Multi-tasking through Quantum Annealing

Jargalsaikhan Artag[1] and Shirakashi Jun-ichi[1]
[1]Department of Electrical Engineering and Computer Science,
Tokyo University of Agriculture and Technology,
2-24-16 Nakachou, Tokyo,1848588, Japan.
Tel.: 813887919.

Abstract

Quantum annealing efficiently solves combinatorial optimization problems using an adiabatic system. In this system, the system's Hamiltonian evolves from a general state to a problem-specific state. The problem-specific Hamiltonian can consist of multiple components, allowing multi-tasking through quantum annealing (MTQA). In this scenario, the adiabatic quantum system naturally performs parallel computations through parallel embedding on the hardware. This study explored the potential of parallel processing in quantum systems using MTQA, which solves multiple NP-hard problems in parallel to fully utilize quantum hardware resources (e.g., qubits and couplers). By focusing on two distinct problems, we evaluated the multi-tasking efficiency of quantum annealing. Our results demonstrate that MTQA achieves task solution quality comparable to traditional single-problem quantum annealing approaches.

1 Introduction

Quantum annealing has emerged as a powerful approach for solving complex combinatorial optimization problems by leveraging the unique properties of quantum mechanics to efficiently explore vast solution spaces [1, 2]. This technique is based on the principle of adiabatic quantum computation, which encodes the solution to a problem in the ground state of a time-dependent quantum Hamiltonian [1, 2, 3]. Initially, the system is set in the simple ground state of the initial Hamiltonian, evolving slowly to the ground state of a problem-specific Hamiltonian that encodes the desired optimization problem [4]. The Hamiltonian that governs the quantum annealing process comprises two main components: the initial Hamiltonian and the problem Hamiltonian. This can be expressed as follows:

$$\mathrm{H}(s) = -\frac{A(s)}{2}\left(\sum_i \sigma_i^x\right) + \frac{B(s)}{2}\left(\sum_i h_i \sigma_z^i + \sum_{i>j} J_{i,j}\sigma_z^i\sigma_z^j\right) \quad (1)$$

In this equation, the Pauli-X and Pauli-Z operators on the i-th qubit are denoted by σ_x^i and σ_z^i respectively. Time-dependent coefficients $A(s)$ and $B(s)$ control the evolution of the Hamiltonian from the initial non-interacting state towards the state that encodes the optimization problem. Initially, $A(s)$ is dominant, placing the system in a superposition of all the possible states, which is essential for exploring the solution space. As the process progresses, $B(s)$ becomes dominant, steering the system towards a problem-specific configuration that represents the optimal solution [5, 6].

Quantum annealing has been implemented on platforms such as D-Wave Systems, which employ superconducting circuits to execute the transverse field Ising model [7]. This model is specifically tailored to tackle problems presented as quadratic unconstrained binary optimization (QUBO) challenges [8]. A notable advancement in this domain is the concept of Parallel Quantum Annealing (PQA), which involves simultaneously embedding and solving multiple problem instances on quantum hardware. PQA has laid the foundation for further innovations in quantum annealing, proving that parallel task execution can enhance the effectiveness and utilization of quantum resources [9, 10, 11, 12, 13, 14].

Building on the foundation established by PQA, this study introduces Multi-tasking through Quantum Annealing (MTQA). Unlike classical computers, where multi-tasking involves parallel task execution to enhance efficiency, MTQA leverages the intrinsic parallelism of quantum systems to simultaneously process multiple optimization problems without the communication bottlenecks found in classical systems [14, 15, 16]. Previous studies have indicated a slight decrease in the solution accuracy when solving problems in parallel using PQA [9]. However, our findings show that MTQA achieves task solution quality comparable to traditional single-problem quantum annealing approaches. This approach utilizes quantum resources more effectively, potentially enhancing the performance of the quantum annealers.

In this study, we explore the practicality and benefits of MTQA by focusing on two NP-hard problems: the Minimum Vertex Cover Problem (MVCP) [17, 18] and the Graph Partitioning Problem (GPP) [18, 19, 20, 21]. We employ an advanced approach to embed these problems into quantum hardware, allowing for the parallel solving of multiple problems to fully utilize quantum hardware resources such as qubits and couplers. Our implementation involves careful consideration of embedding strategies, chain strength calculations based on chain length, and scaling of QUBO to fit the hardware bias for individual instances. This study aims to demonstrate the multi-tasking capabilities of quantum annealing, gauge the quality of solutions, and compare these with the outcomes from traditional sequential quantum annealing and classical optimization solvers such as ILOG CPLEX [22] and GUROBI [23].

2 Results

Parallel embedding

The parallel embedding of multiple problem instances on a D-Wave quantum annealer [24, 25] is the cornerstone of MTQA. This technique enables simultaneous processing of distinct problems, thus optimizing the use of quantum resources. Iterative embedding strategies [26, 27], as shown in Fig 1 (b), embed problems closely on the quantum annealer hardware, which can result in the dense configurations depicted in Fig 1 (a). Such configurations often lead to longer chain lengths because adjacent qubits may already be allocated to neighboring problems. To prevent this issue, we introduced an isolation layer to the embedded problems, ensuring that no two problems are directly adjacent, thus preventing interference and optimizing problem isolation, as shown in Fig 1 (c).

