Pauli-Correlation Encoding and Solving Multi-Dimension Knapsack Problems

Monit Sharma

March 19, 2025

Abstract

Combinatorial optimization problems are challenging for classical algorithms due to their complexity, with quantum computing offering promising alternatives through algorithms like the Quantum Approximate Optimization Algorithm (QAOA) and Quantum Random Access Optimization (QRAO). Despite their potential, quantum advantages are limited by hardware constraints, especially for large-scale problems. We enhance the Pauli Correlation Encoding (PCE) method by introducing a multi-phase re-optimization strategy that uses perturbations and local searches to improve solution quality and robustness. By leveraging the equivalence between Quadratic Unconstrained Binary Optimization (QUBO) and Max-Cut problems, we apply PCE to complex problems like the Multi-Dimensional Knapsack Problem (MDKP), achieving superior results. Our analysis shows that the improved PCE method requires fewer qubits than QRAO, supports deeper quantum circuits, and offers enhanced performance. This approach's application to real-world logistics and supply chain problems demonstrates its practical utility and potential impact, advancing the field of quantum-inspired optimization.

Contents

Chapter 1

Introduction

Combinatorial optimization problems involve identifying the best solution from a finite set of possibilities [1]. These problems are prevalent in various domains, including logistics, finance, telecommunications, and bioinformatics. Classic examples include the Traveling Salesman Problem (TSP) [2], knapsack problems [3], and scheduling tasks, where the objectives often focus on minimizing costs, maximizing profits, or optimizing resources within given constraints.

The real-world relevance of these problems is evident in their applications: optimizing delivery routes in logistics to reduce costs, enhancing portfolio management in finance for better returns, and improving network design in telecommunications for reliable data flow. However, as problem size and complexity grow, solving these problems becomes increasingly difficult, with traditional algorithms often struggling to find optimal solutions in a reasonable timeframe due to the vast number of possible configurations.

Significant advancements in quantum computation hardware over the past decade have spurred research into achieving practical quantum advantages in various fields [4]. Combinatorial optimization problems, in particular, have gained attention, although a definitive quantum advantage has yet to be demonstrated [5].

Quantum algorithms like the Quantum Approximate Optimization Algorithm (QAOA) [6] and Grover's search [7] have been developed to tackle combinatorial optimization problems. These algorithms aim to find near-optimal solutions with greater efficiency than classical methods, especially for problems where exact solutions are computationally intensive. As quantum hardware continues to improve, applying quantum computing to combinatorial optimization is becoming increasingly viable, with the potential to transform industries reliant on optimization.

While both algorithms apply to many combinatorial optimization problems, research has primarily focused on quadratic unconstrained binary optimization (QUBO). QUBO and Max-Cut are equivalent via a linear transformation (see Appendix 7.1). QUBO's prominence in quantum computing stems from its natural alignment with many quantum hardware platforms (particularly the Ising model), so problems of interest are often modeled as QUBO.

However, current quantum hardware imposes significant limitations on the size of QUBO problems that can be addressed. The treatable problem size is constrained by hardware capacity and problem density/complexity. Consequently, many problems must be simplified or scaled down to fit within these limitations, often at the expense of solution accuracy.

To fully leverage quantum computing's potential, it is crucial to develop algorithmic techniques that reduce problem size without sacrificing solution quality. Such approaches include preprocessing, decomposition, and hybrid classical–quantum strategies that partition large problems into smaller subproblems. Addressing these challenges can push the boundaries of quantum computing's practical applicability in large-scale combinatorial optimization.

A variety of qubit-efficient schemes have therefore been developed to optimize quantum resource usage [8–23]. Techniques like Quantum Random Access Optimization (QRAO) and Multi-Basis Encoding (MBE) achieve linear space compression by encoding multiple variables per qubit, though each approach has its own trade-offs in compression ratio and measurement complexity.

More aggressive exponential compression schemes exist but often sacrifice expressivity and can degrade solution quality. Partitioning techniques (e.g., circuit cutting) and Light Cone Cancellation (LCC) reduce resource demands but incur exponential classical overhead or become less effective with circuit depth.

In this work, we focus on a hybrid quantum-classical solver using Pauli Correlation Encoding (PCE) [24], which achieves polynomial compression while preserving classical intractability. We introduce a multi-phase re-optimization strategy to mitigate local minima and demonstrate PCE's advantages (in qubit efficiency and performance) over existing methods like QRAO, applying it to challenging instances of the multi-dimensional knapsack problem (MDKP) and real-world logistics optimization.

