Dynamic Entangling Layers: A Novel Approach to Improving Ansatz Expressibility in Solving Combinatorial Problems

Project report submitted
as part of summer internship

by

ALAN AJAY

National Institute of Science Education and Research

Tehsildar Office, Khurda

Pipli, Near, Jatni, Odisha 752050

Under the guidance of

Dr. Anirban Pathak

Professor and Head of Department
Department of Physics and Materials Science and Engineering
Jaypee Institute of Information Technology (JIIT)
Noida, Uttar Pradesh, India

September 3, 2025

DECLARATION

I hereby declare that this is an original project report done under the supervision of Dr. Anirban Pathak from the Jaypee Institute of Information Technology (JIIT).

Signature of the Student

Date:

The project work reported in the project report entitled **"Dynamic Entangling Layers: A Novel Approach to Improving Ansatz Expressibility in Solving Combinatorial Problems"** was carried out by Alan Ajay of the National Institute of Science Education and Research (NISER), Bhubaneswar, under my supervision.

APathak

Signature of the project supervisor

School: PMSE Dept., JIIT, NOIDA

Date: 3/9/25

Dynamic Entangling Layers: A Novel Approach to Improving Ansatz Expressibility in Solving Combinatorial Problems

Alan Ajay[1], Britant[2], and Anirban Pathak[2]
[1] *National Institute of Science Education and Research, Bhubaneswar, Odisha, India and*
[2] *Jaypee Institute of Information Technology, Sector-62, Noida, Uttar Pradesh, India*

(Dated: September 6, 2025)

We present a dynamic hardware-efficient ansatz design for variational quantum algorithms applied to the Max-Cut problem. Our approach, *MaxCutDynamicSolver*, alternates the entanglement pattern across optimization iterations, cycling through qubit pairs to enhance the circuit's expressibility while maintaining a compact parameter count. This alternating structure aims to explore a richer subspace of the Hilbert space than fixed-pattern ansatzes, potentially mitigating limitations such as barren plateaus. We compare the proposed method against the hardware efficient *EfficientSU2* along with a brute force classical algorithm. Our results indicate that dynamic entanglement can improve solution quality for certain graph topologies without increasing circuit depth, suggesting a promising direction for expressibility-driven ansatz design in VQA.

I. INTRODUCTION

Variational Quantum Algorithms (VQAs) are a class of quantum–classical hybrid algorithms that hold significant promise in the noisy intermediate-scale quantum (NISQ) era [1, 2]. In these algorithms, a quantum processor prepares a parameterized quantum state $|\psi(\theta)\rangle$ using an ansatz circuit, and measures an objective function relevant to the problem. A classical optimizer then updates the parameters θ to minimize or maximize this objective.

The Variational Quantum Eigensolver (VQE) is a prominent VQA tailored for optimisation problems, where the objective is to minimize the expectation value of a cost Hamiltonian. Among the many ansatz design strategies, hardware-efficient ansatze and fixed entanglement topologies have been widely adopted due to their low depth and compatibility with current hardware [3]. However, such fixed entanglement structures may limit expressibility or exacerbate issues such as barren plateaus [4, 5].

In this work, we investigate an *alternating entanglement* strategy, where the entangling pattern is systematically varied across layers. This approach aims to improve the ansatz expressibility and exploration of the Hilbert space, while keeping circuit depth and parameter count comparable to conventional designs. Our study applies this method in the context of the Max-Cut problem, and benchmarks it against standard hardware-efficient ansatze.

II. PROBLEM FORMULATION

The Max-Cut problem seeks to partition the vertex set V of a weighted undirected graph $G = (V, E, w)$ into two disjoint subsets such that the total weight of edges crossing the partition (the *cut*) is maximized. Let $w_{ij} \geq 0$ denote the weight of edge $(i, j) \in E$.

A. Cut definition and notation

Given a bipartition of the vertex set V into S and its complement $\bar{S}$,

$$S \subseteq V, \qquad \bar{S} \coloneqq V \setminus S,$$

the cut value associated with this partition is

$$\mathrm{Cut}(S, \bar{S}) \;=\; \sum_{\substack{(i,j)\in E \\ i\in S,\, j\in \bar{S}}} w_{ij}. \tag{1}$$

Here S denotes the set of vertices assigned to one side of the partition and $\bar{S}$ the set assigned to the other side. The sum in Eq. (1) counts each edge that has its endpoints on different sides of the partition exactly once.

