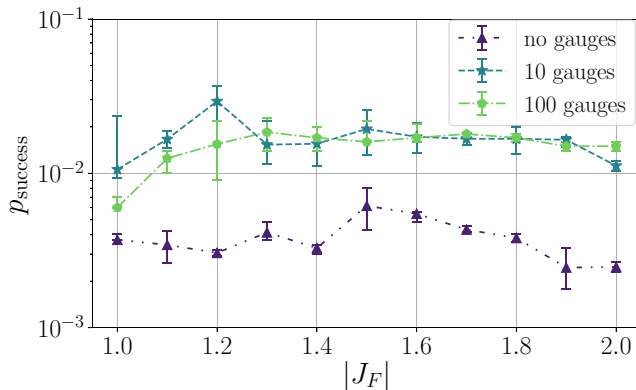


FIG. 13. The improvement of the probability of success with partial gauges. The effect of partial gauges on the probability of success for ten of $n = 4$ instances with varying $|J_F|$ and a no-pause schedule.

are solved by most or all of them: two are solved by ten out of the ten pause runs, one is solved by nine pause runs, and the other one is solved by eight pause runs. There are also two other instances that are solved by the no-pause runs but that, respectively, one and three of the pause runs cannot solve (but all the rest can). There are no instances that are solved by the no-pause runs but are not solved by the pause runs. Of the ten pause runs, the worst one cannot solve six of the 45 instances (making it better than the no-pause runs in that metric). The second worst cannot solve five, there are two that cannot solve four, and the other six cannot solve three.

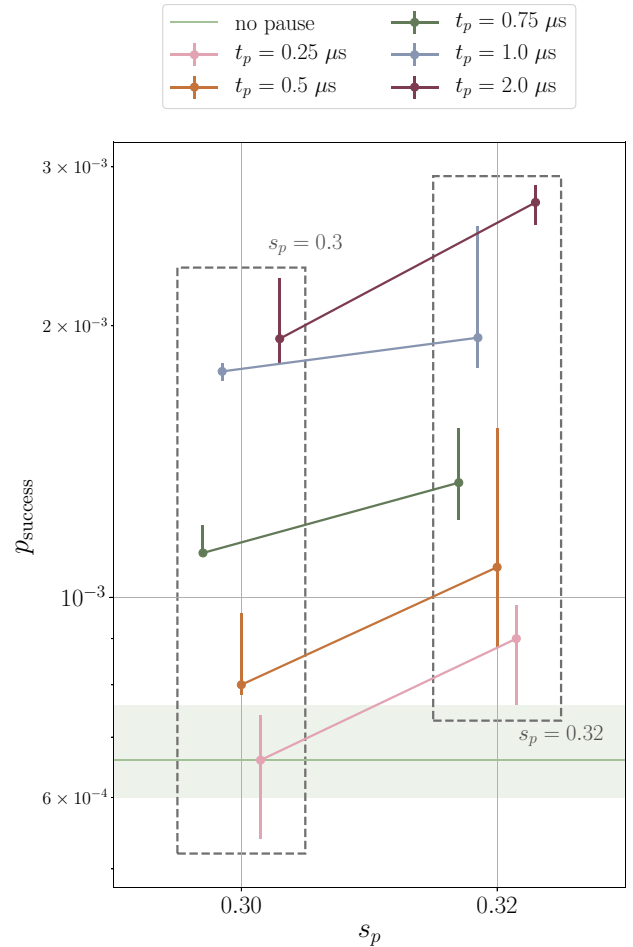


FIG. 14. The effect of the pause duration on the probability of success corresponding to T_S shown in Fig. 3 in Sec. B. With pause durations of $\{0.25, 0.5, 0.75, 1, 2\} \mu\text{s}$ and $|J_F| = 1.8$, the probability of success for an ensemble of 45 instances is shown for pause locations $s_p = 0.3$ and $s_p = 0.32$ (which we find to be optimal during the initial sweep). The reference (horizontal line and band for the median and the 35th and 65th percentiles, respectively) is the no-pausing case with parameters optimal for T_S : $t_a = 1 \mu\text{s}$ and $|J_F| = 1.6$. The data points show the median, with error bars at the 35th and 65th percentiles, after performing 10^5 bootstraps over the set of instances.

