IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 August 2023, accepted 12 September 2023, date of publication 20 September 2023,
date of current version 27 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3317698

RESEARCH ARTICLE

Efficient Pause Location Prediction Using Quantum Annealing Simulations and Machine Learning

MICHAEL ZIELEWSKI[1], KEICHI TAKAHASHI[1,2], (Member, IEEE), YOICHI SHIMOMURA[2], AND HIROYUKI TAKIZAWA[1,2], (Member, IEEE)

[1]Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan
[2]Cyberscience Center, Tohoku University, Sendai 980-8579, Japan

Corresponding author: Michael Zielewski (michael@hpc.is.tohoku.ac.jp)

This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) Next Generation High Performance Computing Infrastructures and Applications Research and Development (R&D) Program ''R&D of A Quantum-Annealing-Assisted Next Generation High Performance Computing (HPC) Infrastructure,'' and in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant 22K19764.

ABSTRACT Despite increases in qubit count and connectivity in quantum annealers, a quantum speedup has yet to be observed for problems of practical significance. In order to further improve annealer performance, some researchers focus on tuning annealer parameters, such as the annealing schedule. In this work, we focus on pausing, an annealing schedule modification that has been shown to improve the probability of solving an optimization problem by orders of magnitude. However, a challenge associated with pausing is selecting an appropriate pause location, as pausing is only effective in problem-dependent regions and ineffective elsewhere. Moreover, there is little advice on how to determine where pausing is effective. Thus, pausing effectively is difficult and often inaccessible to the majority of users. To address these issues, we propose a data-driven method that leverages machine learning to predict optimal pause locations. First, we construct a dataset consisting of optimization problems and their corresponding optimal pause locations. The optimal pause locations are determined using spin-vector Monte Carlo, a method known to yield results similar to quantum annealing. Next, we train a convolutional neural network on this dataset, demonstrating its ability to distinguish between problem types and accurately predict the optimal pause location. Finally, we evaluate the model on multiple types of optimization problems. Our results show that the pause locations predicted by our method improve solution quality for all selected problem types. Additionally, our model can be pre-trained and easily distributed, making the power of pausing accessible to users unfamiliar with annealing schedule modifications.

INDEX TERMS Annealing schedule, convolutional neural network, machine learning, Monte Carlo simulation, pausing, quantum annealing, thermalization.

I. INTRODUCTION

Quantum annealing (QA) is a quantum-inspired heuristic algorithm designed to solve optimization problems in the form of the Ising model [1]. By leveraging quantum fluctuations, QA has the potential to explore the solution space of a problem more efficiently than classical algorithms. Commercially available annealers produced by D-Wave have a sufficient number of qubits to solve optimization problems of nontrivial size and have prompted comparisons with classical solvers in the search for quantum speedup [2]. Evaluations demonstrating the superiority of QA over classical methods typically only arise under specific conditions. For instance, results may indicate that QA outperforms certain selected classical algorithms such as simulated annealing, but does not necessarily outperform the algorithms that exhibit the

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Huang.

best performance for the target problem [3]. While simulated annealing is often compared to QA due to its ease of use and similar characteristics, it is important to note that other solvers, including complete search solvers, may exhibit superior performance and may be more suitable for making a practical comparison [4]. Additionally, an advantage with QA can be shown when specially crafted problems are selected for the evaluation [5]. These problems are usually designed to have properties that challenge classical solvers while providing an advantage to quantum solvers. However, such problems do not necessarily resemble typical optimization problems, and thus conclusions drawn from evaluations on these problems may not apply when solving other real-world problems [6].

One tunable annealing parameter that significantly affects results is the annealing schedule. Increasing the anneal time is a simple adjustment to the standard forward annealing schedule that is generally expected to improve solution quality by reducing diabatic transitions [3]. Alternatively, more advanced schedules, such as those that include a pause or perform reverse annealing, may offer even greater improvements in performance [7], [8]. However, these types of schedules are more complex as they introduce additional variables, such as pause location and pause duration for schedules with a pause, and initial state and reversal location for reverse annealing. Consequently, extra care must be taken when creating these types of schedules, as only a small subset of configuration values results in an improvement in solution quality.

For annealing schedules with a mid-anneal pause, there is little advice on choosing the pause location, which is the primary variable responsible for whether a pause will improve performance [8]. This is evident in many works that investigate the effects of pausing, where authors often employ a grid-search approach and evaluate schedules with pauses placed at predetermined intervals within a given range of pause locations [8], [9]. The primary downside of this approach is that it requires a significant amount of annealer access time. Based on the observation that problems belonging to the same problem class may share optimal pause locations, a method was proposed that reduces the amount of annealer access time required when solving many related problems [10]. However, the method still has a high initial cost and needs to be repeated for problems that are sufficiently different.

The scarcity of literature on pause location selection and current methods used in related works shows that there is room for improving pause location selection methods. In this work, we propose a method to address the issues associated with selecting a pause location. Our proposal includes two key components. First, we utilize a variant of the spin-vector Monte Carlo (SVMC) method to simulate QA and determine the optimal pause location for an optimization problem. The primary benefit of this approach is that no annealer access time is required to find the optimal pause location. An additional benefit of our proposal is that no knowledge of the solution to the problem is required, as we determine the optimal pause location from the cost function of the target problem. In the second component, we train a neural network capable of predicting the optimal pause location. The advantage of using a neural network is that once trained, predictions are produced on the order of tens of milliseconds, offering a significant speedup over a grid search utilizing quantum hardware. Moreover, the weights of the trained model can be easily distributed, granting users unfamiliar with modifying annealing schedules the opportunity to use pausing to improve their results. Our results demonstrate the effectiveness of our proposal in accurately predicting optimal pause locations that yield improvements in various metrics, including energy reduction, the probability of solving a problem, and the fraction of problems solved.

The rest of this paper is organized as follows: In Section II, we review QA, pausing, and the method we use to simulate QA. Next, in Section III, we discuss in detail our proposed method. Section IV contains various details about our evaluation, including parameters for our simulations and model, types of problems and parameters used to generate unique instances, annealing parameters, and metrics used to evaluate the effectiveness of our proposal. Our results and discussion are presented in Section V. Finally, we draw our conclusions in Section VI.