Chapter 2

Multi-Dimensional Knapsack Problems

Selecting appropriate problem instances for benchmarking quantum computing (QC) performance requires a careful balance. Instances must be challenging enough to surpass state-of-the-art classical solvers—justifying quantum approaches—yet small enough for near-term hardware. They should also remain model-independent to support standardized benchmarking frameworks [5].

This work examines both crafted and randomly generated MDKP instances representative of real-world scenarios. Real-world cases often appear in mixed-integer programming (MIP) benchmarks, but truly challenging and publicly available instances are rare. Crafted instances, by contrast, enable controlled evaluation of algorithmic progress.

Definition

Multi-Dimensional Knapsack Problem (MDKP). Given a set $I = \{1, \ldots, n\}$ of items each with profit p_i and resource consumption a_{ij} across m dimensions, and capacities b_j, the MDKP is

$$\max_{x\in\{0,1\}^n} \quad \sum_{i\in I} p_i x_i \tag{2.0.1}$$

$$\text{s.t.} \quad \sum_{i\in I} a_{ij} x_i \le b_j, \quad \forall j \in \{1, \ldots, m\}, \tag{2.0.2}$$

$$x_i \in \{0, 1\}, \quad \forall i \in I.$$

The objective (2.0.1) maximizes total profit while constraints (2.0.2) ensure resource usage does not exceed capacities. MDKP is NP-hard, with complexity driven by the profit-to-resource ratios p_i/a_{ij} and capacity sizes b_j. Small capacities or low consumption values permit dynamic programming or heuristics, but larger, denser instances rapidly become intractable for classical solvers [25, 26], motivating their use as benchmarks for quantum optimization.

Chapter 3

Pauli Correlation Encoding

Pauli Correlation Encoding (PCE) [24] is a qubit-efficient method for solving combinatorial optimization problems with $m = \mathcal{O}(n^k)$ binary variables using only n qubits. Rather than mapping each variable to a single Pauli-Z operator (as in QAOA), PCE encodes variables into k-body Pauli correlators across multiple qubits.

Definition

PCE Encoding. Let $\{\Pi_i\}_{i=1}^m$ be a selected subset of mutually commuting k-body Pauli strings (excluding the identity). A binary variable $x_i \in \{0, 1\}$ is encoded via

$$x_i = \operatorname{sgn}\Big(\langle \Psi | \Pi_i | \Psi \rangle\Big),$$

where $\langle \Pi_i \rangle = \langle \Psi | \Pi_i | \Psi \rangle$ is the expectation value of Π_i under the parameterized state $|\Psi(\theta)\rangle$.

Restricting to the union of $X^{\otimes k}, Y^{\otimes k}, Z^{\otimes k}$ correlators reduces the encoding capacity to

$$m \;=\; 3\binom{n}{k},$$

balancing expressivity and measurement feasibility.

PCE solves a weighted Max-Cut formulation (equivalent to any QUBO problem) by optimizing a variational state $|\Psi(\theta)\rangle$ to minimize the loss

$$\mathcal{L} = \sum_{(i,j)\in E} W_{ij} \tanh\Big(\alpha \langle \Pi_i \rangle\Big) \tanh\Big(\alpha \langle \Pi_j \rangle\Big) + \beta\, \nu \left[\frac{1}{m}\sum_{i=1}^{m} \tanh^2\Big(\alpha \langle \Pi_i \rangle\Big)\right]^2, \tag{3.0.1}$$

where W_{ij} are Max-Cut edge weights, $\alpha \approx n^{\lfloor k/2 \rfloor}$ scales correlators into the nonlinear regime, and $\nu = (w(G)/2 + w(T_{\min})/4)$ is the Poljak–Turzík lower bound on the cut value. After training, the final bitstring is extracted via the sign rule and refined through a classical local search before mapping back to the original MDKP solution.