APPENDIX F: SUPPORTING INSTANCES SHOWCASING THE SHIFT OF THE OPTIMAL PAUSE LOCATION, IMPROVEMENTS WITH PARTIAL GAUGES, AND THE EFFECT OF PAUSE ON PROBABILITY OF SUCCESS

In Fig. 10, we show the shift of the optimal pause location with $|J_F|$ for a problem instance for bounded degree $\Delta = 3$.

In Fig. 11, we show a few more instances from the instance ensemble for $\Delta = 2$, $n = 5$.

Figure 12 illustrates the clear shifting of the optimal pause location for an instance ensemble over all $|J_F|$ values that we examine (in the range $[1.2, 2]$). For clarity on the figure, pausing results for just three values of $|J_F|$ are shown earlier in the bottom panel of Fig. 4 and discussed in Sec. C.

As discussed in Sec. D, the improvement in p_{success} saturates as one increases the number of gauges applied. Figure 13 shows, for an $n = 4$ ensemble, that applying as few as ten gauges yields similar p_{success} to 100 gauges. As detailed in Sec. D, these results indicate that with ten gauges we are already likely to encounter one or more positive outliers, leading to the large improvement in p_{success} . As the number of gauges increases further, the effect is not as dramatic, indicating that the spread in the gauge quality approaches the intrinsic distribution.

Figure 3 in Sec. B contains results for T_S for an ensemble in the narrowed parameter range discussed in this section. The corresponding p_{success} values are shown in Fig. 14.

-
- [1] P. Ray, B. K. Chakrabarti, and Arunava Chakrabarti, Sherrington-Kirkpatrick model in a transverse field: Absence of replica symmetry breaking due to quantum fluctuations, *Phys. Rev. B* **39**, 11828 (1989).
 - [2] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, Quantum annealing: A new method for minimizing multidimensional functions, *Chem. Phys. Lett.* **219**, 343 (1994).
 - [3] Tadashi Kadowaki and Hidetoshi Nishimori, Quantum annealing in the transverse Ising model, *Phys. Rev. E* **58**, 5355 (1998).
 - [4] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda, A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem, *Science* **292**, 472 (2001).
 - [5] Roman Martoňák, Giuseppe E. Santoro, and Erio Tosatti, Quantum annealing by the path-integral Monte Carlo method: The two-dimensional random Ising model, *Phys. Rev. B* **66**, 094203 (2002).
 - [6] Giuseppe E. Santoro, Roman Martoňák, Erio Tosatti, and Roberto Car, Theory of quantum annealing of an Ising spin glass, *Science* **295**, 2427 (2002).
 - [7] J. Marshall, D. Venturelli, I. Hen, and E. G. Rieffel, Power of Pausing: Advancing Understanding of Thermalization in Experimental Quantum Annealers, *Phys. Rev. Appl.* **11**, 044083 (2019).
 - [8] G. Passarelli, V. Cataudella, and P. Lucignano, Improving quantum annealing of the ferromagnetic p -spin model through pausing, *Phys. Rev. B* **100**, 024302 (2019).
 - [9] H. Chen and D. A. Lidar, Why and When Pausing is Beneficial in Quantum Annealing, *Phys. Rev. Appl.* **14**, 014100 (2020).
 - [10] Vijay V. Vazirani, *Approximation Algorithms* (Springer Science & Business Media, Berlin, Germany, 2003).
 - [11] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms* (MIT Press, Cambridge, MA, 2009).
 - [12] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman & Co., New York, NY, USA, 1979).
 - [13] V. Choi, The effects of the problem Hamiltonian parameters on the minimum spectral gap in adiabatic quantum optimization, *Quantum Inf. Process.* **19**, 90 (2020).
 - [14] Eleanor G. Rieffel, Davide Venturelli, Bryan O’Gorman, Minh B. Do, Elicia M. Prystay, and Vadim N. Smelyanskiy, A case study in programming a quantum annealer for hard operational planning problems, *Quantum Inf. Process.* **14**, 1 (2015).
 - [15] Andrew Lucas, Ising formulations of many NP problems, *Front. Phys.* **2**, 5 (2014).
 - [16] V. Choi, Minor-embedding in adiabatic quantum computation: I. The parameter setting problem, *Quantum Inf. Process.* **7**, 193 (2008).
 - [17] Yan-Long Fang and P. Warburton, [arXiv:1905.03291](https://arxiv.org/abs/1905.03291) (2019).
 - [18] D. Venturelli, S. Mandrà, S. Knysh, B. O’Gorman, R. Biswas, and V. Smelyanskiy, Quantum Optimization of Fully-Connected Spin Glasses, *Phys. Rev. X* **5**, 031040 (2015).
 - [19] Zhihui Wang, Mostafa Adnane, Bryan O’Gorman, Stuart Hadfield, Riccardo Mengoni, Davide Venturelli, Zoe Gonzalez Izquierdo, and Eleanor Rieffel, Mapping spanning graph problems to quantum heuristics: Techniques for handling global connectivity (to be published).
 - [20] QPU properties: D-wave 2000Q system at NASA Ames, D-Wave User Manual 09-1151A-D (2019).
 - [21] Sergio Boixo, Troels F. Rønnow, Sergei V. Isakov, Zhihui Wang, David Wecker, Daniel A. Lidar, John M. Martinis, and Matthias Troyer, Evidence for quantum annealing with more than one hundred qubits, *Nat. Phys.* **10**, 218 (2014).
 - [22] Troels F. Rønnow, Zhihui Wang, Joshua Job, Sergio Boixo, Sergei V. Isakov, David Wecker, John M. Martinis, Daniel A. Lidar, and Matthias Troyer, Defining and detecting quantum speedup, *Science* **345**, 420 (2014).
 - [23] Tameem Albash and Jeffrey Marshall, Comparing Relaxation Mechanisms in Quantum and Classical Transverse-Field Annealing, *Phys. Rev. Appl.* **15**, 014029 (2021).
 - [24] Sergio Boixo, Tameem Albash, Federico M. Spedalieri, Nicholas Chancellor, and Daniel A. Lidar, Experimental signature of programmable quantum annealing, *Nat. Commun.* **4**, 2067 (2013).
 - [25] M. Kim, D. Venturelli, and K. Jamieson, in *Proceedings of the ACM Special Interest Group on Data Communication*,