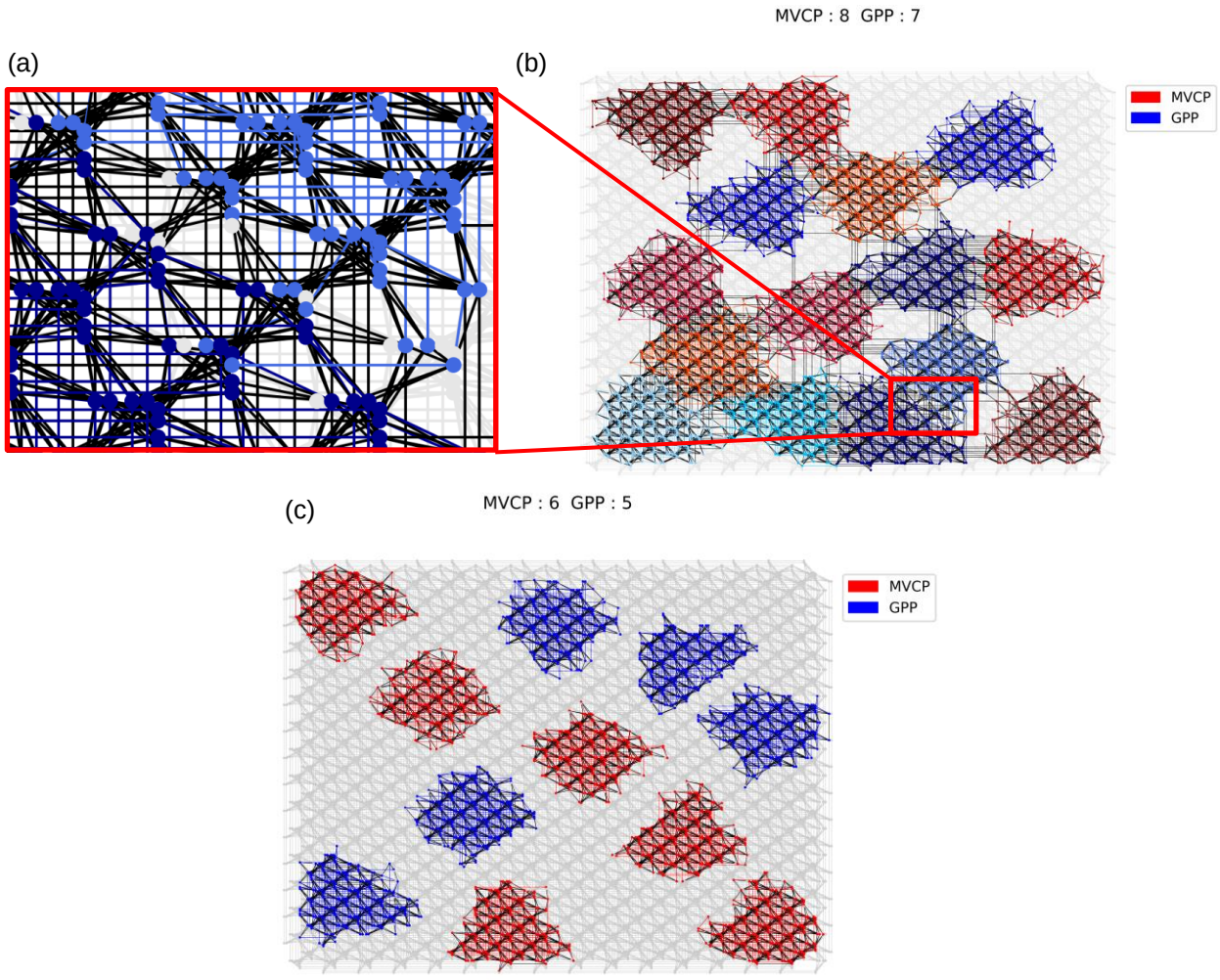

Fig. 1 Parallel embedding in the D-Wave Pegasus topology. (a) A dense embedding without isolation, where problems are tightly embedded on the hardware. It shows a congested qubit environment where adjacent qubits support distinct tasks. (b) An iterative embedding of the MVCP and GPP, maximizing hardware utilization. We embedded MVCP 8 times and the GPP 7 times with this method. For clarity, we used slightly different colors. (c) Parallel embedding with neighbor isolation embedding strategy, which separates different problem instances on the

hardware. We embedded MVCP 6 times and the GPP 5 times with this method.

The chain length, which represents the number of physical qubits mapped to a single logical qubit, is influenced by the size and connectivity demands of the problem [24, 25, 26, 27]. As these parameters increase, so does the chain length, which affects the number of problem instances that can be concurrently embedded [28, 29]. This relationship is shown in detail in Fig 2, where we examine the scalability of our embedding strategies.

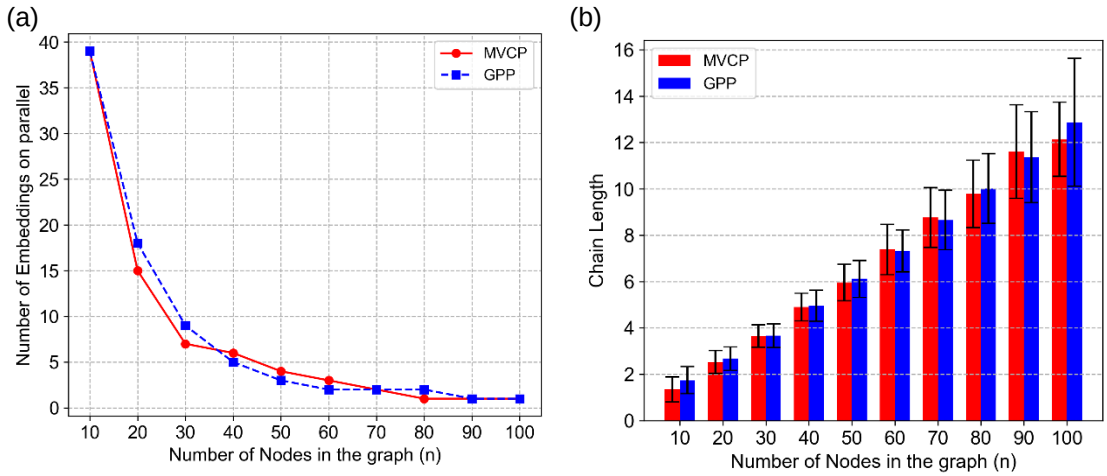

Fig. 2 (a) Number of problem instances embedded in parallel as a function of graph size, highlighting the direct impact of increasing problem complexity on embedding capacity. (b) Chain length variations as a function of graph size, demonstrating how embedding complexity escalates with larger problem dimensions. These illustrations provide crucial insights into the scalability of MTQA and the challenges associated with embedding larger problem instances.