II. BACKGROUND

A. QUANTUM ANNEALING

Quantum annealing solves optimization problems whose solutions can be encoded as the ground state, or minimal energy state, of the classical Ising model in the form of

$$H_{\text{Ising}}(\sigma) = \sum_{i=1}^{N} h_i \sigma_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{i,j} \sigma_i \sigma_j, \tag{1}$$

where the local biases h_i and coupling terms $J_{i,j}$ define the optimization problem, N is the number of variables, and $\sigma_i \in \{-1, 1\}$ are the spin values for each variable. After being programmed on the annealer, the system evolves under the time-dependent quantum Hamiltonian

$$H(s) = A(s)H_D + B(s)H_P, \tag{2}$$

where H_D is the transverse field driver Hamiltonian $H_D = -\sum_{i=1}^{N} \sigma_i^x$ providing quantum fluctuations, H_P is the problem Hamiltonian $H_P = \sum_{i=1}^{N} h_i \sigma_i^z + \sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{i,j} \sigma_i^z \sigma_j^z$ obtained by replacing spins in (1) with the σ^z Pauli matrix, and $A(s)$ and $B(s)$ are functions determining the strengths of H_D and H_P, respectively. These functions depend on the annealing schedule parameter $s \in [0, 1]$. In the case of the standard forward anneal, an example of which is shown in Fig. 1, s is given by $s = t/t_a$, where t is time and t_a is the annealing time. However, s can also take different forms depending on the desired schedule.

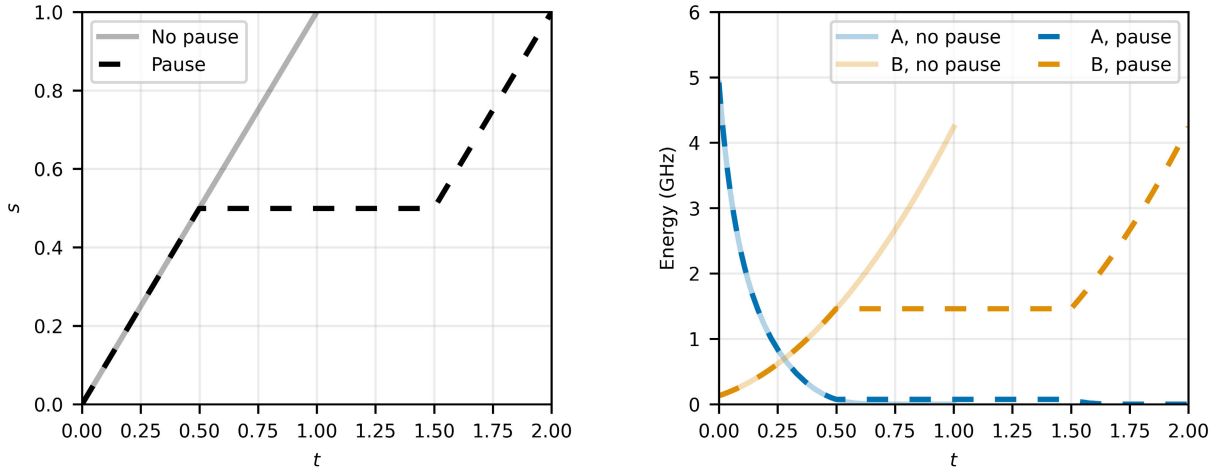


FIGURE 1. (left) Annealing schedules for a 1 µs forward anneal and the same schedule with a 1 µs pause at $s = 0.5$. (right) Corresponding $A(s)$ and $B(s)$ for the two schedules.

B. PAUSING

By adjusting the rate of change of s with respect to t, a wide range of unique forward annealing schedules can be generated. A pause is introduced when this rate is set to zero for a period of time, causing s to remain constant. Forward annealing schedules that incorporate a single mid-anneal pause can be characterized by two parameters: the pause location $s_p \in [0, 1]$, indicating the point in the schedule where the pause starts, and the pause duration t_p, representing the length of the pause. Fig. 1 shows an example schedule containing a pause in which $t_a = 1$ µs, $s_p = 0.5$, and $t_p = 1$ µs.

Marshall et al. [8] conducted a study on the effects of pausing in D-Wave annealers and demonstrated that pausing can lead to orders of magnitude enhancement in P_{solve}, the probability of finding the ground state or solving an optimization problem. However, their results also showed that pauses are only effective within an instance-dependent region. Outside this region, pausing provided no advantage over forward annealing schedules without a pause. Therefore, the time allocated to ineffective pauses could be better utilized by performing more anneals without pauses. These results emphasize the importance of properly placing pauses to efficiently utilize annealing time.

The theory behind why pausing improves results is also provided by Marshall et al. [8], which we briefly summarize here. Early in the anneal, H_D dominates and the system is expected to be in its instantaneous ground state. In the middle of the anneal, the system approaches the minimum gap, which is where the energy difference between the ground state and the next lowest energy state is smallest. In and around this region, there will be population loss from low energy states to higher energy states. Shortly after the minimum gap, a pause can be placed to increase population in lower energy states through thermalization. Finally, late in the anneal, dynamics are frozen and significant population transfer does not occur. Various works have expanded upon the theory and further investigated pausing [10], [11], [12], [13].

C. SVMC

Proposed by Shin et al. [14], SVMC is a classical model that has been shown to produce output similar to that of D-Wave annealers. SVMC uses a model consisting of rotors represented by angles $\theta_i \in [0, \pi]$ to represent qubits and has a Hamiltonian in the form of

$$H(s) = -A(s)\sum_{i=1}^{N}\sin\theta_i + B(s)\left(\sum_{i=1}^{N}h_i\cos\theta_i + \sum_{i=1}^{N}\sum_{j=i+1}^{N}J_{i,j}\cos\theta_i\cos\theta_j\right). \tag{3}$$

The model starts in an initial configuration, and evolves over time. At each timestep, or sweep, random new angles are proposed, and current angles are updated in Metropolis-Hastings style. Albash and Marshall [15] observe that this model fails to capture the effect of freeze-out [16], which can be described as a point late in the anneal when the transverse field strength is weak and after which no system dynamics occur. They propose spin-vector Monte Carlo with transverse-field-dependent updates (SVMC TF), a modified version of SVMC that attempts to replicate freeze-out. In this modified version, angle proposals are no longer completely random, but are instead selected from a range around current angles. The relative strength of the transverse field determines the size of the range, which is given by $[\theta_i - \min\{1,\ A(s)/B(s)\}\pi,\ \theta_i + \min\{1,\ A(s)/B(s)\}\pi]$. Early in the anneal this range is large, and angle proposals in SVMC TF may resemble those in SVMC. However, as the anneal progresses and the transverse field strength decreases, angle proposals will be closer to the current angle values.