B. Binary / spin encodings

To convert Max-Cut into a Hamiltonian optimization problem suitable for quantum algorithms, we introduce binary variables $b_i \in \{0, 1\}$ (or equivalently spin variables $s_i \in \{\pm 1\}$). A common mapping is $s_i = 1 - 2b_i \in \{+1, -1\}$, equivalently $b_i = \frac{1-s_i}{2}$. Under this mapping two vertices i and j belong to *different* parts of the partition if and only if $s_i s_j = -1$. The indicator that i and j are in different sets can be written as

$$\mathbf{1}\{b_i \neq b_j\} \;=\; \frac{1 - s_i s_j}{2}. \tag{2}$$

Consequently the cut value admits the spin representation

$$\mathrm{Cut}(S, \bar{S}) \;=\; \sum_{(i,j)\in E} w_{ij}\, \frac{1 - s_i s_j}{2}. \tag{3}$$

C. Ising / Pauli-Z Hamiltonian Representation

On a quantum device, each vertex i is represented by a qubit. The classical spin variable $s_i \in \{\pm 1\}$ is mapped to the eigenvalues of the Pauli-Z operator, Z_i. A convenient problem Hamiltonian for Max-Cut can then be formulated as:

$$H_{\text{MaxCut}} = \sum_{(i,j)\in E} \frac{w_{ij}}{2}\big(I - Z_i Z_j\big). \tag{4}$$

For any computational basis state, the expectation value $\langle H_{\text{MaxCut}}\rangle$ is equal to the classical cut value. Therefore, maximizing this expectation value solves the Max-Cut problem.

For implementation purposes, it is common to use an equivalent, sign-flipped Hamiltonian. The formulation used in the accompanying code [6] is:

$$\widetilde{H} = \sum_{(i,j)\in E} \frac{w_{ij}}{2}\big(Z_i Z_j - I\big) = -H_{\text{MaxCut}}. \tag{5}$$

With this convention, the expectation value $\langle \widetilde{H}\rangle$ corresponds to the *negative* of the cut value. Consequently, the goal of the classical optimizer becomes to **minimize** the energy $E = \langle \widetilde{H}\rangle$, and the final Max-Cut value is recovered as $-E$.

D. VQE Formulation for Max-Cut

The Variational Quantum Eigensolver (VQE) is a hybrid quantum-classical algorithm designed to find the minimum eigenvalue of a Hamiltonian by leveraging the variational principle. This principle states that for a given Hamiltonian H, the expectation value for any trial wavefunction $|\psi(\boldsymbol{\theta})\rangle$ is always greater than or equal to the true ground-state energy E_0:

$$\langle \psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle \geq E_0. \tag{6}$$

The algorithm aims to find an optimal set of parameters $\boldsymbol{\theta}_{\text{opt}}$ that minimizes this expectation value, thereby providing an approximation of the ground-state energy.

The VQE workflow proceeds as follows:

1. A parameterized quantum circuit, known as an *ansatz*, prepares a trial state $|\psi(\boldsymbol{\theta})\rangle$ on the quantum computer. The parameters $\boldsymbol{\theta}$ are typically rotation angles that can be adjusted.

2. The expectation value of the problem Hamiltonian (in our case, $\widetilde{H}$ from Eq. (5)) is measured for the state $|\psi(\boldsymbol{\theta})\rangle$. This value serves as the cost function for the classical optimizer.

3. A classical optimizer receives the expectation value and suggests a new set of parameters $\boldsymbol{\theta}'$ to try in the next iteration.

4. This quantum-classical loop is repeated until the energy value converges to a minimum.

For the Max-Cut problem, minimizing the expectation value $E(\boldsymbol{\theta}_{\text{opt}}) = \langle \psi(\boldsymbol{\theta}_{\text{opt}})|\widetilde{H}|\psi(\boldsymbol{\theta}_{\text{opt}})\rangle$ is equivalent to finding the ground state of the Hamiltonian. As established in Eq. (5), this minimum energy corresponds to the negative of the maximum cut value. Therefore, the solution to the Max-Cut problem is recovered as $-E(\boldsymbol{\theta}_{\text{opt}})$.

E. Measurements and recovering bitstrings

In practice, the expectation value C_p is estimated by repeated measurements in the computational basis. A single measurement yields a bitstring $b \in \{0,1\}^n$; the empirical distribution of sampled bitstrings approximates the probability distribution $|\langle b \mid \psi_p\rangle|^2$. For each observed bitstring we compute the classical cut value using Eq. (1) (or equivalently Eq. (3)) and select the bitstring with the largest cut among the samples as the candidate solution. The expected value estimated from sampling converges (in the limit of many shots) to the true quantum expectation $\langle H_C\rangle$.

