



Enhancing the Reach and Reliability of Quantum Annealers by Pruning Longer Chains

Ramin Ayanzadeh , *Member, IEEE*, and Moinuddin Qureshi , *Fellow, IEEE*

Abstract—Analog Quantum Computers (QCs), such as D-Wave’s *Quantum Annealers (QAs)* and QuEra’s neutral atom platform, rival their digital counterparts in computing power. Existing QAs boast over 5,700 qubits, but their single-instruction operation model prevents using SWAP operations for making physically distant qubits adjacent. Instead, QAs use an *embedding* process to chain multiple *physical qubits* together, representing a *program qubit* with higher connectivity and reducing effective QA capacity by up to 33x.

We observe that, post-embedding, nearly 25% of physical qubits remain unused, becoming trapped between chains. Additionally, we observe a “Power-Law” distribution in the chain lengths, where a few *dominant chains* possess significantly more qubits, thereby exerting a considerably more significant impact on both qubit utilization and isolation. Leveraging these insights, we propose *Skipper*, a software technique designed to enhance the capacity and fidelity of QAs by skipping dominant chains and substituting their program qubit with two measurement outcomes. Using a 5761-qubit QA, we observed that by skipping up to eleven chains, the capacity increased by up to 59% (avg 28%), and the error decreased by up to 44% (avg 33%).

Index Terms—Adiabatic Quantum Computing, Embedding, Power-Law, Quantum Annealers.

I. INTRODUCTION

QUANTUM computers (QCs) harness the power of quantum bits (qubits) to solve problems that surpass the capabilities of classical computing [8]. Two main types of QCs exist: digital machines, exemplified by IBM, Google, and IonQ, and analog devices such as superconducting *Quantum Annealers (QAs)* by D-Wave, as well as neutral atom platforms by QuEra and PASQAL [1], [3], [4], [8].

While both digital and analog QCs have polynomial equivalent computing power and are accessed via the cloud, their operation models and design trade-offs differ significantly. In digital QCs (a.k.a. gate-based or circuit model QCs), qubits undergo a scheduled sequence of quantum operations defined by the quantum algorithm to directly manipulate their states [8]. Conversely, analog QCs operate as single-instruction systems, where the qubit environment is incrementally modified based on the evolution of a physical system, called “Hamiltonian”, thereby allowing natural qubit evolution and indirect state alteration [1], [3]. Unlike classical realm, both types offer equivalent polynomial computational power [1], [8].

Recent QAs have over 5,700 qubits, outscaling IBM’s Osprey QC with 433 qubits, but their single-instruction model limits effective program qubit handling [3]. Full connectivity of qubits at scale is infeasible. In digital QCs, compilers introduce successive SWAP operations to make physical qubits adjacent, allowing every program qubit to be represented by one physical qubit. Conversely, analog QCs cannot apply

Ramin Ayanzadeh is with the School of Computer Science, Georgia Institute of Technology, Atlanta, GA (e-mail: ayanzadeh@gatech.edu). He was supported by the NSF/CRA Computing Innovation Fellows program.

Moinuddin Qureshi is with the School of Computer Science, Georgia Institute of Technology, Atlanta, GA (e-mail: moin@gatech.edu).

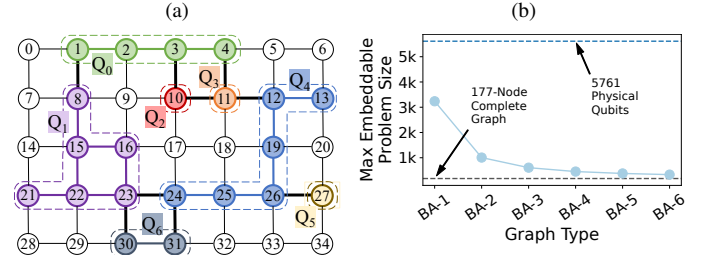


Fig. 1. (a) Embedding seven program qubits (Q_i) onto a 5×7 grid of physical qubits utilizes twenty qubits, leaving fifteen qubits unutilized. (b) Max embeddable Barabasi–Albert (BA) graphs on a 5761-qubit QA device for different preferential attachment factors (m), ranging from sparse BA-1 ($m = 1$) to dense BA-6 ($m = 6$) structures.

operations to qubits, thus preventing the use of SWAPs for qubit routing. Instead, QAs employ *embedding* where multiple physical qubits are *chained* (or entangled) to represent a program qubit with higher connectivity, as shown in Fig. 1(a) [3].