Albash and Marshall demonstrate that their modification captures the effects of freeze-out and furthermore, produce results that resemble the output of annealing with a mid-anneal pause.

III. PAUSE LOCATION PREDICTION METHOD

A. MOTIVATIONS

In this work, we address four issues associated with selecting a pause location. These issues include practicality, the amount of knowledge about a problem necessary to use the method, run time, and ease of use and reproducibility.

First, the primary goal of this work is creating a method to select a pause location that improves annealing performance, as few works focus on determining where a pause should be placed. Existing approaches either provide guidelines that are not tailored to specific problem instances, or propose methods that are inefficient when solving multiple different types of problems [10], [12]. Additionally, other works investigating pausing find optimal pause locations in ways that can be considered inaccessible for general users of QA. For example, a grid search of pause locations is often employed to determine the effectiveness of pausing at various locations [9], [11]. However, such an approach requires a substantial amount of annealer access time for a single problem instance, and would be costly and impractical to perform for multiple instances. Another example is inferring the optimal pause location from the minimum gap of a problem [8]. Calculating the minimum gap is accomplished through exact diagonalization of the Hamiltonian, a process that is intractable for problems with more than a few tens of variables. Our proposal avoids being impractical as it predicts pause locations that improve performance, makes predictions tailored to specific problem instances, and requires no annealer access time.

The second issue that we address is determining the efficacy of a pause, which typically involves knowledge of the optimal configuration for a problem instance [8]. Given that a common goal when using QA for optimization is to find the optimal configuration for a problem, knowledge of that configuration pre-annealing is counterintuitive. For this reason, we do not assume that the optimal configuration is known.

The third issue that we address is the time in which the optimal pause location can be found. We develop a method that predicts pause locations without any significant delay, such as the time required for a grid search using QA or simulations of QA. Estimates of these times are provided in Table 1. A QA grid-search approach, such as the one performed in [10], can be performed in a few minutes. Most of this time is annealer access time, while a small amount comes from other factors, such as queueing and network overhead. The estimate for SVMC TF simulations of QA comes from experiments using our own single core implementation of the algorithm in this work. It is important to note that our implementation is not optimal in terms of execution time, and that while speedups can be achieved through code optimization, parallelization, or accelerators, we do not expect simulations to require less time than QA.

TABLE 1. Approximate timescales for determining optimal pause locations.

Approach	Timescale
SVMC TF Simulations	Tens of hours
QA Grid Search	Minutes

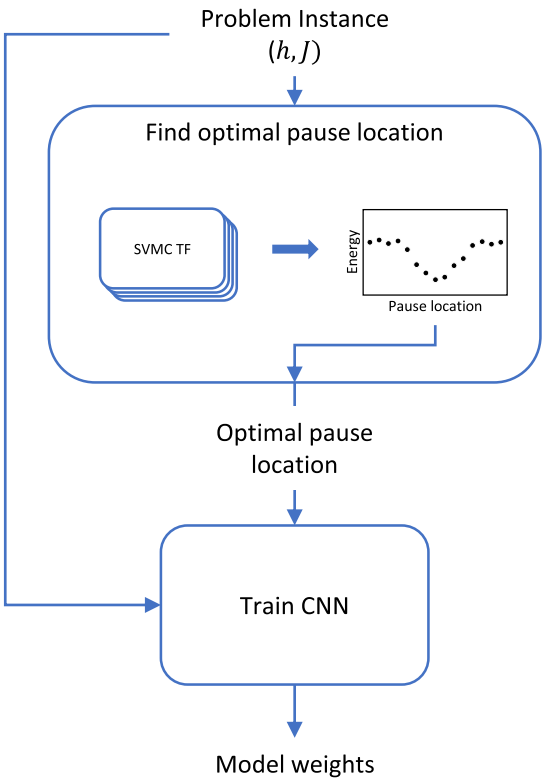


FIGURE 2. A diagram depicting our method. First, SVMC TF simulations are used to estimate the optimal pause location for an optimization problem. Next, a CNN is trained on pairs of optimization problems and their optimal pause locations. After training, the model that provides the most accurate predictions is saved for future use.

The final issue that we address relates to reproducibility and the ease at which a method can be distributed and used. Our method will be self-contained, easily distributable, and not require users to have special hardware for intense computations or other data that may not be publicly available.

B. PROPOSAL

In this work, we present a method for finding optimal pause locations that consists of two primary components: simulating QA to determine optimal pause locations and applying machine learning to predict optimal pause locations. An overview of our method is shown in Fig. 2. First, we use SVMC TF to simulate QA with a pause and determine the optimal pause location for a problem instance.

The motivation behind using SVMC TF is that it is a classical simulation of QA, and it has been shown to be able to simulate the effects of pausing. We repeat the simulation process many times to build a dataset comprising problem instances and their simulated optimal pause locations. Next, the dataset is used to train a convolutional neural network (CNN) to quickly predict optimal pause locations. CNNs have shown exceptional performance on tasks involving input data with spatial relationships. In QA, spatial relationships can be found in the coupling terms whose value and sign encode a relationship between variables, making CNNs a natural choice for our input data. After training, the model can be distributed in the form of model weights.

In the first component of our method, we use SVMC TF to create a dataset consisting of pairs of problem instances and their optimal pause locations. The first step in this component is formulating optimization problems as their Ising equivalent, in the form given in (1). Each type of problem will have a unique formulation process, and some examples can be found in [17]. The output of problem formulation is the weights of an Ising model, which can be represented by an $N \times N$ matrix with the local biases h on the diagonal and the coupling terms J in the off-diagonal elements. Following formulation, the next steps adjust the instance to be identical to what would be solved on the annealer, but are not always necessary depending on the specific instance. One adjustment that may be required is called embedding [18]. Embedding refers to the process in which an instance is mapped to qubits on the annealer. Due to the limited physical connectivity of the annealer, a single variable may be mapped to more than one qubit. After embedding, the instance may also need to be rescaled to fit within the supported bias and coupler ranges of the annealer, which may vary by annealer generation. In addition to these adjustments, SVMC TF must also be configured to match a specific annealer. The values necessary for this configuration are $A(s)$, $B(s)$, and the temperature of the annealer, all of which are provided by D-Wave [19].