Eigenspectrum Analysis

We performed an eigenspectrum [30, 31, 32, 33, 34] analysis to deepen our understanding of the dynamic behavior of quantum systems under MTQA [35, 36, 37, 38]. This analysis is essential for assessing how the energy levels vary during the annealing process and involves calculating both the ground state and the first excited state energies using the Linear Algebra Package (LAPACK) [39, 40]. This process allows us to examine the evolution of the system's state within its Hilbert space [41], which is a complex vector space that provides a complete description of the system's quantum states [42]. We examined three scenarios: a single problem instance, two identical parallel problems, and two different parallel problems. We incorporated embedding techniques, chain strength calculations, and scaling adjustments into the problem Hamiltonians to accurately reflect their real-world implementation in quantum annealers.

1. Single Problem (MVCP and GPP): For a single problem, such as MVCP or GPP, we calculated the ground state energy $E_{0,\text{single}}$ and the first excited state energy $E_{1,\text{single}}$ are calculated, along with the energy gap, $\Delta E_{\text{single}} = E_{1,\text{single}} - E_{0,\text{single}}$. The results are presented in Fig 3 (a) and (b).

2. Two Identical Problems in Parallel (MVCP): When solving two identical instances of MVCP in parallel, the combined Hamiltonian is constructed in a block diagonal form, effectively doubling the Hamiltonian matrix dimensions without introducing interactions between the two instances. The Hamiltonian is given by:

$$H_{parallel} = \begin{bmatrix} H_{MVCP} & 0 \\ 0 & H_{MVCP} \end{bmatrix} \quad (2)$$

This configuration doubles the ground state energy $E_{0,parallel} = 2E_{0,MVCP}$ and adjusts the first excited state energy $E_{1,parallel} = 2E_{0,MVCP} + \Delta E_{MVCP}$ as shown in Fig 3 (c). Thus, the energy gap remains the same as that for a single problem $\Delta E_{parallel} = \Delta E_{MVCP}$.

3. Two Different Problems in Parallel (MVCP and GPP): To solve two different problems, MVCP and GPP, the combined Hamiltonian is also constructed in a block diagonal form. This construction ensures that the Hamiltonian of each problem affects only its corresponding part of the system without interaction. The Hamiltonian is represented as

$$H_{combined} = \begin{bmatrix} H_{MVCP} & 0 \\ 0 & H_{GPP} \end{bmatrix} \quad (3)$$

The total ground state energy for this system is the sum of the individual ground states $E_{0,combined} = E_{0,MVCP} + E_{0,GPP}$, and the first excited state energy, as illustrated in Fig 3 (d), is $E_{1,combined} = E_{0,MVCP} + E_{0,GPP} + \Delta E_{min}$. Where ΔE_{min} is the smaller of the individual energy gaps, $\Delta E_{min} = min(\Delta E_{MVCP}, \Delta E_{GPP})$.