F. Classical baseline and verification

For small n we compute the exact classical optimum by exhaustive enumeration: evaluate $\text{Cut}(S,\bar{S})$ for every 2^n partition (or, equivalently, for every bitstring b) and take the maximum. This classical ground truth is used to compute approximation ratios and to verify the correctness of the quantum-derived solutions in the numerical experiments described below.

III. METHODS

Our methodology is centered on a variational quantum-classical hybrid approach. We introduce and evaluate a novel dynamic ansatz, benchmarking its performance against a standard, fixed quantum ansatz and an exact classical solver. The optimization for the quantum solvers is driven by the COBYLA algorithm, which minimizes the expectation value of the problem Hamiltonian computed via Qiskit's noiseless `StatevectorEstimator` [7].

A. Dynamic Alternating Ansatz

The core of our proposed method is the `MaxCutDynamicSolver`, which employs a hardware-efficient ansatz with a non-fixed entanglement structure [6]. The circuit consists of two main components:

1. **Rotation Layer:** A layer of single-qubit rotations, specifically $R_Y(\theta_i)$ on each qubit, parameterized by a vector of angles $\boldsymbol{\theta}$.

2. **Entanglement Layer:** A single CNOT (CX) gate applied to a pair of qubits.

The key feature of this design is that the pair of qubits entangled by the CNOT gate is chosen cyclically at each iteration of the classical optimization loop. By systematically varying the entanglement across all possible qubit pairs, this approach aims to enhance the circuit's expressibility and its ability to explore the solution landscape without increasing circuit depth or the number of parameters.

B. Baseline Quantum Solver: Fixed SU(2) Ansatz

To provide a standard quantum baseline for comparison, we utilize a conventional hardware-efficient ansatz from Qiskit's circuit library, `EfficientSU2` [3, 7]. This circuit consists of alternating layers of single-qubit rotations (R_Y) and a *fixed* pattern of entangling CNOT gates. For our experiments, we configured this ansatz with a single repetition and 'full' entanglement, where CNOTs are applied across all qubit pairs. This ensures a fair comparison against our dynamic method by using a similar class of circuit with a comparable parameter count.

C. Classical Benchmark: Brute-Force Solver

To accurately assess the performance and compute the approximation ratio of the quantum algorithms, we implemented a classical brute-force solver [6]. This algorithm provides the exact optimal solution to the Max-Cut problem by exhaustively evaluating the cut value for all 2^n possible partitions of a graph with n vertices. While computationally intractable for larger graphs, it serves as the definitive ground truth for the small-scale problem instances analyzed in this study.

IV. RESULTS AND DISCUSSIONS

To evaluate the performance of the VQE implementation, we compared the Max-Cut values obtained on a complete graph of 8 nodes using two different ansatz choices: the conventional `SU2` ansatz and the proposed `Dynamic Ansatz`. Figure 1 shows the progression of Max-Cut values across 20 iterations for both ansatzes.

The results reveal several important trends:

- **SU2 Ansatz:** The iterations with the standard SU2 ansatz (blue circles) display significant fluctuations and relatively lower convergence around the optimum. The mean value is approximately 364, with a broad spread reflected in the shaded blue region (mean ± standard deviation). This suggests that while SU2 provides a baseline approach, its expressive power is limited for this problem instance, leading to suboptimal solutions.

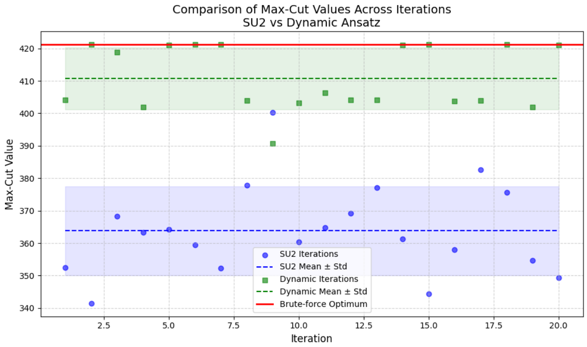

Figure 1: Comparison of Max-Cut values across iterations between the standard SU2 ansatz and the proposed Dynamic Ansatz. The red line indicates the brute-force optimum. Shaded regions represent the mean ± standard deviation for each method.

- **Dynamic Ansatz:** In contrast, the Dynamic Ansatz (green squares) consistently achieves values much closer to the brute-force optimum (red line at ~ 421). The green shaded region, representing the mean ± standard deviation, shows a much narrower distribution compared to SU2. This indicates both higher accuracy and greater stability in optimization. The mean for the Dynamic Ansatz lies around 411, which is significantly closer to the optimum, underscoring the advantage of adapting entanglement patterns dynamically.