Compiling quantum circuits in digital QCs preserves qubit utilization (using 1-to-1 mapping between program and physical qubits); however, embedding in QAs can substantially increase physical qubit utilization [3] (due to chaining). For instance, the 5761-qubit QA can accommodate up to 177 program qubits with all-to-all connectivity, highlighting nearly 33x reduced *logical capacity*.

Real-world applications typically involve irregular “Power-Law” graphs [2], [7], and Barabasi–Albert (BA) graphs are widely considered representative of such real-world graphs [2], [5], [7]. Fig. 1(b) illustrates the largest embeddable BA graphs on a 5761-qubit QA for different preferential attachment factors (m), ranging from sparse BA-1 ($m = 1$) to dense BA-6 ($m = 6$) structures. As m increases linearly, the logical capacity reduces superpolynomially, converging to the 177-node fully connected graph.

Not all chains are created equal. We observe that chain lengths follow a “Power-Law” distribution, where a few *dominant chains* are significantly longer than most other chains. Furthermore, as shown in Fig. 1(a), we observe that a significant number of physical qubits remain unused as they become trapped in chains.

In this study, we aim to improve the capacity and fidelity of QAs through eliminating dominant chains, as they account for a substantial portion of qubit utilization and are the main reason for isolating physical qubits. We propose *Skipper*, which *prunes* these chains by removing their corresponding program qubits and replacing them with two possible measurement outcomes: -1 and $+1$. Each chain cut bifurcates the search space of the initial problem; hence c cuts create 2^c disjoint sub-spaces. *Skipper* examines all of these subspaces for a guaranteed full recovery.

II. BACKGROUND AND MOTIVATION

A. Quantum Computers: Digital vs. Analog

QCs fall into two categories: digital and analog. Digital QCs, such as IBM and Google's, use precise quantum operations to manipulate qubits [8]. Conversely, analog QCs, exemplified by D-Wave and QuEra, adjust the environment continuously, guiding qubits along specified paths [1], [3].

B. Quantum Annealers

Quantum Annealers (QAs) are a form of analog QCs that can sample from the ground state (the configuration with the lowest energy value) of a physical system, called Hamiltonian [1], [4]. QAs by D-Wave are single-instruction optimization accelerators that can only sample from the ground state of the following problem Hamiltonian (or Ising model):

$$\mathcal{H}_p := \sum_i \mathbf{h}_i \mathbf{z}_i + \sum_{I \neq j} J_{ij} \mathbf{z}_i \mathbf{z}_j \quad (1)$$

acting on spin variables $\mathbf{z}_i \in -1, +1$, where $\mathbf{h}_i \in \mathbb{R}$ and $J_{ij} \in \mathbb{R}$ are linear and quadratic coefficients, respectively [3].

C. Operation Model of Single-Instruction QAs

QAs operate as single-instruction computers, and during each execution trial, they only draw a single sample to approximate the global minimum of (1). Therefore, we *cast* real-world problems into Hamiltonians, where \mathbf{h} and J are defined in such a way that its global minimum represents the optimal solution to the problem at hand [1], [3]. The abstract problem Hamiltonian is then *embedded* into the connectivity map of the QA hardware to generate an executable Quantum Machine Instruction (QMI) [6]. Casting and embedding in QAs are akin to designing and compiling quantum circuits in digital QCs, respectively (Figure 2). The QMI is executed for several trials, and the outcome with the lowest objective value is deemed as the ultimate result [3].

D. Embedding for QAs

The connectivity of QA qubits is sparse, thereby limiting users to only specify J_{ij} for those qubits that are physically connected. Thus, as shown in Fig. 3, the abstract problem Hamiltonian is *embedded* into QA hardware where a program qubit (Q_i) with higher connectivity is represented by multiple physical qubits (q_i) called *chain*. Satisfying the following conditions is sufficient to guarantee that both the abstract Hamiltonian and the Hamiltonian executed on the QA hardware have identical ground states:

- 1) All sub-graphs representing program qubits must be a connected component.
- 2) There must be at least one connection between chains whose corresponding program qubits are connected.