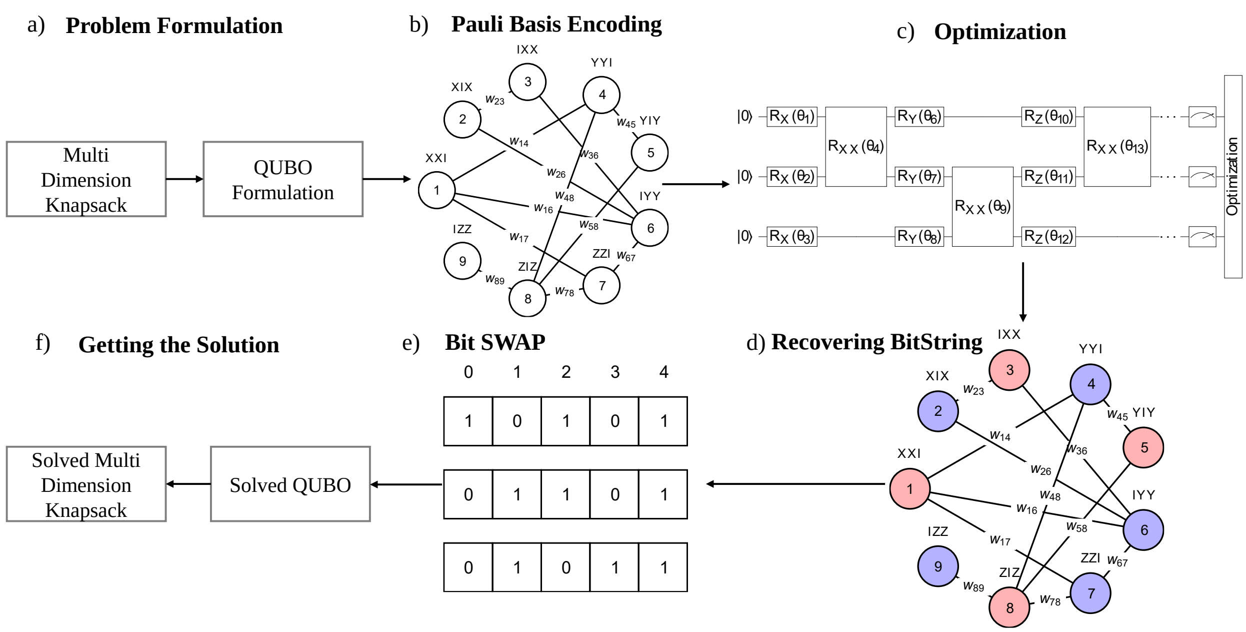

Figure 3.1: Pauli Correlation Encoding (PCE) for Multi-Dimensional Knapsack (MDK) Problems: **(a)** We begin with an instance of the MDK problem, characterized by a specific number of items and dimensions. The first step involves transforming the problem into a Quadratic Unconstrained Binary Optimization (QUBO) formulation, which is a prerequisite for execution on quantum computing platforms. The linear programming (LP) to QUBO conversion introduces additional slack variables, significantly increasing the problem's dimensionality. Following the methodology outlined in Barahona et al. [27], the QUBO problem, initially involving n variables, is further converted into a weighted Max-Cut problem on $n+1$ vertices. **(b)** In this subfigure, we illustrate the weighted Max-Cut graph, where each binary variable from the QUBO formulation is encoded using k-body Pauli correlators—specifically, $k = 2$ in this case. As depicted, node 1, representing the binary variable x_1, is encoded into the Pauli operator $X \otimes X \otimes I$, where the tensor product order corresponds to the respective qubits. This encoding achieves a quadratic compression of the solution space, reducing m variables to $n = \mathcal{O}(m^{1/2})$ qubits, such as compressing 9 binary variables into 3 qubits. More generally, k-body correlations enable polynomial compression of order k. The chosen Pauli set consists of three subsets of mutually commuting Pauli strings, which optimizes the measurement process by minimizing the number of required measurements. **(d)** In this step, the solution bitstring x is extracted from the Max-Cut solution by analyzing which nodes are in the cut and which are not. This solution is then further refined in **(e)**, where a classical bit-swap search around the solution x is conducted to identify potentially superior solutions x^* in the vicinity. **(f)** Finally, the improved Max-Cut solution is translated back into the QUBO framework and, subsequently, into the original MDK problem space, yielding the final results.

Chapter 4

Experiments

We generated four sets of MDKP instances varying items, dimensions, and variable counts (see table 4.1). For each instance set, we used a brickwork variational ansatz (illustrated in figure 4.2) whose gate count scales linearly with the number of encoded variables. To offset the qubit reduction, circuit depth was increased to roughly $10 \times n$ layers, where n is the number of qubits. All experiments used $k = 2$ Pauli encoding and gradient-based optimizers (SLSQP, ADAM, AMSGRAD) with default settings.

We compared PCE against QRAO only on Set 1; larger sets demanded too many qubits for QRAO simulations. QRAO employed settings from [28], excluding linear relaxation. figure 4.1 plots qubit requirements versus binary variable count, demonstrating PCE's superior compression.

All solutions were benchmarked against state-of-the-art classical solver outputs.