After the problem instance has been prepared and SVMC TF configured, SVMC TF simulations can be run to obtain spin configurations for the problem instance. These spin configurations correspond to the embedded problem instance, and as such must be further processed, or unembedded, in order to obtain a configuration for the unembedded problem instance. To determine the effectiveness of a pause, many works typically perform unembedding and calculate P_{solve}. However, calculating P_{solve} requires knowledge of the ground state or solution of an optimization problem, which may not be known in realistic applications of QA. Thus, to increase the applicability of our method, we do not use P_{solve} to measure the benefit of pausing at a location and instead use the energy associated with embedded spin configuration output from an SVMC TF simulation. To determine the optimal pause location for a problem, many SVMC TF simulations are repeated at each pause location, and the pause location that minimizes the average energy of the embedded samples is selected.

The second component of our method is a CNN that will learn the relationship between problem instances and their optimal pause locations. The input to the CNN is the same matrix of weights prepared for SVMC TF. After passing the weights as input through convolutional layers, a fully connected layer is used to transform the learned features into a pause location. The model is then trained to minimize loss, as calculated by the mean squared error (MSE) between the simulated and predicted optimal pause locations. During training, the model weights minimizing loss represent the best model and can be saved for future use.

Our proposal addresses several gaps in current research and offers advantages over simple methods, such as grid search, for finding optimal pause locations. One advantage of our proposal is that no annealer access time is used in determining the optimal pause location, as SVMC TF is a purely classical algorithm. Another advantage of our proposal is that it does not require knowledge of the optimal configuration for a problem because it measures the efficacy of a pause by changes in energy. A third advantage of our method is the time in which predictions can be made. The inference time of a CNN, or the time it takes to produce output from input values, can be accomplished in tens of ms [20], which is significantly faster than either approach in Table 1. A final advantage of our method is the trained model is easy to use and readily distributable in the form of model weights, allowing a wide range of users to benefit from pausing.

IV. EVALUATION SETUP

In this section, we present the specific details of our evaluation, including the problem types in our dataset, parameters for SVMC TF simulations and the CNN, QA parameters, and evaluation metrics.

A. PROBLEM TYPES

To evaluate our proposal, we select four types of problems. The first problem type we include is the Sherrington Kirkpatrick (SK) model [21], a model consisting of random coupling terms that is commonly used in evaluations of QA. We generate instances with no local fields and coupling terms randomly sampled from a normal distribution with a mean of 0 and a standard deviation of 1. The second problem type we use is the subset sum problem (SSP), which seeks to determine whether there exists a subset of a random set of integers that sums up to a target value [10]. Generated instances use integers sampled uniformly from the range $[1, 50]$, and set the target sum to be half of the sum of the set of integers. The third and fourth problem types we use are based on the not-all-equal 3-satisfiability (NAE3SAT) problem, an NP-complete variation of the classic 3-satisfiability problem [22]. We differentiate between these final two problem types by their clause to variable ratio, a parameter that sets the total number of clauses to be satisfied in an instance to be a multiple of the number of variables.

In the third type, the clause to variable ratio is set to 1, while in the fourth type the clause to variable ratio is set to 2. These types are labeled NAE3SAT-1 and NAE3SAT-2, respectively. All problem instances contain 50 variables, however, the number of qubits used may be higher due to embedding.

B. SVMC TF AND CNN PARAMETERS

All of the problem instances in this work will require embedding for QA. Hence, we carry out an identical embedding procedure prior to SVMC TF simulations. Two of the problem types used in this work, the SK and SSP, are represented by fully connected graphs and can use the same embedding. This embedding, called a clique embedding, is found using software provided by D-Wave [23]. The other two problem types, the NAE3SAT-1 and NAE3SAT-2, are not fully connected. Because connectivity varies by problem instance, an embedding that works for one instance may not work for others. Thus, we generate one embedding per instance using a heuristic embedding method [18]. After embeddings have been generated and problem instances have been embedded, 1,000 spin samples are generated using SVMC TF. Each simulation run of SVMC TF consists of 20,000 sweeps, half of which are pause sweeps for which $A(s)$ and $B(s)$ are held constant at the designated pause location. This specific number of sweeps was selected to match a 2 µs schedule containing a 1 µs pause, based on the values in [15]. Pause locations are selected at intervals of 0.01 in the range [0.3, 0.7].

The CNN used in this work follows patterns common in other popular CNN architectures [24], [25]. The network consists of five convolutional layers each having 128 filters and a kernel size of 3×3. Each convolutional layer is followed by a single max pooling layer with a kernel size of 2. The final layer in the model is a fully connected layer that produces a single value. The rectified linear unit activation function is used after each convolutional layer, and we apply the sigmoid function to the output of the fully connected layer to ensure the output falls within the range [0, 1]. Since the arrangement of layers in our CNN expects input of a certain size, we zero-pad all input weight matrices to the size of our largest input. The dataset we use consists of 1,000 problems of each of the problem types introduced in Section IV-A for a total of 4,000 problems. Before training, we partition the dataset into training and validation sets using an 80-20 split, while maintaining the balance of problem types in each set. The model is trained on the training set, and the validation set is used to monitor performance and determine the set of weights resulting in the lowest MSE.

C. QA PARAMETERS

We use QA to determine the effectiveness of an annealing schedule with a pause at a location predicted by our method. The baseline that we compare our proposal with is a standard forward annealing schedule. We generate 100 additional instances per problem type for this comparison, ensuring instances are not identical to those used to train our CNN. An identical approach to embedding as detailed in Section IV-B is employed, and the same embedding is used for both the baseline and our proposal. We set $t_a = 1$ µs for all anneals, and anneals with a pause are assigned a pause duration of $t_p = 1$ µs. Each problem is annealed 10,000 times with 50 gauge transformations, a process used to reduce the effects of hardware biases on our results [11]. Lastly, during the unembedding process, we discard samples that contain inconsistent values. Specifically, if the qubits representing a single logical variable take multiple values, often referred to as a broken chain, the entire sample is discarded. When all qubits representing a logical variable take the same value, the logical variable is assigned that value. All anneals are executed on the D-Wave Advantage system [26].