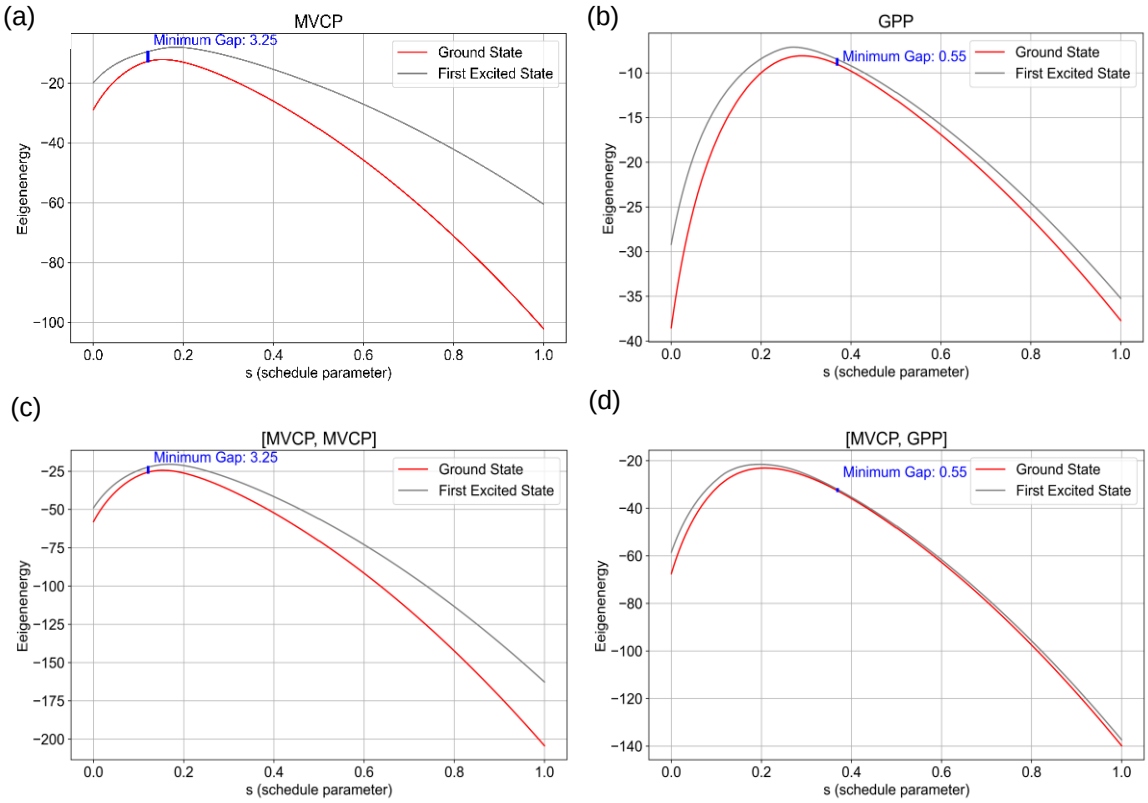

Fig. 3 Eigenspectrum Analysis of Parallel Problem Solving in Quantum Annealing (a) Eigenspectrum of a single MVCP problem, illustrating the ground and first excited state energies. (b) Eigenspectrum of a single GPP problem. (c) Eigenspectrum of two identical MVCP problems in parallel, showing maintained energy gap consistency. (d) Eigenspectrum of MVCP and GPP problems in parallel, detailing the additive nature of energy states and minimal gaps.

Throughout the annealing process, we continuously calculated the eigenvalues of the system Hamiltonian to track the evolution of energy levels. This comprehensive analysis enabled us to monitor the stability of quantum state transitions and evaluate the likelihood of the system maintaining the lowest possible energy state, which is essential for optimal problem-solving.

Our analysis suggests that combining multiple QUBOs (or Hamiltonians) into a single Hamiltonian through multi-tasking does not increase the computational complexity of solving the entire Hamiltonian. The system maintains stable quantum state transitions, enabling effective attainment of the ground state through quantum annealing. Specifically, for two identical problems, the energy gap remained consistent with a single problem, indicating no additional difficulty in preserving the ground state during the annealing process. For two different problems, the energy gap is determined by the smaller of the individual gaps; however, the overall ground-state energy is simply the sum of the individual ground-state energies. This implies that the multi-tasking approach efficiently applies quantum annealing to address multiple problems simultaneously without intensifying the overall computational challenge. The stability of quantum state transitions under multi-tasking conditions underscores the robustness of this approach in maintaining ground state fidelity, which is crucial for obtaining optimal solutions to embedded problems.

Solution Quality

In this study, we measured the efficiency of MTQA by focusing on two specific NP-hard problems: MVCP and GPP. QA successfully solved MVCP but struggled with GPP for graph sizes exceeding 20 nodes. To comprehensively evaluate the MTQA's performance, we examined metrics such as Time to Solution (TTS) and Ground State Probability (GSP), focusing on MVCP, and the number of cut edges for the GPP.

Time to Solution (TTS) metrics

The Time-to-Solution (TTS) metric is a performance indicator used to evaluate the efficiency of a quantum annealer in locating the optimal solution to a problem [43, 44, 45]. It provides an estimate of the expected time required to find the ground state (optimal solution) with a certain probability. TTS can be expressed mathematically as the ratio of the time taken for one run t_{run} to the probability of finding the optimal solution. The TTS can be expressed as:

$$TTS = t_{run}\frac{log(1 - P_{success})}{log(1 - p)} \tag{4}$$

where $P_{success}$ is the desired success probability (often set as 0.99). This metric helps compare the efficiency of different algorithms and hardware configurations in solving optimization problems. For the MTQA case, TTS is adapted to account for running multiple problems in parallel. Given t_{run} as the time required for one run, we must consider that multiple problems are embedded and solved in parallel. The TTS for the same problem embedded multiple times in parallel is calculated as follows:

$$t_{run} = \frac{1}{A}\left(\frac{T_{QPU}}{n} + \sum_i^n U_i\right) \tag{5}$$

Where A is number of samplings, T_{QPU} is the D-Wave QPU-Access-Time, U_i is the CPU processing time used to unembed solutions for i-th instance, and n is the number of the same problem instances embedded in parallel. The average probability of finding the optimal solution in the parallel embedded same problem, p_{avg}, is given by

$$p_{avg} = \frac{1}{n}\sum_i^n p_i \tag{6}$$

where, p_i is the probability of determining the optimal solution for the i-th problem from n instances. p_{avg} expresses the GSP metrics. The evolved TTS for MTQA, considering the average probability p_{avg}, can be expressed as

$$TTS_{MTQA} = t_{run}\frac{log(1 - P_{success})}{log(1 - p_{avg})} \tag{7}$$

This formulation accounts for the parallel solving of multiple problems and incorporates the time required for both quantum processing and classical post-processing, thereby providing a comprehensive metric for evaluating the performance of MTQA.

Our analysis revealed that in MVCP, MTQA markedly outperformed traditional sequential Quantum Annealing in terms of TTS across all tested graph sizes, as shown in Fig 2 (a). As the size of the graph increases, MTQA becomes equivalent to QA. In addition, our result in Fig 2 (b) shows that GSP for MVCP does not change even if we solve the same problem multiple times.

Number of Cut Edges in GPP:
In the context of GPP, our study compared the number of cut edges resolved by MTQA and sequential QA with the optimal solutions found by classical solvers. As shown in Table 1, MTQA consistently produced solutions with cut edges that were very close to or matched the optimal solutions. In contrast, sequential QA demonstrated a slight but noticeable deviation from optimal solutions. This result underscores MTQA's effectiveness in accurately solving complex optimization problems due to the increased number of samplings generated by parallel embedding. Furthermore, as illustrated in Fig 4 (c), the solution energies obtained from QA and MTQA are approximately identical, suggesting that MTQA performs comparably to QA in terms of the solution quality.