Table 4.1: Configuration of MDKP test instance sets. Bounds $[a, b]$ indicate ranges of randomized parameters. Qubit counts are shown for quadratic (k=2) and cubic (k=3) PCE compression, and QRAO $(3, 1, p)$.

Set	**Items**	**Dimensions**	**Variables**	**PCE Qubits (k=2)**	**PCE Qubits (k=3)**	**QRAO Qubits**
1	[10,20]	{2,3}	[30,50]	[5,7]	[5,6]	[10,30]
2	[50,100]	{2,3}	[80,130]	[8,10]	[7,8]	[40,70]
3	[120,500]	{2,3}	[140,550]	[11,20]	[8,12]	[70,300]
4	1000	3	1026	27	14	550

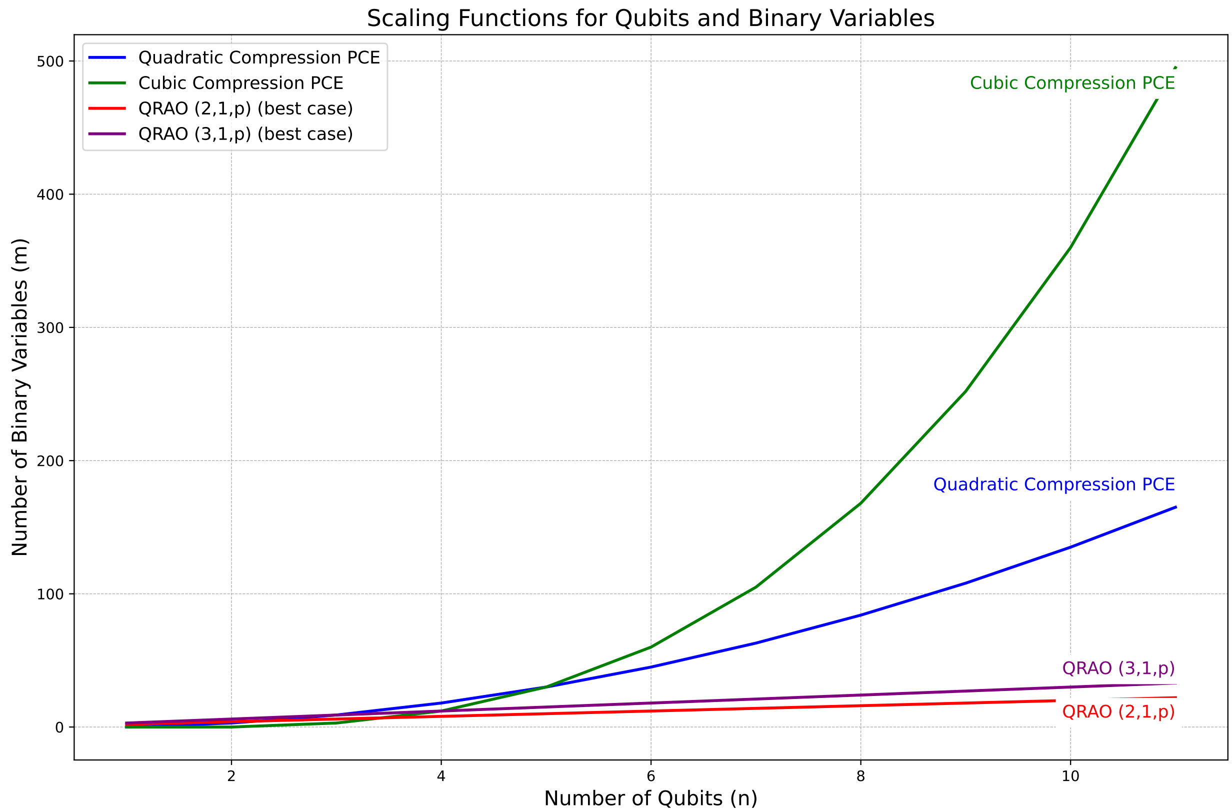

Figure 4.1: Comparison of qubit requirements versus binary variables for PCE (quadratic and cubic) and QRAO under optimal compression ratios.