- **Comparison to Brute-force Optimum:** The brute-force optimum provides an absolute reference point. While SU2 values remain far from the optimum across iterations, the Dynamic Ansatz demonstrates that a well-designed ansatz can substantially bridge the gap between heuristic quantum optimization and exact classical solutions.

- **Statistical Stability:** The smaller spread in the Dynamic Ansatz outcomes highlights improved robustness. This is particularly relevant in noisy intermediate-scale quantum (NISQ) devices, where high variance can lead to unreliable results [2]. Thus, the Dynamic Ansatz is not only more accurate but also more consistent.

Implications: These findings emphasize the importance of ansatz design in VQA. The Dynamic Ansatz, by modifying the entanglement structure during optimization, provides a more expressive variational form that can explore the solution space more effectively. This suggests a promising direction for variational algorithms: instead of relying on fixed, hardware-efficient circuits such as SU2, one can tailor the ansatz to the problem structure, thereby achieving solutions that are both closer to the optimum and less sensitive to noise.

In summary, the results strongly indicate that the Dynamic Ansatz outperforms the conventional SU2 ansatz for the Max-Cut problem, both in terms of accuracy and stability. Future work will investigate scaling this approach to larger graphs and analyzing performance under realistic noise models.

V. CONCLUSION

In this study, we introduced and evaluated a dynamic ansatz design, *MaxCutDynamicSolver*, for solving the Max-Cut problem using a VQE-like variational algorithm. Our approach systematically alternates the entangling gate pattern across optimization iterations, aiming to enhance the expressibility of the quantum circuit without increasing its depth or parameter count.

Our numerical experiments, conducted on different graph topologies upto size 10, demonstrate that the dynamic ansatz consistently outperforms the standard hardware-efficient `EfficientSU2` ansatz. Specifically, the proposed method achieved solutions closer to the classically computed optimum and exhibited greater stability, as evidenced by a lower variance in the results across multiple runs. This suggests that varying the entanglement structure allows the ansatz to explore a more relevant subspace of the Hilbert space, thereby navigating the optimization landscape more effectively and mitigating the risk of getting trapped in poor local minima.

The primary contribution of this work is the demonstration that dynamic, hardware-efficient circuit structures can be a powerful tool for improving VQA performance. While our investigation was limited to noiseless statevector simulations, the results strongly motivate future research in several key directions. These include:

- Comparing the *MaxCutDynamicSolver* against better alternatives of `EfficientSU2`
- Investigating the performance of this approach on larger graphs where classical brute-force solutions are intractable.
- Analyzing the robustness of the dynamic ansatz under realistic hardware noise models to assess its viability for current and near-term quantum devices.
- Exploring more sophisticated, non-cyclic schedules for altering the entanglement, potentially adapting the pattern based on optimizer feedback.
- Applying this dynamic ansatz philosophy to other combinatorial optimization problems and different classes of variational quantum algorithms.

In conclusion, our findings indicate that moving beyond fixed ansatz topologies towards more adaptable, expressibility-driven designs is a promising avenue for unlocking the potential of variational quantum algorithms in the NISQ era [1, 2].

ACKNOWLEDGMENTS

I am thankful to my Internship guide Dr. Anirban Pathak for providing me the opportunity to do this internship and his student Mr. Britant whose mentorship was invaluable for the completion of this project.

[1] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, *et al.*, Nature Reviews Physics **3**, 625 (2021).

[2] J. Preskill, Quantum **2**, 79 (2018).

[3] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Nature **549**, 242 (2017).

[4] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, Advanced Quantum Technologies **2**, 1900070 (2019).

[5] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Nature Communications **9**, 4812 (2018).

[6] A. Ajay, "Summer internship 2025: Explorations in quantum computing," `https://github.com/alan-ajay/Summer-internship-2025.git` (2025), gitHub repository, Accessed: September 1, 2025.

[7] Qiskit contributors, "Qiskit: An Open-source Framework for Quantum Computing," `https://qiskit.org` (2023), dOI: 10.5281/zenodo.2573505.

Appendix A: Code Availability

The source code and Jupyter notebooks used for the experiments in this paper are publicly available in the following GitHub repository:

- `https://github.com/alan-ajay/Summer-internship-2025.git`

The specific notebook containing the implementation of the dynamic ansatz solver can be found at:

- `https://github.com/alan-ajay/Summer-internship-2025/blob/main/ansatz_engineering.ipynb`