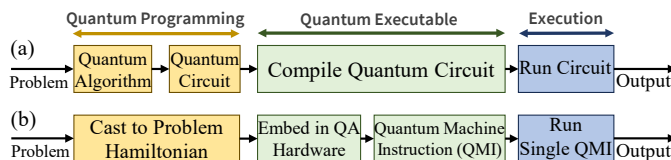


Fig. 2. Operation models: (a) digital QCs vs. (b) analog QAs.

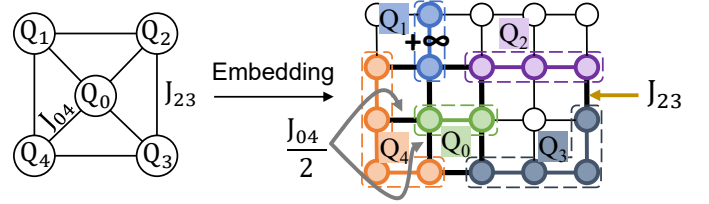


Fig. 3. Embedding example.

- 3) The quadratic coefficient J_{ij} is distributed equally among the couplers connecting Q_i and Q_j .
- 4) The linear coefficient \mathbf{h}_i is distributed equally among all physical qubits of the corresponding chain.
- 5) Inter-chain quadratic coefficients must be large enough to guarantee that all qubits within a chain take an identical value—i.e., a very high penalty for broken chains.

E. Prior Work Limitations

Previous work for solving larger problems on smaller QAs employ iterative schemes involving approximations [9], leading to reduced reliability as problem size increases. Conversely, Skipper explores the entire search space without resorting to approximations. Circuit cutting techniques [10] are infeasible in the analog quantum realm because: (a) the executable in QAs is not a quantum circuit, and (b) partitioning graphs by edge/node removal is nontrivial (e.g., a fully connected graph is non-partitionable).

F. Goal of This Paper

In Figure 4(a), we can see the maximum and average chain lengths for various graph topologies when embedded on a 5761-qubit QA, indicating that a few long chains, known as *dominant chains*, contain more than 7.9x the qubits compared to the average chain lengths. Furthermore, as shown in Figure 3, we observe that a significant number of physical qubits remain unused as they become trapped in chains. Figure 4(b) displays the number of unused qubits when embedding the largest possible graphs on a 5761-qubit QA for different graph topologies, indicating that more than 25% of physical qubits remain unutilized, primarily due to dominant chains. This underutilization of QA qubits, along with utilizing several physical qubits to represent a single program qubit, severely diminishes the capacity of QAs by up to 33x. This paper studies whether the reach and fidelity of QAs can be improved by pruning dominant chains.

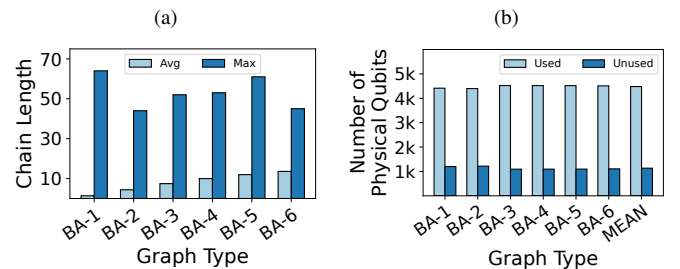


Fig. 4. Maximum embeddable BA graphs on 5761-qubit QA: (a) Avg and Max chain lengths, and (b) Number of unutilized qubits.

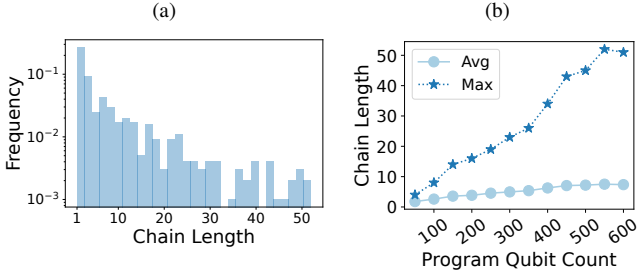


Fig. 5. (a) Chain lengths Histogram of a 600-node BA-3 graph (log-scale). (b) Max/Avg chain lengths of BA-3 graphs on a 5761-qubit QA.