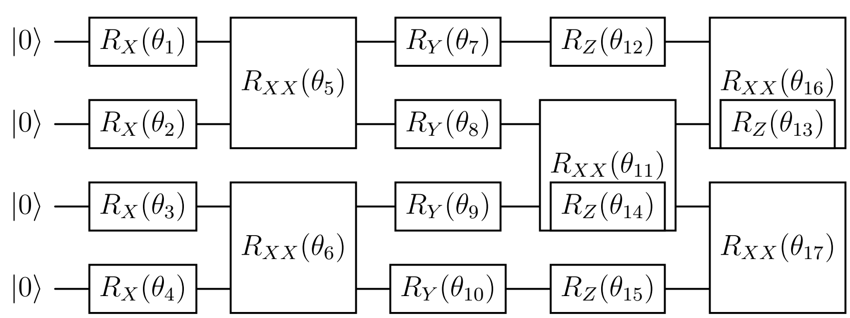

Figure 4.2: Brickwork variational ansatz comprising alternating single-qubit rotations (X, Y, Z) and two-qubit R_{XX} gates.

Chapter 5

Results

We first compared PCE (quadratic compression) against QRAO$(3, 1, p)$ on Set 1, finding comparable solution quality but significantly fewer qubits required by PCE (see table 4.1). QRAO became infeasible for Sets 2 and 3 due to excessive qubit and circuit depth requirements. For Set 4 (over 1000 variables), PCE solved the single instance optimally.

Table 5.1: Comparison of PCE (k=2) and QRAO$(3, 1, p)$ across MDKP instance sets. "Feasible %" and "Optimal %" denote the fraction of instances achieving feasibility and optimality, respectively. The optimality gap is the average relative difference from the heuristic ground truth.

Set	Variables	PCE Qubits	Feasible %	Optimal %	Opt. Gap	QRAO Qubits	Feasible %	Opt. Gap
1	[30,50]	[5,7]	100	92	0.0875	[10,30]	92	0.063
2	[80,130]	[8,10]	100	80	0.0645	[40,70]	—	—
3	[140,550]	[11,20]	100	70	0.0540	[70,300]	—	—
4	1026	27	100	100	0.0000	550	—	—

Chapter 6

Future Outlook

The results of this study demonstrate substantial advancements in employing PCE for addressing large-scale combinatorial optimization challenges. Several promising research directions emerge:

- **Adaptive Re-optimization:** Develop intelligent perturbation and warm-start strategies to reduce iterations required for convergence.
- **Extended Compression Schemes:** Explore cubic, biquadratic, and higher-order compression (figure 6.1) to further reduce qubit counts at the expense of circuit depth.
- **Broader Benchmarking:** Apply PCE to diverse combinatorial problems beyond MDKP, spanning logistics, finance, and healthcare.
- **Comprehensive Evaluation Framework:** Establish standardized benchmarks comparing PCE against classical and other quantum-inspired solvers to rigorously quantify relative strengths and limitations.