D. METRICS

In this work, we use four metrics to evaluate our proposal. The first metric is the average energy of the unembedded samples returned from the annealer. This value represents the quality of a solution, and lower value is better. The second metric is P_{solve}, representing the likelihood that a correct solution is returned for a single anneal, and can be calculated as

$$P_{\text{solve}} = \frac{\text{number of optimal configuration}}{\text{number of anneals}}. \tag{4}$$

Calculating P_{solve} requires knowledge of the optimal solution for each problem instance, which we calculate using classical solvers. The third metric is the fraction of instances solved, calculated as

$$F_{\text{solve}} = \frac{\text{number of } P_{\text{solve}} > 0}{\text{number of instances}}. \tag{5}$$

The final metric is time-to-solution (TTS) with 99% probability, given by

$$\text{TTS} = \frac{\log(1 - 0.99)}{\log(1 - P_{\text{solve}})} \cdot t_{\text{t}}, \tag{6}$$

where the total annealing time $t_{\text{t}} = t_a$ for a schedule with no pause, or $t_{\text{t}} = t_{\text{a}} + t_p$ for a schedule with a pause.

E. COMPARISON WITH OTHER WORK

Reference [10] is one work that proposes a specific method for determining optimal pause locations. Their method uses a grid-search approach to find the optimal pause location for multiple instances, as measured by P_{solve}. After determining the optimal pause locations for multiple instances, the method selects an optimal pause location for an entire class of problems. This method proves to be effective for the SSP, which is used in their evaluation. For the purpose of comparison against a known effective method, we reproduce their approach, which we refer to as class-wise profiling (CWP), in our evaluation. As CWP was only originally evaluated on the SSP, we reproduce CWP for the SSP only. In our reproduction of their method, we generate 30 problem instances for the grid search process, and use QA parameters defined in Section IV-C.

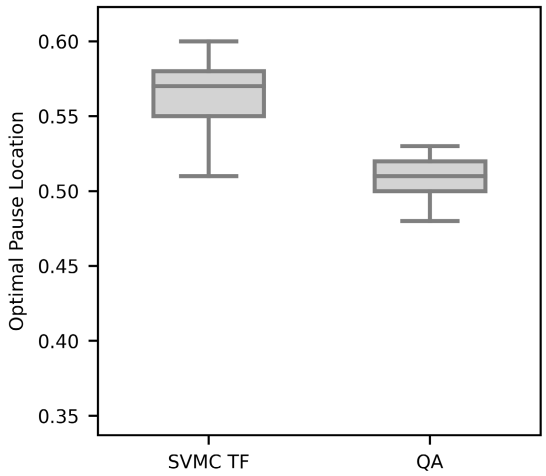


FIGURE 3. Distributions of SVMC TF and QA optimal pause locations for the SSP.

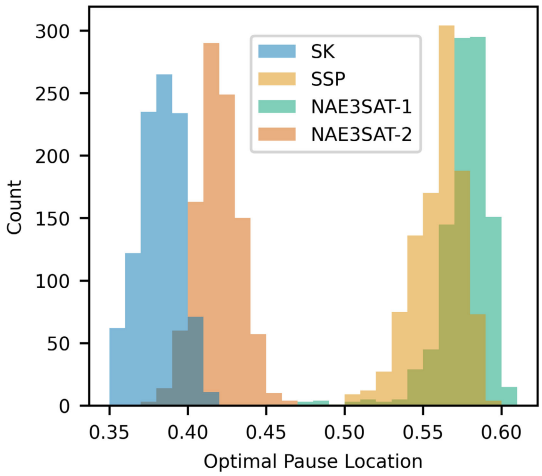


FIGURE 4. Distribution of optimal pause locations from SVMC TF simulations.

V. EVALUATION RESULTS

We present our results in this section, discussing the implementation of our method in Sections V-A and V-B, and focusing on the impact of our method on the annealing process in Section V-C.

A. SVMC TF SIMULATIONS

We first assess how closely the output of SVMC TF matches the output of QA. Specifically, we compare the optimal pause locations obtained using SVMC TF with those obtained by reproducing the CWP method from [10] with QA. Fig. 3 shows boxplots of the optimal pause locations corresponding to each method. This figure shows that the true optimal pause location is generally earlier in the anneal than what would be found using SVMC TF, and may suggest that tuning the numbers of anneal and pause sweeps may be beneficial [15]. While this may limit the peak effectiveness of our method, pauses are effective in the area surrounding the optimal pause location, and a benefit from pausing is still expected.

Next, we analyze the optimal pause locations within our dataset. The distribution of optimal pause locations from SVMC TF simulations for the problem types used in our evaluation is depicted in Figure 4, which illustrates the relationship between problem type and optimal pause location. Distinct regions are observed for the optimal pause locations of different problem types, including problem types that are similar but generated with different parameters, as is the case with the NAE3SAT. Additionally, the figure shows unimodal distributions for each problem type, but when combined into one dataset, a multimodal distribution is observed. This indicates that a model must be able to differentiate between different types of problems to accurately predict the optimal pause location.

B. CNN TRAINING

We also investigate how well our CNN is able to learn to predict the optimal pause locations in our dataset based on its lowest MSE. A simple baseline for prediction would be a method that predicts the mean optimal pause location of the training dataset. Such a method is similar to the one used in [10], and may work well for datasets consisting of one type of problem. However, as our dataset consists of multiple types of problems, we expect this method to result in a relatively high MSE, especially when considering the distribution shown in Fig. 4. A second baseline that would be more appropriate would be to predict the class-wise mean optimal pause location, assuming the problem classes are known at prediction time. This method can achieve a lower MSE as the optimal pause locations for a single class occur in a range that is smaller than that of the entire dataset. We show the performance of these two baseline methods, as well as our trained CNN, in Table 2. The data show that our trained CNN has the lowest MSE. This indicates that the CNN is able to learn both features that can distinguish between problem types, and features that make problem instances unique.

TABLE 2. Mean squared error comparison.

Method	MSE
Mean	7.5×10^{-3}
Class-wise Mean	2.6×10^{-4}
CNN	2.5×10^{-4}

C. QA EVALUATION

In this section, we analyze how well our proposal works when applied to a quantum annealer. We compare schedules with pauses at locations determined by our proposal with a baseline schedule containing no pause. For the SSP, we also show results for CWP [10].

