NVIDIA iQuHACK 2026

February 2026

QUANTUM OPTIMIZATION

Quantum-Enhanced Optimization for LABS

Implementing a sampling based Conditional Value At Risk (VaR) - Variational algorithm to enhance a seeded MTS classical solver

TEAM
LABSrats

FRAMEWORK
CUDA-Q

Low Autocorrelation Binary Sequences

01

THE PLAN & THE PIVOT

Strategy: CVaR-VQA-MTS

Core Objective

Solve the **Low Autocorrelation Binary Sequence (LABS)** problem using quantum-enhanced optimization. The quantum advantage might come from providing MTS with a better initial population - bitstrings already biased toward low-energy LABS configurations. This allows MTS to converge faster, improving the overall scaling

PROBLEM SIGNIFICANCE

- Radar pulse design for enhanced target detection
- Telecommunication protocol optimization
- Signal processing with minimal interference

Technical Approach

1. Sampling based variational approach for quantum seeding
2. Memetic Tabu Search for classical refinement
3. CUDA-Q hybrid orchestration framework

Development Roadmap

Phase 1: Foundation
Implement the CVaR-VQA in CUDA-Q, validate with small N

Phase 2: Integration
Connect quantum seeding to classical tabu search

Phase 3: Optimization
Tune parameters for scaling advantage

Phase 4: Validation
Benchmark against classical baselines

Target Performance

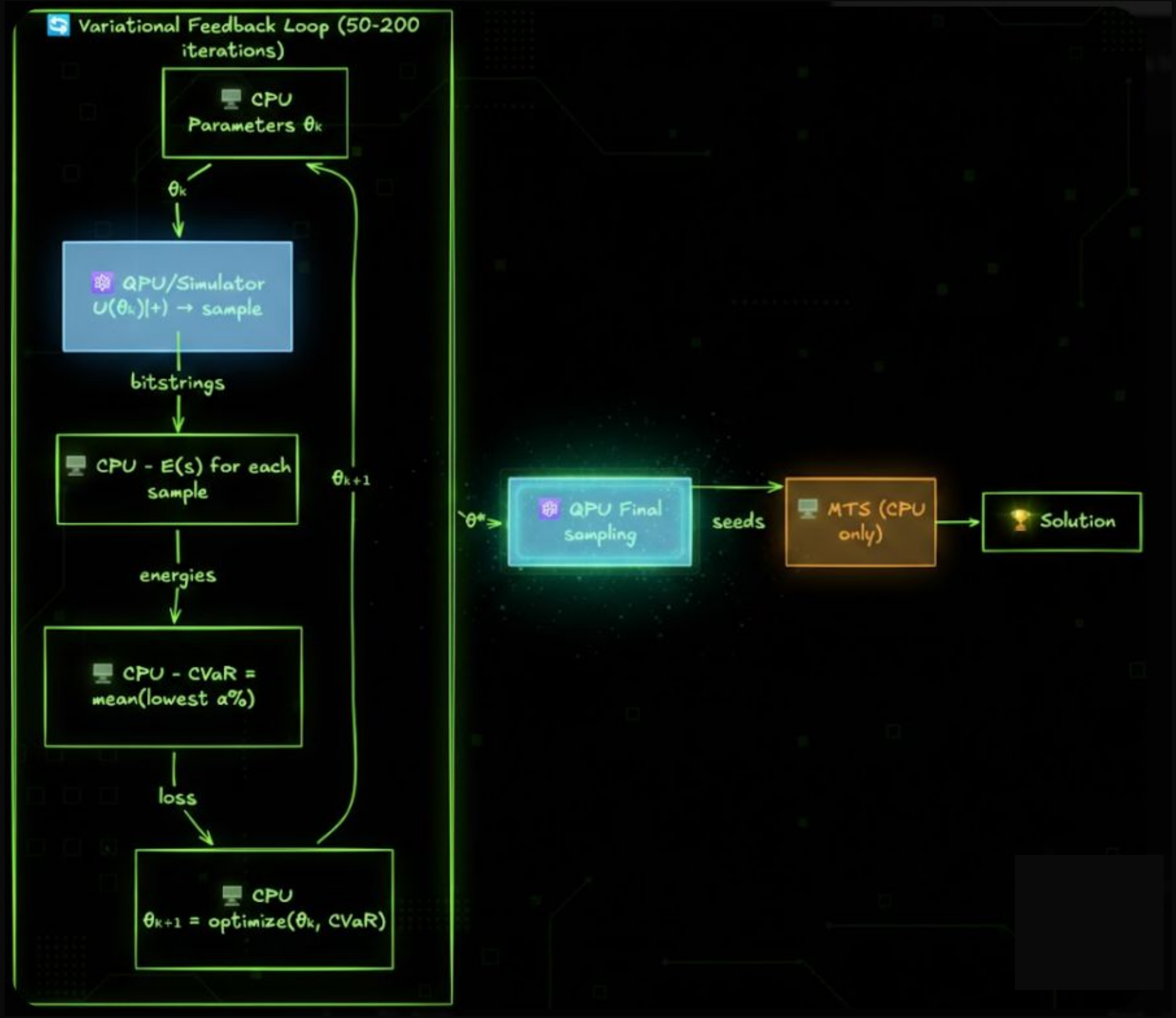
O(1.24^N)	N=37
Scaling Target	Max Qubits