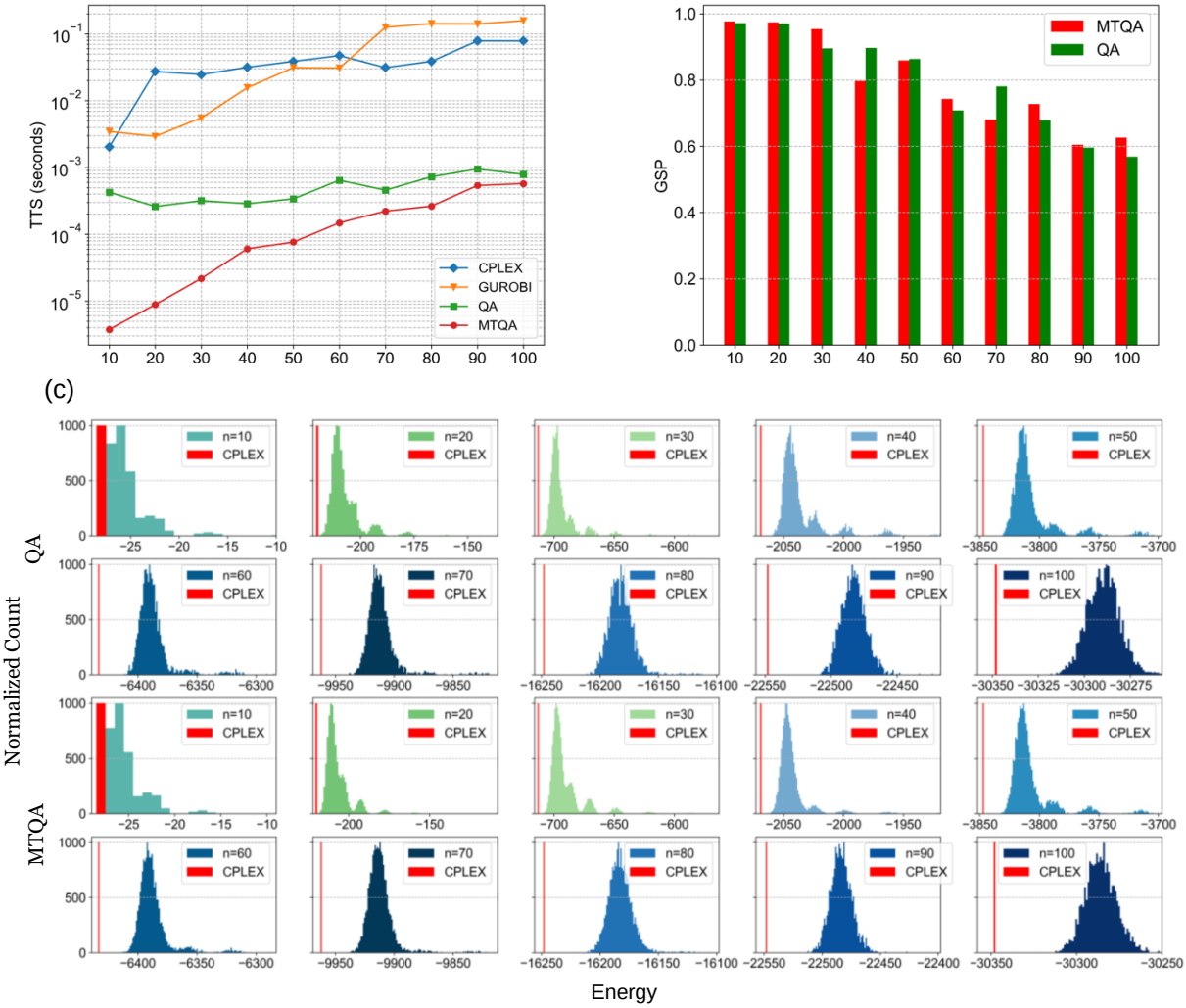

Fig4. (a) TTS metrics for MVCP, graph size increases from 10 nodes to 100 nodes. (b) GSP metrics for MVCP, graph size increases from 10 nodes to 100 nodes. (c) GPP solution energy to normalized counts: due to the different number of parallel embeddings on the hardware, we obtained different numbers of optimized solutions from quantum annealer.

Table1 | GPP number of cut edges for the four different solvers. The results indicate that MTQA generally outperforms QA, particularly for problems involving fewer than 100 nodes. For instances with 60 to 80 nodes, two parallel embeddings were utilized, whereas for instances with 90 and 100 nodes, only one embedding was employed. These findings further suggest that MTQA is more effective than QA in solving these problems.

n	CPLEX	GUROBI	MTQA	QA
10	22	22	22	22
20	80	80	80	81
30	187	187	189	190
40	331	331	337	341
50	528	528	542	545
60	767	767	786	789
70	1063	1063	1086	1087
80	1352	1352	1388	1389
90	1752	1752	1788	1792
100	2152	2152	2188	2182

The experimental data demonstrate that MTQA significantly enhances the computational efficiency and solution accuracy for solving NP-hard problems, particularly in comparison to sequential QA. By fully utilizing the available qubits and enabling the simultaneous processing of multiple problems, MTQA not only maximizes the quantum annealer's computational resources but also presents a scalable and efficient approach for tackling complex optimization challenges.

3. Discussion

Our exploration of the application of MTQA has revealed nuanced insights into its performance, particularly when handling NP-hard problems such as MVCP and GPP. These findings significantly enhance our understanding of the potential of quantum computing and the operational nuances of MTQA.

Strategic Utilization and Efficiency: This study underscores MTQA's enhanced efficiency over traditional QA for certain problem sizes, particularly evident in TTS metrics for MVCP. The potential of MTQA to process multiple problem instances simultaneously demonstrates its capability to leverage quantum hardware more effectively, resulting in higher computational throughput. Additionally, the ability of some applications to decompose large problems into smaller parts and solve them in parallel [11] without distorting solution quality is crucial for fully realizing MTQA's potential. This approach ensures that MTQA can be effectively applied without quality degradation, even as the complexity of the problem increases.