- Series and Number SIGCOMM '19 (Association for Computing Machinery, New York, NY, USA, 2019) p. 241.
- [26] J. Marshall, A. Di Gioacchino, and E. G. Rieffel, Perils of embedding for sampling problems, *Phys. Rev. Res.* **2**, 023020 (2020).
- [27] R. Ravi, Madhav V. Marathe, S. S. Ravi, Daniel J. Rosenkrantz, and Harry B. Hunt III, in *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, ACM (Association for Computing Machinery, New York, NY, USA, 1993) p. 438.
- [28] Martin Fürer and Balaji Raghavachari, in *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms* (Society for Industrial and Applied Mathematics, 1992) p. 317.
- [29] Michel X. Goemans, in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, Series and Number FOCS '06, IEEE (IEEE Computer Society, USA, 2006) p. 273.
- [30] Mohit Singh and Lap Chi Lau, in *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, ACM (Association for Computing Machinery, New York, NY, USA, 2007) p. 661.
- [31] Jochen Könemann and R Ravi, in *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, ACM (Association for Computing Machinery, New York, NY, USA, 2000) p. 537.
- [32] M. S. Zahrani, Martin J. Loomes, J. A. Malcolm, and Andreas A. Albrecht, A local search heuristic for bounded-degree minimum spanning trees, *Eng. Optim.* **40**, 1115 (2008).
- [33] Samir Khuller, Balaji Raghavachari, and Neal Young, Low-degree spanning trees of small weight, *SIAM J. Comput.* **25**, 355 (1996).
- [34] Raja Jothi and Balaji Raghavachari, Degree-bounded minimum spanning trees, *Discr. Appl. Math.* **157**, 960 (2009).
- [35] Thang N. Bui and Catherine M. Zrnčić, in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, ACM (Association for Computing Machinery, New York, NY, USA, 2006) p. 11.
- [36] Thang N. Bui, Xianghua Deng, and Catherine M. Zrnčić, An improved ant-based algorithm for the degree-constrained minimum spanning tree problem, *IEEE Trans. Evol. Comput.* **16**, 266 (2011).
- [37] Mohan Krishnamoorthy, Andreas T. Ernst, and Yazid M. Sharaiha, Comparison of algorithms for the degree constrained minimum spanning tree, *J. Heuristics* **7**, 587 (2001).