We first investigate the impact of the various methods on energy. Our results are shown in Fig. 5, which contains boxplots of the mean energies for each problem instance. This

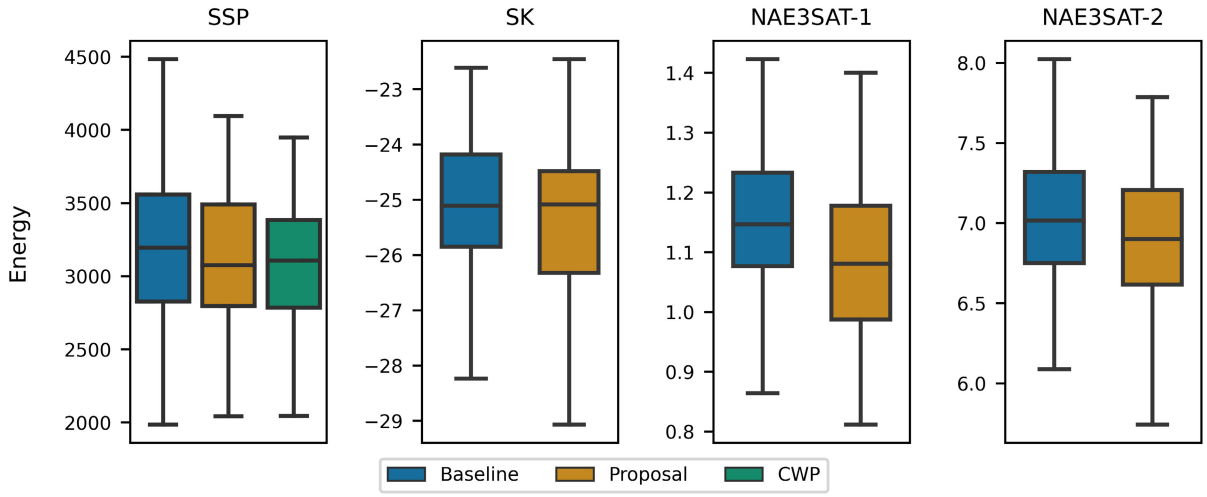


FIGURE 5. Distribution of the mean energy of configurations for each method and problem type.

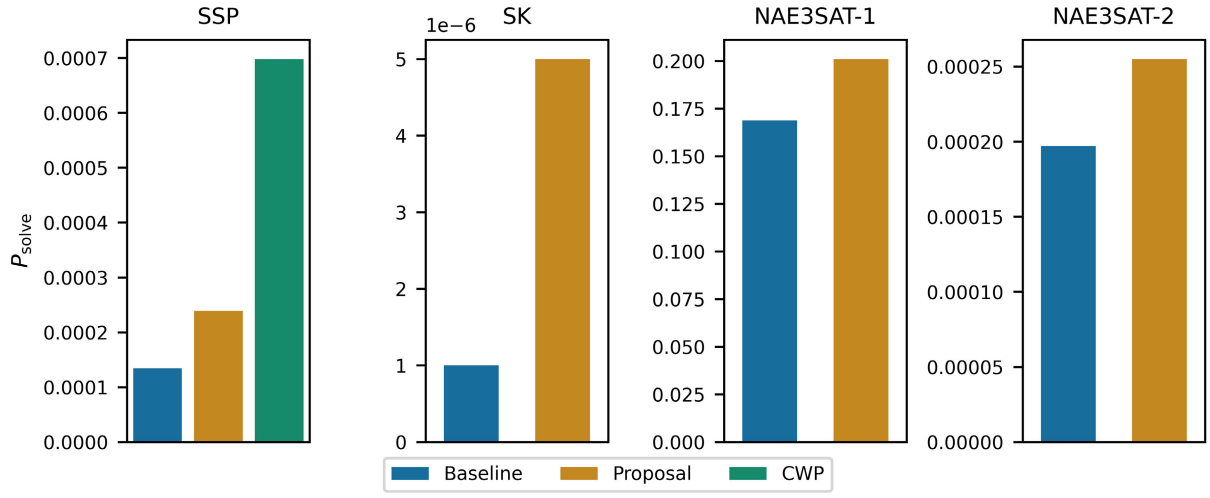


FIGURE 6. Mean P_{solve} for each method and problem type. Higher is better.

figure shows that switching from the baseline schedule to one created with our proposal results in a downward shift in the energy distribution. In a practical setting, this shift results in solutions that are of higher quality or are closer to optimal. Additionally, this shift is observed for all problem types, indicating that the optimal pause location found using SVMC TF is effective despite the differences observed in Fig. 3.

Next, we show the impact on P_{solve} in Fig. 6. Improvements in P_{solve} are shown for all problems when using the proposed method. For the SSP, CWP achieves the highest P_{solve}, likely because the pause location it predicts is closer to the true optimal pause location for each instance. As previously mentioned, the performance of our method can be brought closer to the performance of the method in [10] through further tuning of the number of sweeps performed in each SVMC TF run.

The fraction of instances solved by each method is shown in Fig. 7. The proposed method increases the number of instances solved for the SSP, SK, and NAE3SAT-2. As the baseline method already solves all NAE3SAT-1 instances, no further improvement is possible. As with the results for P_{solve}, CWP improves upon both the baseline method and the proposal, for SSP instances.

Finally, we examine how TTS is impacted by our method. The calculation for TTS given in (6) applies when $P_{\text{solve}} > 0$. However, many of the instances in our evaluation are not solved by QA, and thus have invalid values for TTS. In these cases, for the calculation of TTS only, we set $P_{\text{solve}} = 10^{-4}$ and $t_{\text{t}} = 2\ \mu\text{s}$. This modification is made for both the baseline and proposed methods, in order for a fairer comparison. The total TTS for all instances is shown in Fig. 8. For every type of problem, the proposed schedule results in a higher, or worse, TTS compared to the baseline. These results can be

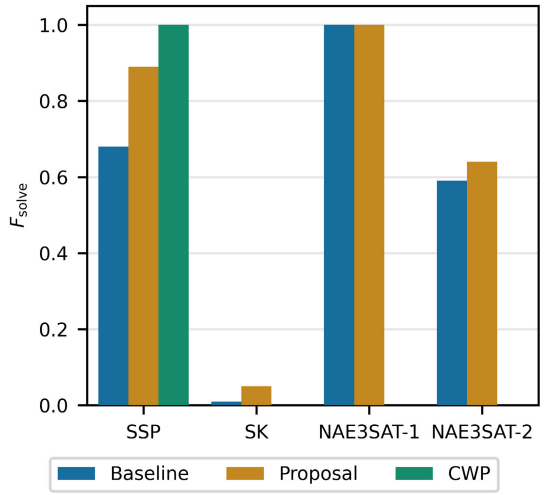


FIGURE 7. The fraction of instance solved at least once. Higher is better.