Scaling Challenges with Larger Graphs: However, as the graph size increases, the advantages of MTQA become constrained by the physical limitations of current quantum annealing technology, particularly the finite number of qubits and couplers [46, 47, 48]. This limitation is critical as it restricts our ability to embed large-scale problems multiple times, which is a fundamental aspect of MTQA. This limitation highlights a crucial bottleneck—while MTQA is theoretically capable of enhancing computational efficiency, in practice, its scalability is curtailed by hardware constraints.

GPP Considerations: For GPP, where QA struggles to find optimized solutions for larger graphs, the application of MTQA offers a strategic advantage by potentially improving solution quality through parallel processing. However, the effectiveness of this approach in handling larger graph sizes requires careful consideration because of the increased resource competition and complexities involved in embedding these larger problems effectively on the available quantum hardware.

Practical Implications and Future Prospects: The results of applying MTQA to both MVCP and GPP have practical implications for industries where rapid and accurate solutions to optimization problems are critical. Despite its current limitations with larger graphs, MTQA's ability to maintain high-quality solutions for moderately sized problems suggests that it can still serve as a valuable tool in quantum computing applications. The insights gained from our study also point towards the need for advancements in quantum annealing technology, including the development of hardware with more qubits and enhanced connectivity to better accommodate the demands of larger and more complex problem instances.

The investigation of MTQA reveals its potential in effectively utilizing quantum annealing to solve optimization problems while emphasizing the need for technological advancements to overcome current limitations. Future research should concentrate on improving the scalability of MTQA, examining hybrid (quantum-classical) methods, and developing more sophisticated algorithms that optimize the embedding and annealing processes. As quantum annealing technology advances, it is predicted that MTQA will play an increasingly significant role in bridging the gap between theoretical quantum advantages and their practical applications in solving some of the most difficult computational problems.

4. Methods

Problem Selection and Formulation
We generated test graphs using the Erdős-Rényi [49, 50, 51] model with nodes ranging from 10 to 100 and an edge probability of 0.9. This model was chosen for its ability to produce dense graphs, presenting significant challenges for optimization and providing a robust testbed for evaluating the effectiveness of MTQA against sequential QA methods.

Minimum Vertex Cover Problem (MVCP):
MVCP seeks to identify the smallest subset of vertices in a graph such that every edge has at least one endpoint in this subset. This problem is well-suited to be formulated as QUBO problem, enabling its solution through quantum annealing techniques [17, 18]. The QUBO

formulation for MVCP is expressed as:

$$H_{MVCP} = A \sum_{(i,j)\in E} (1 - x_i)(1 - x_j) + B \sum_{i \in V} x_i \quad (8)$$

where x_i is a binary variable representing whether vertex i is included in the vertex cover, V is the set of vertices, E is the set of edges. Parameter A serves as a penalty to ensure that all edges are covered, while B penalizes the inclusion of each vertex, promoting a minimal vertex cover. To ensure that the edge coverage is prioritized over vertex inclusion, the parameters are chosen such that $0 < B < A$. Typically, we set $B = 1$ and $A = 2$ to balance these objectives effectively. This formulation guarantees that minimizing H_{MVCP} is equivalent to solving the MVCP [18].

Graph Partitioning Problem (GPP):
GPP partitions the vertices of a graph into two subsets of equal size, minimizing the number of edges between these subsets [18, 19, 20, 21]. This problem can also be effectively represented as a QUBO problem, which makes it suitable for quantum annealing. The QUBO formulation for GPP is expressed as

$$H_{GPP} = A\left(\sum_{i \in V} x_i - \frac{|V|}{2}\right)^2 + B \sum_{(u,v)\in E} (x_u + x_v - 2x_u x_v) \quad (9)$$

Here, x_i is a binary variable indicating the subset assignment of vertex i, and $x_{u,v}$ is a binary variable indicating edges u, v. A is a penalty parameter for enforcing equal subset sizes, and $|V|$ is the total number of vertices. The first term ensures that the partitioning results in subsets of equal size, while the second term aims to minimize the number of edges between different subsets. To determine a rather simple lower bound on A, we set $B = 1$, and calculated as follows:

$$A \geq B\frac{min(2\Delta, |V|)}{8} \quad (10)$$

where, Δ denotes the maximum degree of the graph. This choice ensures a balanced emphasis between penalizing unequal partition sizes and minimizing inter-subset edges [18].

Quantum Annealing Hardware and Experimental Parameters
This study investigates multi-tasking in solving NP-hard problems using the D-Wave Advantage 6.4 quantum annealer, which features a Pegasus graph structure with 5612 working qubits [24]. This hardware facilitates the embedding of complex problem graphs, which is necessary for accurately solving complex problems. For the experimental parameters we set the annealing time to 10μs, the number of reads to 2500 and the other parameters are default, and we used D-Wave annealer which we accessed through D-Wave Leap.

MTQA Implementation
Embedding Strategy: For each problem instance, embeddings were found using the minorminer [26] tool from D-Wave's Ocean SDK. Distinct embeddings were identified for each problem to ensure parallel processing without a qubit overlap. A neighbor isolation strategy was employed for each embedding.

Chain Strength Calculation: Chain strength is a critical parameter in quantum annealing, ensuring that the physical qubits representing the same logical qubit remain aligned throughout the annealing process [28, 29]. We calculated the chain strength for each instance to embed in hardware because chain strength is calculated by the chain length, which is different for every embedding on the hardware. We used two methods for calculating the chain strength, chosen based on the characteristics of the problems and empirically determined prefactors:

1. **Uniform Torque Compensation, UTC**: This method balances the magnetic torque experienced by qubits in a chain. It is particularly effective for problems such as MVCP, in which maintaining alignment despite varying interactions is crucial. We applied a prefactor of 0.5, determined empirically, to balance chain strengths.

