PRODUCT REQUIREMENTS DOCUMENT (PRD)

Project Name: 2B or not 2B
Team Name: FLMA8
GitHub Repository: https://github.com/sareensabra/2026-NVIDIA

Role	Name	GitHub	Discord
Project Lead (Architect)	Sareen Sabra	@sareensabra	Nope2345956
GPU Acceleration PIC (Builder)	Ahmad Sabra Fadia Hamami	@Ahmads-Eng @FadiaHamami	ahmds1218983 f_ezz_h
Quality Assurance PIC (Verifier)	Leena Hamami	@leena241126	leena05127
Technical Marketing PIC (Storyteller)	Sara Sabra	@sarasabra	sa_ra9493

ARCHITECTURE

The Low Autocorrelation Binary Sequence (LABS) problem is formulated as an energy minimization task over binary sequences $S \in \{-1, +1\}^N$ where the objective is to minimize the autocorrelation value of the *sequence.*

This objective can be expressed as a classical Ising-style Hamiltonian, whose ground state corresponds to a sequence with minimal autocorrelation energy.
We treat the LABS energy function as a cost Hamiltonian and adopt a hybrid quantum–classical optimization workflow, where a variational quantum algorithm generates structured sequence population that can be sampled and refined using a classical Memetic Tabu Search (MTS) solver to get the most optimal sequence.

ALGORITHM

Variational Quantum Eigensolver (VQE) with a hardware-efficient, layered rotation–entanglement ansatz.

Motivation

We selected VQE due to its natural alignment with energy-based optimization problems that can be mapped to an Ising-style Hamiltonian. LABS exhibits a complex energy landscape with many local minima, making it well-suited for a variational approach that explores a continuous parameter space rather than a fixed-depth, problem-structured circuit.

Unlike QAOA, which constrains the search to a predefined alternating operator structure, VQE enables flexible ansatz design. This flexibility is particularly valuable for our goal to generate high-quality, structured seeds that improve the convergence and solution quality of the downstream classical MTS solver.

The variational framework also supports iterative refinement driven by classical feedback, allowing the quantum and classical components to co-evolve within a unified optimization loop.

Method

Quantum Stage (VQE):
A parameterized quantum circuit is optimized to minimize the LABS Hamiltonian expectation value. The resulting measurement samples and low-energy bitstrings are extracted as candidate binary sequences.

Seeding Stage:
These quantum-generated sequences are used to initialize the population of the classical Memetic Tabu Search, replacing purely random initial guesses.

Classical Stage (MTS):
The MTS algorithm performs local and global search using tabu memory and neighborhood exploration, refining the quantum seeds into high-quality final solutions.

This approach leverages the quantum model's ability to explore correlated, structured regions of the solution space, while relying on the classical solver's strength in systematic exploitation and convergence.

SUPPORTING LITERATURE

VQE Applications

Khalid et al., *Quantum Computing for Intelligent Transportation Systems: VQE-Based Traffic Routing and EV Charging Scheduling*, *Mathematics* (2025).
https://doi.org/10.3390/math13172761

This article highlights the advances in quantum computing that have led to the development of robust quantum computational frameworks capable of addressing complex optimization problems while exceeding the capabilities of classical algorithms, especially through hybrid quantum-classical (HQC) methodologies.

More research is needed to bridge the gap between idealized simulation environments and real-world quantum hardware implementations, A comprehensive error mitigation strategy that is for NISQ devices needs more detailed investigation.

This research also demonstrated the applicability of VQE and Quantum computing and optimization in non chemistry domains, such as the Transportation sectors, through the Intelligent transportation Systems (ITS) for traffic control and infrastructure. Existing practices are also implemented for Market Share analysis and Sports Timetabling problems.

Warm Starting Optimization

Eggers et. al. Warm Starting Quantum Optimization, 2020
https://doi.org/10.48550/arXiv.2009.10095

The results demonstrate that warm-starting the Quantum Approximate Optimization Algorithm (QAOA) significantly improves performance at shallow circuit depths. This finding is important for dense combinatorial optimization problems when in noisy quantum hardware. Warm-start QAOA consistently identifies higher-quality solutions than the standard QAOA, highlighting the value of informed initialization for hybrid quantum-classical workflows.