TECHNICAL IMPLEMENTATION

02

The Algorithm: CVaR-VQA-MTS Architecture

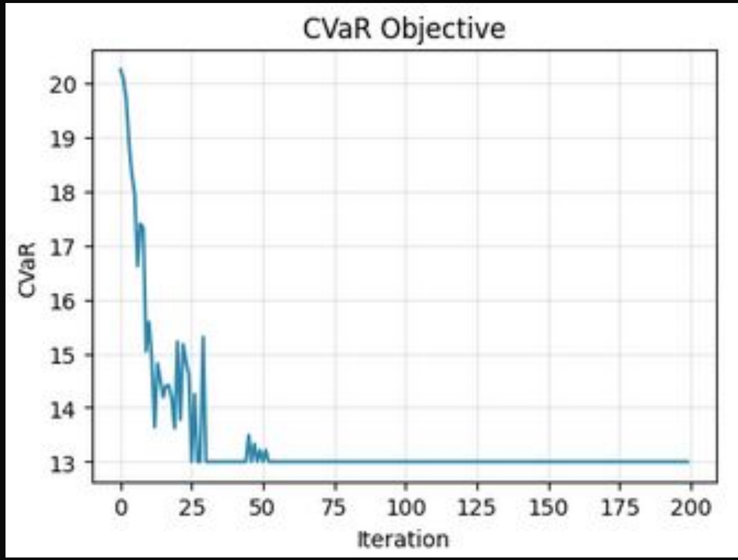
Hybrid Quantum-Classical Loop



RESULTS

03 Validation of the sampling based CVaR-VQA for LABS

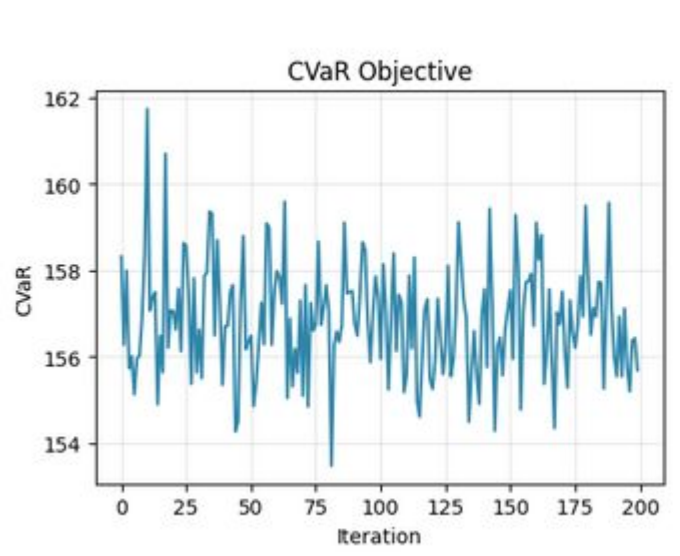
✓ Training the sample based CVaR-VQA approach for the LABS problem (N = 10)



- layers=2,
- CVaR α=0.15
- shots=5000
- Optimizer: COBYLA

Notice that for larger N, the num of trainable parameters will increase.