2. **Scaled**: This method involves scaling the chain strength according to the specific requirements of the problem. For GPP, where a balance between different parts of the problem is essential, we used a prefactor of 1.5, which was also determined empirically.

This approach ensures that the qubits maintain their correct state throughout the annealing process, thereby reflecting the complexity and connectivity of the problem.

Scaling of QUBO: The QUBO models were scaled to fit the dynamic range of the D-Wave annealer, ensuring accurate processing within the hardware constraints on the interaction strengths and qubit biases [28, 29]. Each embedding QUBOs were scaled individually. Scaling was carefully managed to avoid issues with small problem instances being scaled disproportionately, ensuring that each problem's energy landscape is appropriately represented on the hardware.

Unembedding Strategy: Unembedding is the process of translating the solutions from physical qubits back to logical problem variables [28, 29, 48]. We unembedded each individual embedding solution. We employed several unembedding methods from the D-Wave Ocean SDK to achieve a binary solution:

1. **Majority Vote**: This method selects the most common value among the physical qubits that represent a single logical qubit. This is useful for problems where robustness to noise is critical.

2. **Weighted Random**: This method assigns weights to each possible solution based on their energies and selects among them probabilistically. We used this method for MVCP because it balances solution quality with diversity, which is essential for covering all edges effectively.

3. **Minimize Energy**: This method selects the solution with the lowest energy. It is highly effective for problems in which the exact minimization of the objective function is crucial. We applied this method to GPP because of its requirement for precise partitioning to minimize cut edges.

By carefully selecting these methods based on the characteristics of each problem, we ensured that the unembedding process was tailored to maximize the performance and accuracy of the solutions obtained from the quantum annealer.

Classical Solvers
To benchmark MTQA's performance, we utilized the classical optimization solvers, ILOG CPLEX and GUROBI. These solvers are known for their efficiency in solving NP-hard problems and served as a baseline for evaluating the outcomes of quantum annealing solutions. By comparing the results of the MTQA with those obtained from classical solvers, we were able to assess the relative advantages of quantum annealing in terms of solution quality and computational efficiency.

5. Code Availability

The code for performing the MTQA can be accessed via this link
https://github.com/shrakashlab/MTQA

6. References

1. Farhi, E., Goldstone, J., Gutmann, S. & Sipser, M. Quantum computation by adiabatic evolution. arXiv preprint quant-ph/0001106 (2000).
2. Morita, S. & Nishimori, H. Mathematical foundation of quantum annealing. Journal of Mathematical Physics 49, 125210 (2008).
3. Albash, T. & Lidar, D. A. Adiabatic Quantum Computation. Reviews of Modern Physics 90, (2018).
4. Kadowaki, T. & Nishimori, H. Quantum annealing in the transverse Ising model. Physical Review E 58, 5355 (1998).
5. Hauke, P., Katzgraber, H., Lechner, W., Nishimori, H. & Oliver, W. Perspectives of Quantum annealing: Methods and Implementations. Zenodo (CERN European Organization for Nuclear Research) (2020). https://doi.org/10.5281/zenodo.4436323
6. Das, A. & Chakrabarti, B. K. Quantum annealing and related optimization methods. vol. 679 (Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008).