ACCELERATION STRATEGY

Quantum Accelerations

To reduce simulation cost and improve throughput, we apply circuit-level optimizations prior to GPU execution.

Transpilation & Gate Reduction

All variational circuits are transpiled to minimize the use of native gate operations and reduce overall gate count. We target hardware-efficient decompositions that collapse multi-qubit operators into the smallest possible set of native rotations and entangling gates. This reduces circuit depth and lowers the computational burden of statevector evolution on the GPU.

Depth Minimization

We prioritize shallow circuit constructions by limiting entanglement range and reordering commuting operations. This decreases the number of sequential gate layers, improving both numerical stability and simulation performance.

Mini-Circuit Decomposition & Parallelization

Expectation value estimation is decomposed into multiple smaller sub-circuits corresponding to disjoint Hamiltonian terms. These mini-circuits are executed concurrently across GPU threads and, where available, across multiple GPUs. This enables parallel sampling and batched evaluation of variational parameters, significantly reducing wall-clock time per VQE iteration.

Impact

This approach transforms a monolithic, deep quantum circuit into a collection of shallow, parallel workloads that better match the GPU execution model, improving scalability, memory efficiency, and time-to-solution.

Classical Acceleration (MTS + FFT)

The dominant computational cost in MTS is repeated evaluation of the LABS energy function during neighborhood exploration. We address this bottleneck by reformulating autocorrelation as a convolution operation.

Strategy:

Autocorrelation is computed using the Fast Fourier Transform (FFT), reducing computational complexity from $O\ (N^2)$ to $(N\ log\ N)$ GPU acceleration is enabled via batched FFT execution (cuFFT), allowing thousands of candidate neighbor sequences to be evaluated in parallel.

Impact

Transforms the MTS inner loop from a sequential bottleneck into a massively parallel workload. It also Enables large-scale neighborhood evaluation without sacrificing throughput. Through this there is improved scalability.

Hardware Targets

Development Environment:

CPU-based quantum simulation for logic validation and unit testing.

VERIFICATION PLAN

Acceleration Environment:
GPU-backed CUDA-Q simulation and cuFFT-based classical kernels on NVIDIA L4 and A100-class GPUs for large-scale benchmarking and final performance evaluation.

AI Guardrails & Code Validation

All AI-generated quantum circuits, CUDA-Q kernels, and GPU acceleration code must pass a CPU–GPU parity test suite before integration.
This ensures that acceleration does not compromise correctness and that performance gains do not mask numerical or logical errors.

Core Correctness Tests

Test 1 — Symmetry Invariance
LABS energy must be invariant under sequence negation and reversal. For any sequence S, we assert that $E(S) = E(-S) = E(reverse(S))$. This verifies correct implementation of autocorrelation and Hamiltonian symmetry.

Test 2 — FFT vs Direct Autocorrelation
FFT-based energy computation is validated against a direct $O(N^2)$ autocorrelation implementation for small and medium problem sizes. Outputs must match within numerical tolerance.

Test 3 — GPU vs CPU Parity
All quantum and classical energy evaluations are compared between GPU and CPU implementations to ensure consistency across hardware backends.

Test 4 — Ground Truth for Small N
For small values of N, where brute-force solutions are available, computed energies are validated against known optimal values.

Algorithmic Validation

Seed vs Non-Seed Comparison
We benchmark MTS convergence using Random initialization and VQE-seeded initialization

Metrics include final energy after a fixed iteration budget and time-to-convergence, validating whether quantum seeding provides a measurable advantage.

Ansatz Sensitivity Testing
We evaluate different VQE ansatz structures and depths to analyze the impact of circuit expressivity on seed quality and downstream classical convergence.

RESOURCE MANAGEMENT PLAN

Division of Credits and Runtime
We ensured that our credits are managed and divided for pre-testing, post code testing with first draft and then final testing prior to the GPU

Monitoring
The GPU acceleration PIC would monitor the testing, we also had a strategy for using the portals and GPU at different times to minimize the workload and runtime.