III. SKIPPER: SKIPPING DOMINANT CHAINS

A. Key Insights: Not All Program Qubits are Equal

Figure 5(a) displays a log-scaled histogram of chain lengths for the BA-3 graph on a 5761-qubit QA, showing a “Power-Law” distribution with some notably longer *dominant chains* and many shorter chains. Figure 5(b) shows maximum and average chain lengths in BA-3 graphs as node count increases, highlighting growing chain length variability with larger problem sizes. These intriguing observations extend beyond the BA-3 graph type, and we observe it in all benchmark graphs.

B. Overview of Skipper

Leveraging the Power-Law distribution of chain lengths and qubit underutilization in QAs, we propose *Skipper* to enhance QA capacity and fidelity by pruning dominant chains. Figure 6 shows the overview of Skipper. For a given problem, Skipper prunes the top c longest (dominant) chains. Eliminating each dominant chain accomplishes two significant objectives: firstly, it frees up physical qubits previously used within pruned chains, and secondly, it eliminates the isolation of solitary qubits resulting from dominant chains. As a result, Skipper enables the handling of larger problems by accommodating a significantly higher number of program qubits. Additionally, Skipper enhances QA fidelity by substantially mitigating the impact of dominant chains, a primary factor in compromising QA reliability.

C. How to Skip Chains?

Skipping a chain in QAs is akin to freezing a qubit in digital QCs [2]. Fig. 7 shows how eliminating a chain from a five-variable problem yields two *independent* sub-problems. Skipping involves replacing the program qubit with +1 and -1, effectively removing the node and its edges from the graph. Unlike digital QCs, where removing one program qubit results in reducing the physical qubit utilization by one, in QAs, removing one program qubit liberates all the physical qubits involved in its corresponding chain.

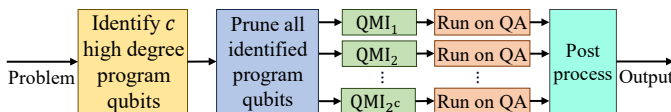


Fig. 6. Overview of Skipper.

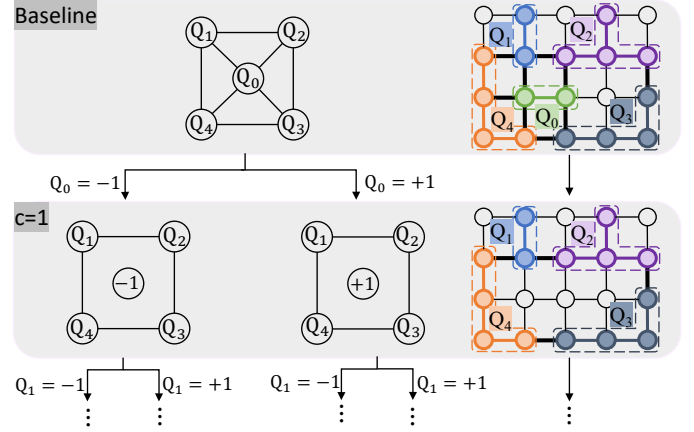


Fig. 7. Substituting Q_0 with ± 1 in a baseline of five spin variables generates two sub-problems, each with four spin variables ($c = 1$). The same embedding is applied to all 2^c sub-problems at each c -th level in the binary tree.

D. Skip Count: A Cost-Performance Tradeoff

Skipping c chains leads to 2^c sub-problems. Skipper runs all the corresponding QMIs for exact solution recovery, resulting in an exponential quantum overhead. By default, up to eleven chains are pruned. However, the nontrivial embedding and the necessity to run 2^c embeddings can pose a bottleneck for Skipper. Fortunately, the identical structure of all sub-problems at the c -th level in the binary tree allows for sharing the same embedding across them, as shown in Fig. 7. Note that in Skipper c does not scale with problem size, and Skipper always skips up-to eleven chains.

