Accelerated Quantum-Enhanced Memetic Search (A-QEMS): Phase 1 Technical Specification & Verification

Arife Nur Ayaz - QuAy
MIT iQuHACK 2026 - NVIDIA Challenge

1 Executive Summary

The project aims to solve the Low Autocorrelation Binary Sequence (LABS) problem for $N > 40$ by overcoming the "Glassy Landscape" limitation of standard quantum annealers. We replace the tutorial's standard Counterdiabatic Driver with a **Gradient-Biased Non-Hermitian Driver**. This system utilizes energy-gradient-mapped imaginary potentials and a **First-Order Symplectic Split-Operator** scheme. This architecture enables a **Digitized Counterdiabatic (DCQO)** approach that preserves the Hamiltonian structure during 'gradient kicks' while avoiding the numerical overhead of higher-order Runge-Kutta stages.

2 Verification Strategy (QA Plan)

We reject "ad-hoc" print statements. Our verification strategy relies on a dedicated automated test suite (`tests.py`) that runs before every major compute job.

The "System Check" Protocol

Ground Truth Validation ($N = 3$):
Test: Brute-force calculate all $2^3 = 8$ sequences.
Assertion: `calculate_energy([1, 1, -1]) == 1.0` (Manual verification).

Symmetry Invariant Testing:
Test: Generate random sequence S.
Assertion: Energy(S) == Energy($-S$) **AND** Energy(S) == Energy(Reverse(S)).
Purpose: Ensures the Hamiltonian construction hasn't violated physical symmetries.

Benchmark Calibration (Barker-13):
Test: Input the known Barker-13 sequence.
Assertion: Merit_Factor(Barker13) ≈ 14.08.
Purpose: Calibrates the objective function against known literature values.

Quantum Sanity Check:
Test: Compare Mean Energy of Quantum Ansatz (E_Q) vs. Random Noise (E_R) for $N = 20$.
Assertion: $E_Q < E_R$ (The quantum algorithm must provide a head-start).

Symplectic Invariant Assertion:
Test: Run the evolution for 100 Trotter steps with $\Delta t = 0.01$.
Assertion: Energy deviation $\Delta E < 10^{-5}$ across the trajectory.
Purpose: Validates that the **Velocity Verlet** / Split-Operator integrator preserves the phase-space volume, essential for long-range sequence optimization.

Stagnation Recovery Test:
Test: Apply Gradient-Biased Shockwave to a trapped population (e.g., $E = 108$).
Assertion: Post-shock mean Inverse Participation Ratio (IPR) < 0.5.
Purpose: Confirms the driver successfully delocalizes the state for tunneling out of local minima.

3 The Research Requirement

Choice of Quantum Algorithm: Gradient-Biased Non-Hermitian Optimization

We have evolved the Hatano-Nelson model into a dynamic **Quantum Gradient Descent** driver.

- **Gradient Mapping:** Unlike random drift, our drift parameter δ is determined by $\nabla E(s)$. This biases the quantum flow toward lower energy configurations, acting as a "quantum kick" through high-energy barriers.
- **Symmetry-Protected Projection:** We implement an active projection operator $\mathcal{P}_{\mathbb{Z}_2 \times \mathbb{Z}_2}$ that maps the state back to the Symmetric ($S = S_{rev}$) or Skew-Symmetric ($S = -S_{rev}$) subspace after every integration step. This ensures that the quantum driver never wanders into redundant configurations, mathematically guaranteeing the 75% search space reduction.

Theoretical Basis:

1. Hatano, N., & Nelson, D. R. (1996). "Localization Transitions in Non-Hermitian Quantum Mechanics." (*Phys. Rev. Lett*).
2. Gomez Cadavid, A., et al. (2025). "Scaling advantage with quantum-enhanced memetic tabu search." (*arXiv:2511.04553*).
3. Chandarana, P., et al. (2023). "Digitized counterdiabatic quantum algorithm for protein folding." (*Phys. Rev. Appl. 20, 014024*).

GPU Acceleration Strategy

- **Quantum Kernel:** We utilize CUDA-Q with the `tensornet` (Matrix Product State) backend. This allows us to simulate the non-unitary Hatano-Nelson evolution efficiently on GPU by compressing the state vector.
- **Classical Kernel:** We replace the sequential Python loops of the Tabu Search with **CuPy**.
- **Batching:** We evaluate the energy delta of all N possible bit-flips simultaneously using GPU matrix operations ($O(1)$ parallel depth) rather than iteratively ($O(N)$).

4 Execution Tactics

Agentic Workflow

- **Agent 1 (The Architect):** Responsible for mapping the Physics equations to the LABS Hamiltonian.
- **Agent 2 (The Coder):** Responsible for translating the math into `cupy` kernels and refactoring the `main.py` loop.

Success Metrics

- **Mathematical Stability:** Zero divergence in the energy gradient during $N = 40$ symplectic 'kicks', ensuring convergence to $E \leq 108$ basins within the Tier 2 budget.
- **Optimality:** Achieve Energy $E \leq 80$ for $N = 40$ using gradient-biased seeding.
- **Efficiency:** Achieve a 75% reduction in redundant state evaluations through Symmetry Protection.

Resource Management (Budget: $20)

We treat compute credits as cash. We utilize a Tiered Deployment Strategy:

Tier	**Hardware**	**Cost**	**Usage Strategy**	**Est. Cost**
Tier 1: Dev	L4 GPU	$\sim$ \$0.80/hr	Code debugging, Unit Tests, Small N ($N < 30$) runs.	5 hrs = \$4.00
Tier 2: Prod	A100 GPU	$\sim$ \$2.50/hr	Only for "Hero Runs" ($N = 40, 50, 60$) and final benchmarking.	4 hrs = \$10.00
Buffer	N/A	N/A	Reserved for mistakes/overruns.	\$6.00

Protocol: Auto-shutdown script (`time_limit=900`) will be implemented on all batch jobs to ensure no instance runs overnight.