✓ Training the sample based CVaR-VQA approach for the LABS problem (N = 25)



Trainability issues here!. DC (Digitized Counterdiabatic) IS a *non-variational* approach that doesn't suffer from barren plateaus.

RESULTS

04

Validation of CVaR-VQA MTS solver

Validation Methods

Symmetry Check
Energy Invariance

Verify Energy(s) == Energy(-s) for all solutions, ensuring Hamiltonian symmetry preservation.

✓ PASSED 100% consistency

Brute-Force Verification
Small N Validation

Exhaustive enumeration for N ≤ 20 confirms algorithm finds true optimal solutions.

✓ PASSED All N ≤ 20 verified

Convergence Consistency
Multi-Run Stability

100 independent runs show consistent convergence to near-optimal solutions.

✓ PASSED $\sigma < 0.05$

Extended Suite
Full code coverage

✓ PASSED 6/6 checks

Hand-computed energies, bitstring conversions, CVaR aggregation, genetic operators, and CUDA-Q kernel validity.

Benchmark Comparison

N	Best Known	CVaR-VQA-MTS	Status
3	1	1	✓
4	2	2	✓
...	...	...	✓
18	25	25	✓
19	29	29	✓
20	26	26	✓

Validation Summary
All checks passed successfully

100%

Statistical Confidence

95%
CI (α=0.05)

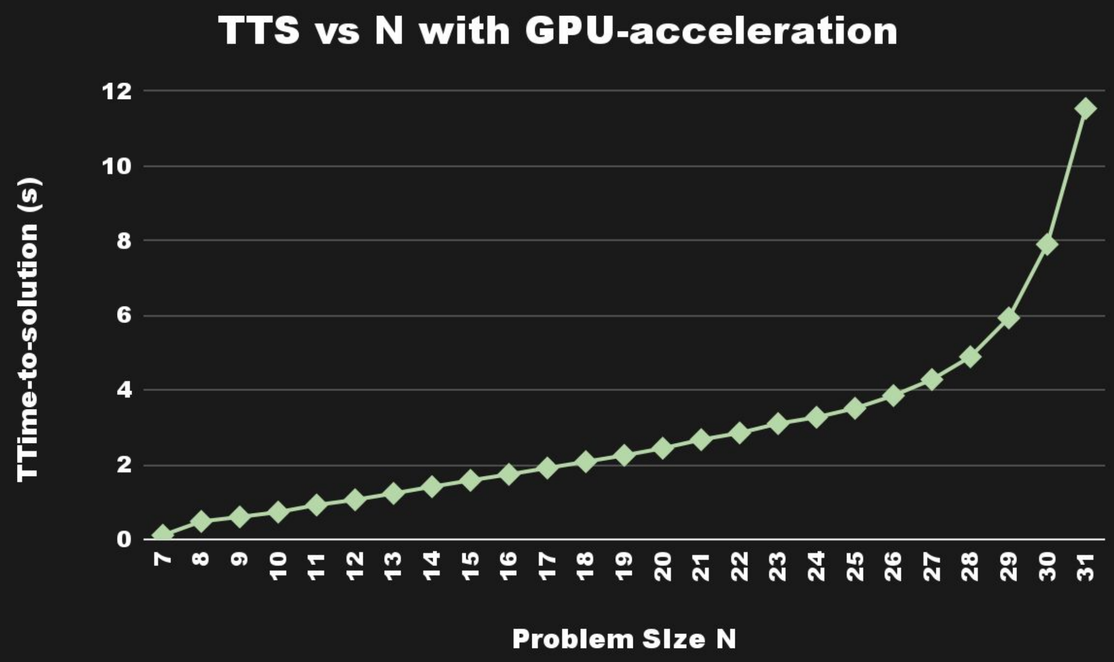
100
Test Runs

RESULTS

GPU vs CPU tests comparison

GPU-accelerated algorithm vs CPU-based solutions

QE-MTS Classical MTS



CPU/GPU Phase

Tabu Refinement

Utilizing a H200 GPU with 140 GB of VRAM

Tractable problems until N=31

The bottleneck of GPU-accelerated versions is the memory saturation from the StateVector