E. Decoding Outcomes

The input problem Hamiltonian comprises n variables. Skipper fixes c variables ($n \rightarrow n - c$ variables), and embedding represents each program qubit with multiple physical qubits ($n - c \rightarrow N$ variables), where $n \ll N$.

Skipper employs the *majority vote* scheme to *unembed* and retrieve the value of program qubits ($N \rightarrow n - c$ length bitstring). The values of the c pruned program qubits are then reinstated ($n - c \rightarrow n$ length bitstring).

F. Deriving the Final Output

In Skipper, all 2^c sub-problems are executed independently, each one corresponding to a separate sub-space of the primary problem. Consequently, in Skipper, the sample with the lowest energy or objective value is deemed as the ultimate output, with the originating sub-space of this global optimum being of no consequence.

G. Overhead of Skipper

Let c be the number of pruned chains, e denote the edges in the problem graph, r symbolize the number of trials on the QA, while n and N correspond to the number of program and physical qubits, respectively. Skipper supports up to eleven cuts, requiring at most 2048 independent quantum executables. All sub-problems use one embedding, maintaining constant complexity ($O(1)$). With $c \ll n \ll N \ll r$, Skipper’s time complexity is $O(2^c (rN + c))$, and memory usage scales as $O(rN2^c)$. Notably, c is independent of problem size, with a maximum of eleven chain prunes.

IV. SKIPPER EVALUATIONS

A. Methodology

Hardware & Software Platform—For our evaluations, we used the 5,761-qubit D-Wave Advantage System with a 20-microsecond annealing time, following the device-specific recommended anneal schedule. Each problem was run for 4,000 trials. We employed the *minorminer* tool [6] to discover embeddings for arbitrary problem Hamiltonians on D-Wave QA working graph.

Benchmarking—We evaluate Skipper using Power-Law graphs generated by the Barabási–Albert (BA) algorithm [5] with different preferential attachment factor values: $m = 1$ to 6, denoted as BA-1 to BA-6. These graphs represent most real-world applications [7], spanning from sparse (BA-1) to highly connected (BA-6) topologies. Edge weights are assigned randomly using a standard normal distribution, a common practice in benchmarking QAs [3], [4].

Figure of merit—We use the *Energy Residual (ER)* to assess the reliability of QA as

$$\text{Energy Residual (ER)} = |E_{\min} - E_{\text{global}}|, \quad (2)$$

where E_{global} represents the global minimum of the benchmark problem, and E_{\min} corresponds to the best solution obtained by the QA. A lower ER is desired. We used the state-of-the-art MQC technique [4] to approximate the global optimum of the benchmarks.

B. Results

1) *Impact on Capacity of QAs*: Figure 8(a) demonstrates that Skipper reduces underutilization of QA qubits by up to 57% (average 22.14%) with up to eleven trimmed chains.

We define the *Embedding Factor (EF)* of QAs as the ratio of program qubit count to physical qubit count, denoted as $EF = \frac{\text{Program Qubit Count}}{\text{Physical Qubit Count}}$. A value of $EF = 0$ indicates a failed embedding, while $EF = 1$ reflects a successful representation of every program qubit by a physical qubit (higher is desired). The capacity of QAs to handle specific graph types (BA-1 to BA-6) is measured by the maximum attainable EF value. In Skipper, Fig. 8(a) shows that QA capacity improves with increasing c across various graph topologies. Figure 8(b) demonstrates that Skipper enables the embedding of larger problems onto current QAs, with an increase of up to 59.61% (average 28.26%). It is important to note that this growth in the number of program qubits necessitates a substantial increase in the number of physical qubits, as one program qubit is represented by multiple physical qubits.

2) *Boosting QA Reliability*: In addition to addressing larger problem sizes on existing QA architectures, Skipper can be employed to enhance the reliability of currently executable quantum programs. Figure 9(a) shows that increasing the number of skipped chains in Skipper reduces the Energy Residual (ER), indicating a progressive approach towards the global optimum. Moreover, Figure 9(b) shows a remarkable maximum reduction of 44.4% (average 33.08%) in the gap between the global optimum and the best solution achieved by QAs using Skipper, when up to five chains are pruned, compared to the baseline.