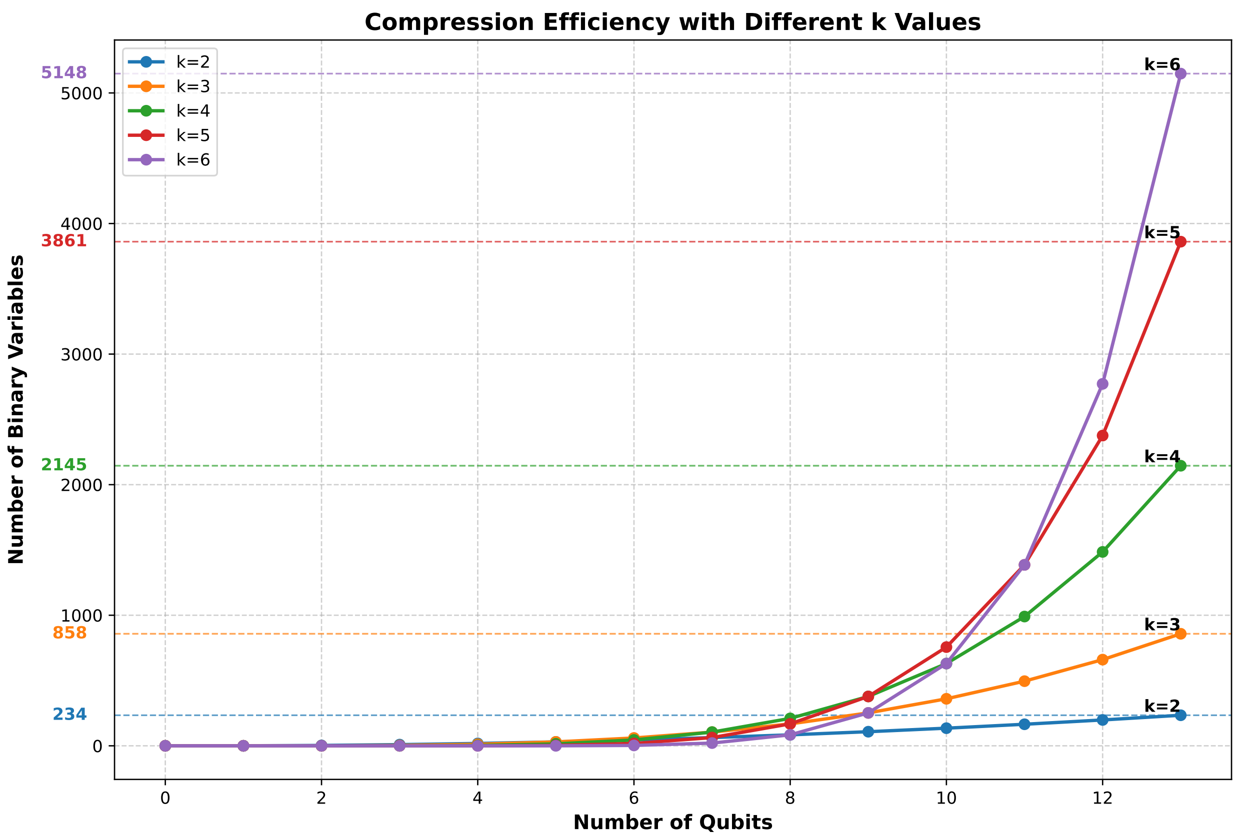

Figure 6.1: Compression efficiency for varying values of k. Each curve plots the maximum number of binary variables $m = 3\binom{n}{k}$ achievable versus qubit count n. Dashed lines mark the variable capacity at $n = 13$ for each k.

Chapter 7

Appendix

7.1 Proof

The objective function for a quadratic unconstrained binary optimization (QUBO) problem can be written in the form:

$$f(x) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} q_{ij} x_i x_j + \sum_{i=1}^{n} c_i x_i. \tag{7.1.1}$$

The equivalence between quadratic 0-1 optimization and the max-cut problem was pointed out by [29] and [27]. For completeness, we present the reduction here.

7.1.1 Reduction

Setting $s_i = 2x_i - 1$, we get $x_i = \frac{s_i+1}{2}$ and can write:

$$f(x(s)) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} q_{ij} \left(\frac{s_i+1}{2}\right)\left(\frac{s_j+1}{2}\right) + \sum_{i=1}^{n} c_i \left(\frac{s_i+1}{2}\right). \tag{7.1.2}$$

Expanding and simplifying, we get:

$$\begin{aligned} f(x(s)) = &\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} q_{ij} \left(\frac{1}{4} s_i s_j + \frac{1}{4} s_i + \frac{1}{4} s_j + \frac{1}{4}\right) \\ &+ \sum_{i=1}^{n} c_i \left(\frac{s_i}{2} + \frac{1}{2}\right). \end{aligned} \tag{7.1.3}$$

Combining terms, we have:

$$\begin{aligned} f(x(s)) = &\frac{1}{4} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} q_{ij} s_i s_j + \frac{1}{4} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} q_{ij} s_i \\ &+ \frac{1}{4} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} q_{ij} s_j + \frac{1}{4} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} q_{ij} \\ &+ \frac{1}{2} \sum_{i=1}^{n} c_i s_i + \frac{1}{2} \sum_{i=1}^{n} c_i. \end{aligned} \tag{7.1.4}$$

Grouping similar terms, we get:

$$\begin{aligned} f(x(s)) &= \frac{1}{4}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} q_{ij}s_is_j + \left(\frac{1}{4}\sum_{i=1}^{n}\sum_{j=i+1}^{n} q_{ij} + \frac{1}{2}\sum_{i=1}^{n} c_i\right) s_i \\ &+ \left(\frac{1}{4}\sum_{i=1}^{n}\sum_{j=i+1}^{n} q_{ij} + \frac{1}{2}\sum_{i=1}^{n} c_i\right) s_j \\ &+ \frac{1}{4}\sum_{i=1}^{n}\sum_{j=i+1}^{n} q_{ij} + \frac{1}{2}\sum_{i=1}^{n} c_i. \end{aligned} \tag{7.1.5}$$

