Multi-Class Support Vector Machines with Multi-Tasking Quantum Annealing

1st Jargalsaikhan Artag  
*Tokyo University of Agriculture and Technology*Tokyo 184-8588, Japan  
s234051x@st.go.tuat.ac.jp  
2nd Moe Shimada  
*Tokyo University of Agriculture and Technology*Tokyo 184-8588, Japan  
s217094x@st.go.tuat.ac.jp  
3rd Jun-ichi Shirakashi  
*Tokyo University of Agriculture and Technology*Tokyo 184-8588, Japan  
shrakash@cc.tuat.ac.jp

*Abstract*— This study explores the application of quantum annealing (QA) and Multi-tasking Quantum Annealing (MTQA) for classification tasks, comparing these quantum computing approaches with traditional methods such as Support Vector Machines (SVC) and Simulated Annealing (SA). The research utilizes the D-Wave Advantage 6.4 system across three benchmark datasets, including subsets of handwritten digits and the iris dataset with different feature focuses. While SVC and SA generally provide robust performance, quantum approaches demonstrate competitive but slightly varied results. Specifically, sequential QA and its parallel counterpart, MTQA, show promise in their ability to handle complex computational tasks concurrently. However, their performance is influenced by the current limitations of quantum technology and the empirical selection of model parameters. The findings suggest that with advancements in quantum algorithm optimization and more sophisticated parameter tuning, the efficacy of QA and MTQA could enhance, potentially surpassing conventional methods. This study underscores the nascent potential of quantum computing in machine learning and highlights the need for continued development to fully realize its capabilities in practical applications.

Keywords—Quantum annealing, PQA, Combinatorial problems for real-world application

# Introduction

Quantum annealing, a computational method leveraging quantum mechanics, offers an effective approach for solving NP-hard optimization problems. Nonetheless, optimal utilization of available qubits, particularly when handling multiple problems concurrently, remains a challenge [1, 4, 5]. In the existing applications of PQA, this method has been employed to address a single problem multiple times or different problems with minor embedded combinatorial optimization [1]. This study introduces a specialized application of PQA to concurrently solve multiple distinct NP-hard problems. We focus on three significant NP-hard problems relevant to integrated circuit design: the GCP, the MVCP, and the GPP [2]. Our refined approach to PQA potentially contributes to the ongoing evolution of quantum computing field by facilitating simultaneous problem-solving, thus augmenting efficiency and solution accuracy.

# Methodology

## Problem Selection and Description

The first step was to transform these problems into a form that could be solved on a quantum annealer. This involved formulating each problem as a QUBO problem. The process of transforming these problems into QUBO form involves representing the problem variables as binary variables and the problem constraints and objective function as a quadratic function.

**GCP**: The QUBO formulation for the GCP is designed to ensure that each node in the graph is assigned a unique color and that no two adjacent nodes share the same color [6-8].

**MVCP**: The QUBO formulation for the MVCP problem is designed to minimize the sum of the binary variables associated with the nodes in the vertex cover [8].

**GPP**: The QUBO formulation for the GPP problem is designed to partition the graph into two sets of nodes of equal size, with the number of edges between the sets minimized [8].

## Parallel Quantum Annealing (PQA) Approach

Our methodology utilizes PQA for simultaneous resolution of multiple NP-hard problems by efficiently exploiting the D-Wave's Pegasus architecture. Initially, the problems are transformed into QUBO problems and subsequently embedded into the D-Wave Advantage 6.2 quantum annealing machine using the “minorminer” [10] method. To ensure distinct problem allocation, each problem is assigned a unique set of qubits on the machine until full utilization.

Following the embedding, a combined QUBO is formed that consists of individual QUBOs of each problem in their embedding order. The uniqueness of variables in the combined QUBO is maintained by initializing the variables of the succeeding problem from n+1, where n is the number of embedded variables in the preceding QUBO problem. The distinct variables guarantee problem independence, thereby enabling representation on non-overlapping subgraphs of the Pegasus architecture.

The quantum annealing process is then executed on the combined QUBO, enhancing qubit utilization and solution accuracy. After annealing, the results are decoded into their respective problems for further evaluation.

# Results

## Overview

Our research demonstrates a notable increase in Time-To-Target (TTT) [1, 3] performance when compared to traditional QA and SA. Our refined application of PQA consistently outperformed QA and was either comparable to or superior to SA in most cases. Notably, the PQA method was used to calculate the solutions of the three problems concurrently, while the QA and SA solutions were calculated one by one. Additionally, We report the CPU process time for classical computation methods such as SA, providing further insight into efficiency comparisons.