7. Johnson, M. W. et al. Quantum annealing with manufactured spins. Nature 473, 194–198 (2011).
8. Boixo, S. Evidence for quantum annealing with more than one hundred qubits. Nature Physics 10, 218–224 (2014).
9. Pelofske, E., Hahn, G. & Djidjev, Hristo N. Parallel quantum annealing. Scientific Reports 12, 4499 (2022).
10. Artag, J., Shimada, M. & Shirakashi, J. Parallel Quantum Annealing: A Novel Approach to Solving Multiple NP-Hard Problems Concurrently. 2023 IEEE International Conference on Quantum Computing and Engineering (QCE) 02, 328–329 (2023).
11. Pelofske, E., Hahn, G. & Djidjev, Hristo N. Solving larger maximum clique problems using parallel quantum annealing. Quantum Information Processing 22, 219 (2023).
12. Pelofske, E., Bärtschi, A. & Eidenbenz, S. Short-depth QAOA circuits and quantum annealing on higher-order Ising models. npj Quantum Information 10, 30 (2024).
13. Jałowiecki, K., Więckowski, A., Gawron, P. & Gardas, B. Parallel in time dynamics with quantum annealers. Scientific Reports 10, 13534 (2020).
14. Huang, T., Zhu, Y., Ryan & Luo, T. When quantum annealing meets multitasking: Potentials, challenges and opportunities. Array 17, 100282 (2023).
15. Blake, G., Dreslinski, R. & Mudge, T. A survey of multicore processors. IEEE Signal Processing Magazine 26, 26–37 (2009).
16. Hennessy, J. L. & Patterson, D. A. Computer architecture: a quantitative approach. (Morgan Kaufmann, 2011).
17. Karp, R. M. Reducibility Among Combinatorial Problems. Springer Berlin Heidelberg 219–241 (2010). https://doi.org/10.1007/978-3-540-68279-0_8
18. Lucas, A. Ising formulations of many NP problems. Frontiers in Physics 2, (2014).
19. Ushijima-Mwesigwa, H., Christian & Mniszewski, S. M. Graph partitioning using quantum annealing on the D-Wave System. PMES (2017).
20. Fu, Y. & Anderson, P. J. Application of statistical mechanics to NP-complete problems in combinatorial optimisation. Journal of Physics A: Mathematical and General 19, 1605–1620 (1986).
21. Mézard, M. & Montanari, A. Information, Physics, and Computation. Oxford University Press (2009). https://doi.org/10.1093/acprof:oso/9780198570837.001.0001
22. Nickel, S., Steinhardt, C., Schlenker, H. & Burkart, W. Decision Optimization with IBM ILOG CPLEX Optimization Studio: A Hands-On Introduction to Modeling with the Optimization Programming Language (OPL). Springer Nature, (2022). https://doi.org/10.1007/978-3-662-65481-1.
23. Bob Bixby. The gurobi optimizer. Transp. Research Part B, 41(2), 159–178, (2007).
24. D-Wave Systems Inc. (2024). Technical Description of the QPU — D-Wave System Documentation documentation. docs.dwavesys.com https://docs.dwavesys.com/docs/latest/doc_qpu.html.
25. Boothby, K., Bunyk, P., Raymond, J. & Roy, A. Next-Generation Topology of D-Wave Quantum Processors. ArXiv (Cornell University) (2020). https://doi.org/10.48550/arxiv.2003.00133.
26. Cai, J., Macready, W. G. & Roy, A. A Practical Heuristic for Finding Graph Minors. (2014). https://doi.org/10.48550/arxiv.1406.2741.
27. Djidjev, Hristo N. Logical qubit implementation for quantum annealing: augmented Lagrangian approach. Quantum Science and Technology 8, 035013 (2023).
28. Choi, V. Minor-embedding in Adiabatic Quantum computation: I. The Parameter Setting Problem. Quantum Information Processing 7, 193–209 (2008).
29. Choi, V. Minor-embedding in Adiabatic Quantum computation: II. Minor-universal Graph Design. Quantum Information Processing 10, 343–353 (2010).
30. Lanting, T. et al. Entanglement in a Quantum Annealing Processor. Physical Review X 4, (2014).
31. Susa, Y. & Nishimori, H. Performance Enhancement of Quantum Annealing under the Lechner–Hauke–Zoller Scheme by Non-linear Driving of the Constraint Term. Journal of the Physical Society of Japan 89, 044006–044006 (2020).
32. Susa, Y. & Nishimori, H. Variational optimization of the quantum annealing schedule for the Lechner-Hauke-Zoller scheme. Physical Review A 103, 022619 (2021).
33. Copenhaver, J., Wasserman, A. & Wehefritz-Kaufmann, B. Using quantum annealers to calculate ground state properties of molecules. The Journal of Chemical Physics 154, 034105 (2021).
34. Crosson, E., Farhi, E., Lin, C.-Y.-Y., Lin, H.-Y. & Shor, P. Different strategies for optimization using the quantum adiabatic algorithm. arXiv preprint arXiv:1401.7320 (2014).
35. Dickson, N. G. et al. Thermally assisted quantum annealing of a 16-qubit problem. Nature Communications 4, 1903 (2013).

36. Harris, R. et al. Phase transitions in a programmable quantum spin glass simulator. Science 361, 162–165 (2018).
37. Perdomo-Ortiz, A., Dickson, N., Drew-Brook, M., Rose, G. & Aspuru-Guzik, A. Finding low-energy conformations of lattice protein models by quantum annealing. Scientific Reports 2, 571 (2012).
38. Boixo, S., Albash, T., Spedalieri, F., Chancellor, N. & Lidar, D. Experimental signature of programmable quantum annealing. Nature Communications 4, (2012).
39. Harris, C. R. et al. Array Programming with NumPy. Nature 585, 357–362 (2020).
40. Anderson, E., et al.: LAPACK Users' Guide, 3rd edn. SIAM, Philadelphia (1999).
41. Halmos, P. R. A Hilbert Space Problem Book. Springer Science & Business Media vol. 19 (2012).
42. Santhanam, T. S. & Santhanam, B. The discrete Fourier transform and the quantum-mechanical oscillator in a finite-dimensional Hilbert space. Journal of Physics A: Mathematical and Theoretical 42, 205303 (2009).
43. Michael Ryan Zielewski & Takizawa, H. A Method for Reducing Time-to-Solution in Quantum Annealing Through Pausing. in International Conference on High Performance Computing in Asia-Pacific Region 137–145 (Association for Computing Machinery, 2022). https://doi.org/10.1145/3492805.3492815.
44. King, J., Sheir Yarkoni, Nevisi, M. M., Hilton, J. P. & McGeoch, C. C. Benchmarking a quantum annealing processor with the time-to-target metric. (2015). https://doi.org/10.48550/arxiv.1508.05087.
45. Hamerly, R. et al. Experimental investigation of performance differences between coherent Ising machines and a quantum annealer. Science Advances 5, (2019).
46. Venturelli, D., J. J. Marchand, D. & Rojo, G. Quantum Annealing Implementation of Job-Shop Scheduling. (2015). https://doi.org/10.48550/arxiv.1506.08479.
47. King, A. D. et al. Observation of topological phenomena in a programmable lattice of 1,800 qubits. Nature 560, 456–460 (2018)
48. Vinci, W., Albash, T., Paz-Silva, G., Hen, I. & Lidar, D. A. Quantum annealing correction with minor embedding. Physical Review A 92, 042310 (2015).
49. Bollobás, B. Random Graphs. Springer New York NY 215–252 (1998). https://doi.org/10.1007/978-1-4612-0619-4_7.
50. Durrett, R. Erdös–Rényi random graphs. in 27–69 (Cambridge University Press, 2006).
51. Hagberg, A., Swart, P. J. & Schult, D. A. Exploring network structure, dynamics, and function using NetworkX. https://www.osti.gov/biblio/960616 (2008).

7. Competing interests

The authors declare no competing interests.

8. Author contributions

A.J. conducted the full study, including the conceptualization, run experiment, data analysis, and manuscript writing. SH.J. provided supervision.