To translate this into a max-cut problem, we introduce a graph $G = (V, E)$ with edge weights w_{ij} for $ij \in E$. We also introduce a dummy node 0 and define additional edge weights w_{0j} for $1 \leq j \leq n$ as:

$$w_{0j} = \frac{1}{4}\left(\sum_{i=1}^{j-1} q_{ji} + \sum_{j=i+1}^{n} q_{ij}\right) + \frac{1}{2}c_i, \quad 1 \leq j \leq n, \tag{7.1.6}$$

$$w_{ij} = \frac{1}{4}q_{ij}, \quad 1 \leq i < j \leq n, \tag{7.1.7}$$

Define the graph $G = (V, E)$ with node set $V = \{0, 1, 2, \ldots, n\}$ and edge set $E = \{(i, j) \mid 0 \leq i < j \leq n\}$, and take w_{ij} as the edge weight of $ij \in E$. Each assignment of values $+1$ and -1 to the variables s_i corresponds to a partition of V into $V^+ = \{i \in V \mid s_i = +1\}$ and $V^- = \{i \in V \mid s_i = -1\}$.

$$g(s) = \sum_{i=0}^{n-1}\sum_{j=i+1}^{n} w_{ij}s_is_j + C_1 \quad \text{where} \quad s \in \{-1, +1\}^{n+1}. \tag{7.1.8}$$

Ignoring the constant term, the objective function can be written as:

$$\text{maximize } c(\delta(W)) \text{ where } W \subseteq V, \tag{7.1.9}$$

where $\delta(W)$ is the cut-set of W, and $c(\delta(W))$ is the sum of the weights of the edges in the cut.

If (V^+, V^-) is a partition of V and $C_2 = \sum_{ij\in E} w_{ij}$, then the equivalent max-cut problem is:

$$\max c(\delta(W)) \text{ where } W \subseteq V. \tag{7.1.10}$$

Let:

$$x_i = \begin{cases} 0 & \text{if } i \in V^-, \\ 1 & \text{if } i \in V^+. \end{cases} \tag{7.1.11}$$

This x_i is a solution to the quadratic 0-1 problem with value $-2z + C_1 + C_2$.

Thus, any QUBO problem can be translated into an equivalent weighted max-cut problem.

To get the solution back from the weighted max cut we make use of the weighted MaxCut solution, the cut $\delta(W)$, then for $i \in \{1, \ldots, n\}$ we set $x_i = 1$ if the $0i \notin \delta(W)$, otherwise set $x_i = 0$.

By utilizing this transformation, we can effectively leverage the pre-trained hyper parameters and the specified loss function outlined in [24].

Bibliography

[1] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. Algorithms and Combinatorics. Springer Berlin Heidelberg, 2006.

[2] Gilbert Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247, 1992.

[3] Valentina Cacchiani, Manuel Iori, Alberto Locatelli, and Silvano Martello. Knapsack problems ‚Äî an overview of recent advances. part i: Single knapsack problems. *Computers & Operations Research*, 143:105692, 2022.

[4] Eric Hyyppä, Suman Kundu, Chun Fai Chan, András Gunyhó, Juho Hotari, David Janzso, Kristinn Juliusson, Olavi Kiuru, Janne Kotilahti, Alessandro Landra, Wei Liu, Fabian Marxer, Akseli Mäkinen, Jean-Luc Orgiazzi, Mario Palma, Mykhailo Savytskyi, Francesca Tosto, Jani Tuorila, Vasilii Vadimov, Tianyi Li, Caspar Ockeloen-Korppi, Johannes Heinsoo, Kuan Yen Tan, Juha Hassel, and Mikko Möttönen. Unimon qubit. *Nature Communications*, 13(1), November 2022.

[5] Amira Abbas, Andris Ambainis, Brandon Augustino, Andreas Bartschi, Harry Buhrman, Carleton Coffrin, Giorgio Cortiana, Vedran Dunjko, Daniel J. Egger, Bruce G. Elmegreen, Nicola Franco, Filippo Fratini Fratini, Bryce Fuller, Julien Gacon, Constantin Gonciulea, Sander Gribling, Swati Gupta, Stuart Hadfield, Raoul Heese, Gerhard Kircher, Thomas Kleinert, Thorsten Koch, Georgios Korpas, Steve Lenk, Jakub Marecek, Vanio Slavov Markov, Guglielmo Mazzola, Stefano Mensa, Naeimeh Mohseni, Giacomo Nannicini, Corey O'Meara, Elena Peña Tapia, Sebastian Pokutta, Manuel Proissl, Patrick Rebentrost, Emre Sahin, Benjamin C. B. Symons, Sabine Tornow, V'ictor Valls, Stefan Woerner, Mira L.A. Wolf-Bauwens, Jon Yard, Sheir Yarkoni, Dirk Zechiel, Sergiy Zhuk, and Christa Zoufal. Quantum optimization: Potential, challenges, and the path forward. 2023.