## Problem Settings and Classical Algorithms

(c)

(a)

(b)

To evaluate the performance of our PQA method, we established five distinct problem sets, each with varying numbers of nodes and utilizing a specific sparse graph with the density of 0.4. Classical algorithms acted as benchmarks for each problem in the sets. For the GCP, we employed the Welsh Powell algorithm [6], whereas the MVCP employed the Greedy algorithm [8], and the GPP utilized the Kernighan Lin Bisection algorithm [9]. These algorithms provided optimal solutions to compare with the quantum solutions.

1. Train and test accuracy of datasets

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Data | Train Points | Test Points | SMO Accuracy | | SA Accuracy | | QA |
| Train | Test | Train | Test |  |
| Blob |  |  | 10 |  | 20 |  | 20 |
| Iris | 30 | 120 | 10 |  | 25 |  | 25 |
| Digit 0,1,2 |  |  | 12 |  | 30 |  | 30 |
| Digit 3,4,5 |  |  | 12 |  | 35 |  | 35 |
| Digit 6,7,8,9 |  |  | 14 |  | 40 |  | 40 |

## Time-To-Target (TTT) Measure

The Time-To-Target (TTT) measure is a critical performance metric for quantum annealers, assessing the time needed to reach a solution that's 99% optimal solution. This measure is important given the probabilistic nature of quantum annealing, which doesn't guarantee the best solution at every instance. From (1), one can see the ,

Here, A is the number of samples per annealing run, is the time of quantum processing unit spends on the problem, and is the classical computer's time spent on processing QUBOs and decoding results, K represents concurrently solved problems, and is the average probability of obtaining a solution 99% as good as the classical solution for all concurrent problems. is indicative of PQA's efficiency in simultaneous problem solving, enabling performance comparison between PQA and other methods. The results, as shown in Figs. 1 (a)-(c), PQA consistently outperforms both QA and SA in terms of TTT, with a noticeable edge in MVCP across all problem sets.

# Discussion and Future work

This study represents an extended exploration into the application of PQA for simultaneously addressing distinct NP-hard problems, especially related to integrated circuit design. The demonstrated enhancement in TTT performance over traditional QA and SA methods holds promise. However, broader application within the field of integrated circuits necessitates further rigorous exploration.

Future research endeavors should not only concentrate on assessing PQA's versatility across a wide range of problem domains but also particularly explore its adaptability and potential in integrated circuit design. This targeted approach could reveal more facets of PQA's ability to tackle intricate real-world problems within this specific field, further optimizing the utilization of quantum computing resources.

In summary, this investigation offers a nuanced application of PQA within the field of integrated circuit design, building on existing methods. While demonstrating promising results, it recognizes the need for further research and exploration to fully grasp the potential of PQA in this specific domain. The study serves as a step forward, inviting continued examination and refinement in the context of complex real-world problems.

##### References

1. E. Pelofske, H. Georg, and H. N. Djidjev, "Parallel quantum annealing," Sci Rep 12, 4499, 2022.
2. T. Lengauer, “Combinatorial algorithms for integrated circuit layout,” Springer Science & Business Media, 2012.
3. A. Yoshida, T. Miki, M. Shimada, Y. Yoneda, and J. Shirakashi, “Mimicking of thermal spin dynamics by controlling sparsity of interactions in Ising spin computing with digital logic circuits,” Appl. Phys. Express 15.6, 067002, 2022.
4. P. Ray, B. K. Chakrabarti, and Arunava Chakrabarti, “Sherrington-Kirkpatrick model in a transverse field: Absence of replica symmetry breaking due to quantum fluctuations,” Phys. Rev. B Condens Matter 39.16, 1989, pp. 11828–11832.
5. T. Kadowaki and H. Nishimori, “Quantum annealing in the transverse Ising model,” Phys. Rev. E 58.5, 1998, pp. 5355–5363.
6. D. J. Welsh and M. B. Powell, "An upper bound for the chromatic number of a graph and its application to timetabling problems," The Comput. J. 10.1, 1967, pp.85-86.
7. F. Glover, G. Kochenberger, and Y. Du, “Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models,” 4or, 17, 2019, pp. 335-371.
8. N. Biggs, E. K. Lloyd, and R. J. Wilson, “Graph Theory, ” 1736-1936. Oxford University Press, 1986.
9. T. Bui, C. Heigham, C. Jones, and T. Leighton, “Improving the performance of the Kernighan-Lin and simulated annealing graph bisection algorithms,” Proceedings of the 26th ACM/IEEE DAC, June 1989, pp.775-778.
10. J. Cai, W. G. Macready, and A. Roy, “A practical heuristic for finding graph minors,” arXiv preprint arXiv:1406.2741 2014.