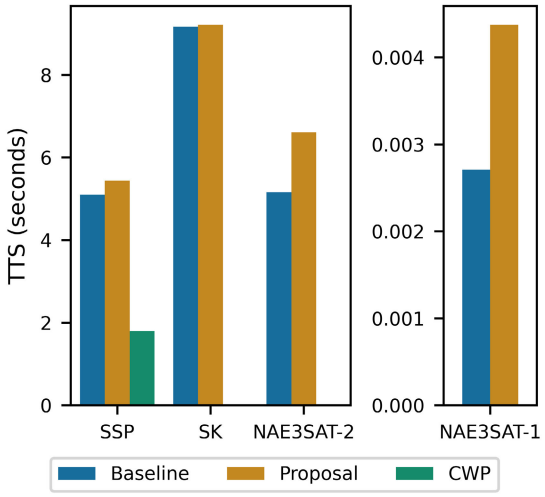


FIGURE 8. The sum of TTS values for each problem type. Lower is better.

explained by Fig. 6 and the duration of the pause; the relative increase in P_{solve} over the baseline is not great enough to offset the additional time spent pausing. While the proposal does not improve upon the baseline, CWP does, for the SSP. That method results in a reduction in TTS, similar what was observed in the original work.

In summary, we showed that our proposal achieves improvements in solution quality as measured by energy, P_{solve}, and the fraction of instances solved. These improvements are evidence that our method is effective and successfully identifies the region in which pauses are effective for each instance. However, one metric that our method did not show any improvement in is TTS. This can be attributed to a slight discrepancy between the predicted pause location and the true optimal pause location for a problem instance, resulting in limits on the maximum performance improvement. To remedy this, further investigation into the number of SVMC TF sweeps used to approximate a specific annealing time is required. Ultimately, these results show that while our method works effectively, modifying the simulation component of our method may further improve results.

VI. CONCLUSION AND FUTURE WORK

In this work, we identified a need within the QA community to find a method that determines the optimal pause location in an instance-wise fashion, without requiring any annealer access time, and in a user-friendly way. We then proposed a method in which we use SVMC TF simulations to train a CNN to predict optimal pause locations. Our results show that SVMC TF is suitable as a simulator of QA, and our model successfully learns features that are predictive of the optimal pause location. Moreover, the evaluation of our proposal shows its effectiveness. Our proposal results in an improvement in solution quality, P_{solve}, and the fraction of instances solved. We expect that more complex neural network architectures and implementations of SVMC TF that are fine-tuned to specific annealers will further boost the performance of our method, and potentially result in an improvement in TTS.

While our results do show the effectiveness of our proposal, they also show there is room for improvement. The discrepancy between SVMC TF optimal pause locations and those of QA indicate that SVMC TF is not a perfect simulator for QA. Investigating this is one potential future direction. We expect that tuning the approximation of sweeps to µs will be useful in improving the performance of our method. It will also be useful to determine if the optimal approximation for a specific annealing duration is consistent across problem types and annealers. Another area of interest is determining if there are any modifications that can be made to the core algorithm, such as the modification made to SVMC in SVMC TF, that result in more accurate simulations of QA. One such example can be random longitudinal field noise, which was shown in [27] to be necessary to accurately model the dynamics of QA.

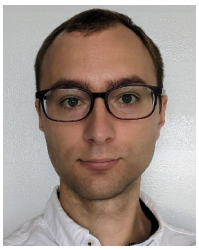
Another potential future direction involves methods for predicting the optimal pause location. Our evaluation shows that a CNN can learn to accurately predict optimal pause locations. This indicates that the CNN is learning features that are useful for prediction, however, it is not clear what the specific features are. Applying techniques that can help visualize or interpret the features learned by a model or encoded in the weights of a model will be useful in determining the features that have an effect on the optimal pause location. Alternatively, evaluating simpler and more interpretable models, such as linear regression, may also help to understand what impacts the optimal pause location.

REFERENCES

[1] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 58, no. 5, pp. 5355–5363, Nov. 1998, doi: 10.1103/PhysRevE.58.5355.

[2] (2020). *Practical Quantum Computing: An Update*. [Online]. Available: https://www.dwavesys.com/media/doaarwgt/14-1047a-a_practical_quantum_computing_an_update.pdf