Skipper’s performance remains consistent regardless of the increasing density of problem graphs (from BA2 to BA6).

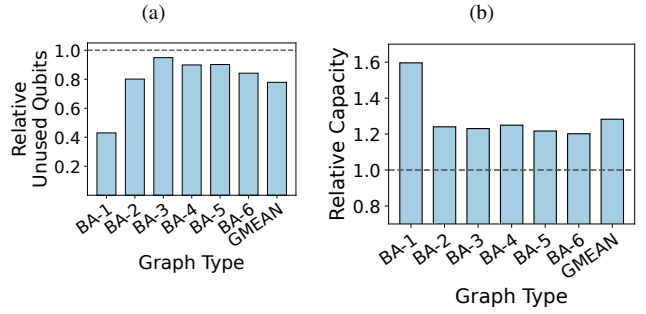


Fig. 8. (a) Relative Number of Unused Physical Qubits in Skipper for up to 11 Chain Cuts, Compared to the Baseline. Lower is better. (b) Relative QA capacity in Skipper compared to baseline. Higher is better.

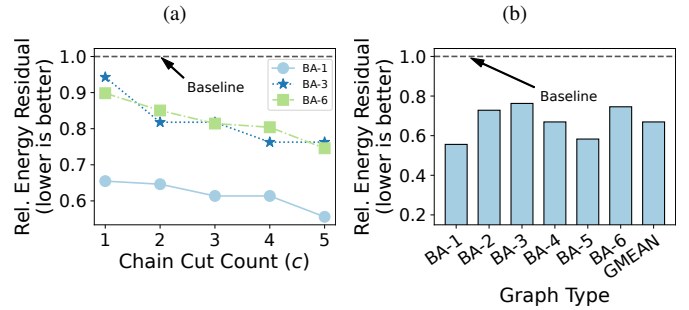


Fig. 9. Relative Energy Residual (ER) in Skipper compared to baseline (lower is better). (a) Relative ER for different graphs as c increases. (b) Overall relative ER for up to five chain cuts.

V. CONCLUSION

We introduce *Skipper*, a software scheme designed to improve the capacity and fidelity of QAs. By observing that chain lengths in QAs follow a “Power-Law” distribution, Skipper strategically prunes these dominant chains. This is achieved by replacing their corresponding program qubits with two potential measurement outcomes, resulting in the liberation of all qubits involved in the dominant chains, as well as freeing an additional 25% of isolated qubits previously trapped in chains. Using a 5761-qubit QA, Skipper managed to tackle up to 59% larger problems (avg 28%) and reduced the error by up to 44% (avg 33%) when pruning up to eleven dominant chains.

REFERENCES

- [1] T. Albash and D. A. Lidar, “Adiabatic quantum computation,” *Reviews of Modern Physics*, 2018.
- [2] R. Ayanzadeh, N. Alavisamani, P. Das, and M. Qureshi, “Frozenqubits: Boosting fidelity of qaoa by skipping hotspot nodes,” in *ASPLOS-2023*, 2023.
- [3] R. Ayanzadeh, P. Das, S. Tannu, and M. Qureshi, “Equal: Improving the fidelity of quantum annealers by injecting controlled perturbations,” in *QCE-2022*, 2022.
- [4] R. Ayanzadeh, J. Dorband, M. Halem, and T. Finin, “Multi-qubit correction for quantum annealers,” *Scientific Reports*, 2021.
- [5] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *science*, 1999.
- [6] J. Cai, W. G. Macready, and A. Roy, “A practical heuristic for finding graph minors,” *arXiv:1406.2741*, 2014.
- [7] A. Clauset, E. Tucker, and M. Sainz, “The colorado index of complex networks,” *Retrieved July*, 2016.
- [8] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
- [9] S. Okada, M. Ohzeki, M. Terabe, and S. Taguchi, “Improving solutions by embedding larger subproblems in a d-wave quantum annealer,” *Scientific reports*, 2019.
- [10] W. Tang, T. Tomesh, M. Suchara, J. Larson, and M. Martonosi, “Cutqc: using small quantum computers for large quantum circuit evaluations,” in *ASPLOS 2021*, 2021.