[6] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

[7] Lov K. Grover. A fast quantum mechanical algorithm for database search, 1996.

[8] Bryce Fuller et al. Approximate Solutions of Combinatorial Problems via Quantum Relaxations. *IEEE Trans. Quantum Eng.*, 5:1–15, 2024.

[9] Taylor L. Patti, Jean Kossaifi, Anima Anandkumar, and Susanne F. Yelin. Variational quantum optimization with multibasis encodings. *Phys. Rev. Res.*, 4:033142, Aug 2022.

[10] Benjamin Tan, Marc-Antoine Lemonde, Supanut Thanasilp, Jirawat Tangpanitanon, and Dimitris G. Angelakis. Qubit-efficient encoding schemes for binary optimisation problems. *Quantum*, 5:454, May 2021.

[11] Marko J. Rančić. Noisy intermediate-scale quantum computing algorithm for solving an n-vertex maxcut problem with $\log(n)$ qubits. *Phys. Rev. Res.*, 5:L012021, Feb 2023.

[12] Elias X. Huber, Benjamin Y. L. Tan, Paul R. Griffin, and Dimitris G. Angelakis. Exponential qubit reduction in optimization for financial transaction settlement, 2023.

[13] Yovav Tene-Cohen, Tomer Kelman, Ohad Lev, and Adi Makmal. A variational qubit-efficient maxcut heuristic algorithm, 2023.

[14] M. R. Perelshtein, A. I. Pakhomchik, Ar. A. Melnikov, M. Podobrii, A. Termanova, I. Kreidich, B. Nuriev, S. Iudin, C. W. Mansell, and V. M. Vinokur. Nisq-compatible approximate quantum algorithm for unconstrained and constrained discrete optimization. *Quantum*, 7:1186, November 2023.

[15] Christophe Piveteau and David Sutter. Circuit knitting with classical communication. *IEEE Transactions on Information Theory*, 70(4):2734–2745, April 2024.

[16] Lukas Brenner, Christophe Piveteau, and David Sutter. Optimal wire cutting with classical communication, 2023.

[17] Friedrich Wagner, Jonas Nüßlein, and Frauke Liers. Enhancing quantum algorithms for quadratic unconstrained binary optimization via integer programming, 2024.

[18] Tianyi Peng, Aram W. Harrow, Maris Ozols, and Xiaodi Wu. Simulating large quantum circuits on a small quantum computer. *Physical Review Letters*, 125(15), October 2020.

[19] Kosuke Mitarai and Keisuke Fujii. Overhead for simulating a non-local channel with local channels by quasiprobability sampling. *Quantum*, 5:388, January 2021.

[20] Christian Ufrecht, Maniraman Periyasamy, Sebastian Rietsch, Daniel D. Scherer, Axel Plinge, and Christopher Mutschler. Cutting multi-control quantum gates with zx calculus. *Quantum*, 7:1147, October 2023.

[21] Lukas Schmitt, Christophe Piveteau, and David Sutter. Cutting circuits with multiple two-qubit unitaries, 2024.

[22] Angus Lowe, Matija Medvidović, Anthony Hayes, Lee J. O'Riordan, Thomas R. Bromley, Juan Miguel Arrazola, and Nathan Killoran. Fast quantum circuit cutting with randomized measurements. *Quantum*, 7:934, March 2023.

[23] Xinjian Yan, Xinwei Lee, Ningyi Xie, Yoshiyuki Saito, Leo Kurosawa, Nobuyoshi Asai, Dongsheng Cai, and HoongChuin Lau. Light cone cancellation for variational quantum eigensolver ansatz, 2024.

[24] Marco Sciorilli, Lucas Borges, Taylor L. Patti, Diego García-Martín, Giancarlo Camilo, Anima Anandkumar, and Leandro Aolita. Towards large-scale quantum optimization solvers with few qubits, 2024.

[25] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer Berlin Heidelberg, 2013.

[26] Frank Gurski, Carolin Rehs, and Jochen Rethmann. Knapsack problems: A parameterized point of view. *Theoretical Computer Science*, 775:93–108, 2019.

[27] F BARAHONA, M JUNG, and G REIN. Experiments in quadratic 0-1 programming. 2005.

[28] Monit Sharma, Yan Jin, Hoong Chuin Lau, and Rudy Raymond. Quantum relaxation for solving multiple knapsack problems, 2024.

[29] Peter L. Ivănescu. Some network flow problems solved with pseudo-boolean programming. *Operations Research*, 13(3):388–399, 1965.