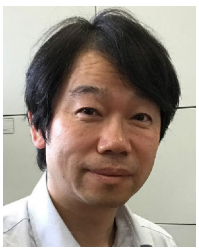
[3] T. Albash and D. A. Lidar, "Demonstration of a scaling advantage for a quantum annealer over simulated annealing," *Phys. Rev. X*, vol. 8, no. 3, Jul. 2018, Art. no. 031016, doi: 10.1103/PhysRevX.8.031016.
[4] M. Ohzeki, A. Miki, M. J. Miyama, and M. Terabe, "Control of automated guided vehicles without collision by quantum annealer and digital devices," *Frontiers Comput. Sci.*, vol. 1, pp. 1–12, Nov. 2019. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fcomp.2019.00009
[5] V. S. Denchev, S. Boixo, S. V. Isakov, N. Ding, R. Babbush, V. Smelyanskiy, J. Martinis, and H. Neven, "What is the computational value of finite-range tunneling?" *Phys. Rev. X*, vol. 6, no. 3, Aug. 2016, Art. no. 031015, doi: 10.1103/PhysRevX.6.031015.
[6] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, "Quantum annealing for industry applications: Introduction and review," *Rep. Prog. Phys.*, vol. 85, no. 10, 2022, Art. no. 104001, doi: 10.1088/1361-6633/ac8c54.
[7] E. Grant, T. S. Humble, and B. Stump, "Benchmarking quantum annealing controls with portfolio optimization," *Phys. Rev. Appl.*, vol. 15, no. 1, Jan. 2021, Art. no. 014012, doi: 10.1103/PhysRevApplied.15.014012.
[8] J. Marshall, D. Venturelli, I. Hen, and E. G. Rieffel, "Power of pausing: Advancing understanding of thermalization in experimental quantum annealers," *Phys. Rev. Appl.*, vol. 11, no. 4, Apr. 2019, Art. no. 044083, doi: 10.1103/PhysRevApplied.11.044083.
[9] M. Zielewski, M. Agung, R. Egawa, and H. Takizawa, "Improving quantum annealing performance on embedded problems," *Supercomputing Frontiers Innov.*, vol. 7, no. 4, pp. 32–48, 2020. [Online]. Available: https://superfri.org/superfri/article/view/332
[10] M. R. Zielewski and H. Takizawa, "A method for reducing time-to-solution in quantum annealing through pausing," in *Proc. Int. Conf. High Perform. Comput. Asia–Pacific Region*, New York, NY, USA, Jan. 2022, pp. 137–145, doi: 10.1145/3492805.3492815.
[11] Z. Gonzalez Izquierdo, S. Grabbe, S. Hadfield, J. Marshall, Z. Wang, and E. Rieffel, "Ferromagnetically shifting the power of pausing," *Phys. Rev. Appl.*, vol. 15, no. 4, Apr. 2021, Art. no. 044013, doi: 10.1103/PhysRevApplied.15.044013.
[12] Z. Gonzalez Izquierdo, S. Grabbe, H. Idris, Z. Wang, J. Marshall, and E. Rieffel, "Advantage of pausing: Parameter setting for quantum annealers," *Phys. Rev. Appl.*, vol. 18, no. 5, Nov. 2022, Art. no. 054056, doi: 10.1103/PhysRevApplied.18.054056.
[13] H. Chen and D. A. Lidar, "Why and when pausing is beneficial in quantum annealing," *Phys. Rev. Appl.*, vol. 14, no. 1, Jul. 2020, Art. no. 014100, doi: 10.1103/PhysRevApplied.14.014100.
[14] S. Woo Shin, G. Smith, J. A. Smolin, and U. Vazirani, "How 'quantum' is the D-wave machine?" 2014, *arXiv:1401.7087*.
[15] T. Albash and J. Marshall, "Comparing relaxation mechanisms in quantum and classical transverse-field annealing," *Phys. Rev. Appl.*, vol. 15, no. 1, Jan. 2021, Art. no. 014029, doi: 10.1103/PhysRevApplied.15.014029.
[16] M. H. Amin, "Searching for quantum speedup in quasistatic quantum annealers," *Phys. Rev. A, Gen. Phys.*, vol. 92, no. 5, Nov. 2015, Art. no. 052323, doi: 10.1103/PhysRevA.92.052323.
[17] A. Lucas, "Ising formulations of many NP problems," *Frontiers Phys.*, vol. 2, p. 5, Jan. 2014. [Online]. Available: https://www.frontiersin.org/article/10.3389/fphy.2014.00005
[18] J. Cai, W. G. Macready, and A. Roy, "A practical heuristic for finding graph minors," 2014, *arXiv:1406.2741*.
[19] (2023). *D-Wave System Documentation*. [Online]. Available: https://docs.dwavesys.com/docs/latest/
[20] M. Tan and Q. Le, "EfficientNetv2: Smaller models and faster training," in *Proc. 38th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 139, M. Meila and T. Zhang, Eds., Jul. 2021, pp. 10096–10106. [Online]. Available: https://proceedings.mlr.press/v139/tan21a.html
[21] D. Sherrington and S. Kirkpatrick, "Solvable model of a spin-glass," *Phys. Rev. Lett.*, vol. 35, pp. 1792–1796, Dec. 1975, doi: 10.1103/PhysRevLett.35.1792.
[22] A. Douglass, A. D. King, and J. Raymond, "Constructing SAT filters with a quantum annealer," in *Theory and Applications of Satisfiability Testing—SAT 2015*, M. Heule and S. Weaver, Eds. Cham, Switzerland: Springer, 2015, pp. 104–120.
[23] *Dwave-System*. Accessed: Aug. 1, 2023. [Online]. Available: https://github.com/dwavesystems/dwave-system
[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Los Alamitos, CA, USA, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[25] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds., Jun. 2019, pp. 6105–6114. [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html
[26] C. McGeoch and P. Farré. (2022). *Advantage Processor Overview*. [Online]. Available: https://www.dwavesys.com/media/3xvdipcn/14-1058a-a_advantage_processor_overview.pdf
[27] Z. Morrell, M. Vuffray, A. Y. Lokhov, A. Bärtschi, T. Albash, and C. Coffrin, "Signatures of open and noisy quantum systems in single-qubit quantum annealing," *Phys. Rev. Appl.*, vol. 19, no. 3, Mar. 2023, Art. no. 034053, doi: 10.1103/PhysRevApplied.19.034053.



MICHAEL ZIELEWSKI received the B.S. degree in computer science from Louisiana State University, in 2017, and the M.S. degree in information science from Tohoku University, in 2020, where he is currently pursuing the Ph.D. degree in information science. His research interests include quantum annealing and applications of machine learning.

KEICHI TAKAHASHI (Member, IEEE) received the Ph.D. degree in computer science from Osaka University, in 2019. In 2018, he was a Visiting Scholar with the Oak Ridge National Laboratory. He was an Assistant Professor with the Nara Institute of Science and Technology, from 2019 to 2021. He is currently an Assistant Professor with the Cyberscience Center, Tohoku University. His research interests include high performance computing and parallel distributed computing.

YOICHI SHIMOMURA is currently a specially appointed Associate Professor with the Cyberscience Center, Tohoku University. He joined NEC Solution Innovator Ltd., in 2005, where he has been in his current position, since 2021. He specializes in program optimization of supercomputers.



HIROYUKI TAKIZAWA (Member, IEEE) received the B.E. degree in mechanical engineering and the M.S. and Ph.D. degrees in information science from Tohoku University, in 1995, 1997, and 1999, respectively. He is currently a Professor with the Cyberscience Center, Tohoku University. His research interests include high-performance computing systems and their applications. He is a member of the IEEE CS, ACM SIGHPC, IEICE, and IPSJ.