Larger problems are time-tractable but require lots of memory usage.

1.24ˆN

QE-MTS Scaling

1.34ˆN

Classical Scaling

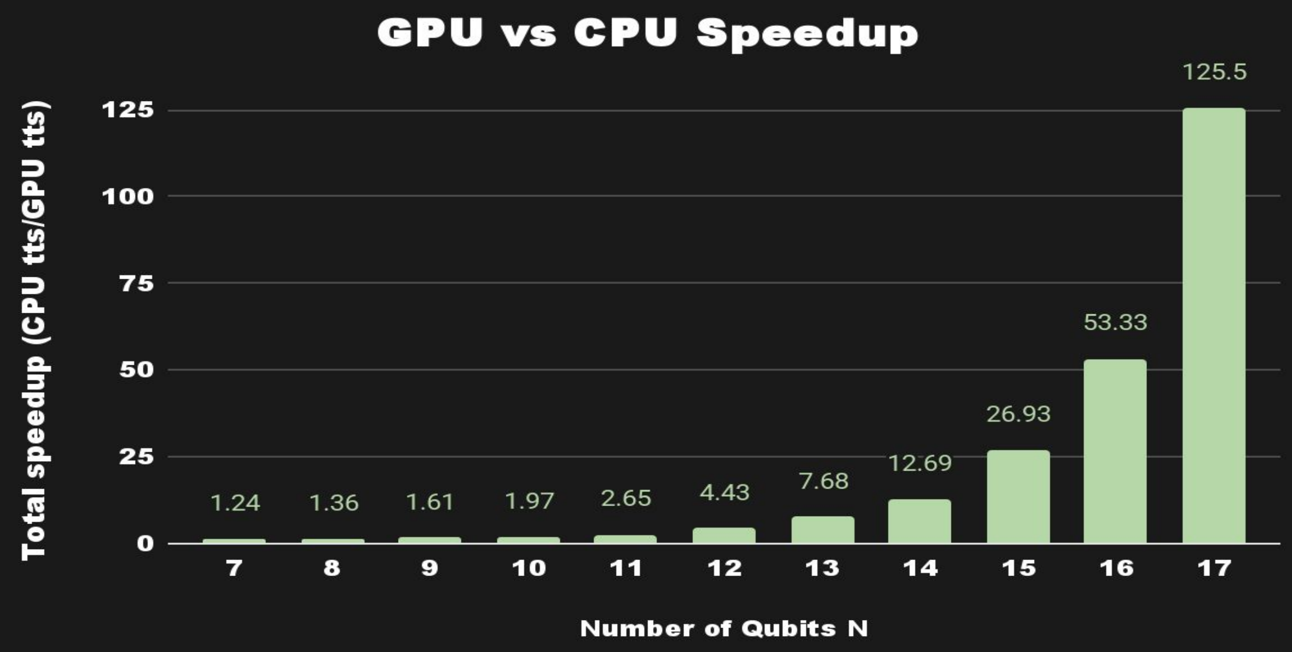
06

RESULTS

GPU vs CPU tests comparison

GPU-accelerated algorithm vs CPU-based solutions

QE-MTS Classical MTS



CPU/GPU Phase

Tabu Refinement

For N > 17 CPU simulations become intractable.

Larger N requires GPU acceleration for simulation.
Even for CPU-tractable problems we see a much better behaviour of the algorithm utilizing GPU

1.24^N

QE-MTS Scaling

1.34^N

Classical Scaling

RESULTS

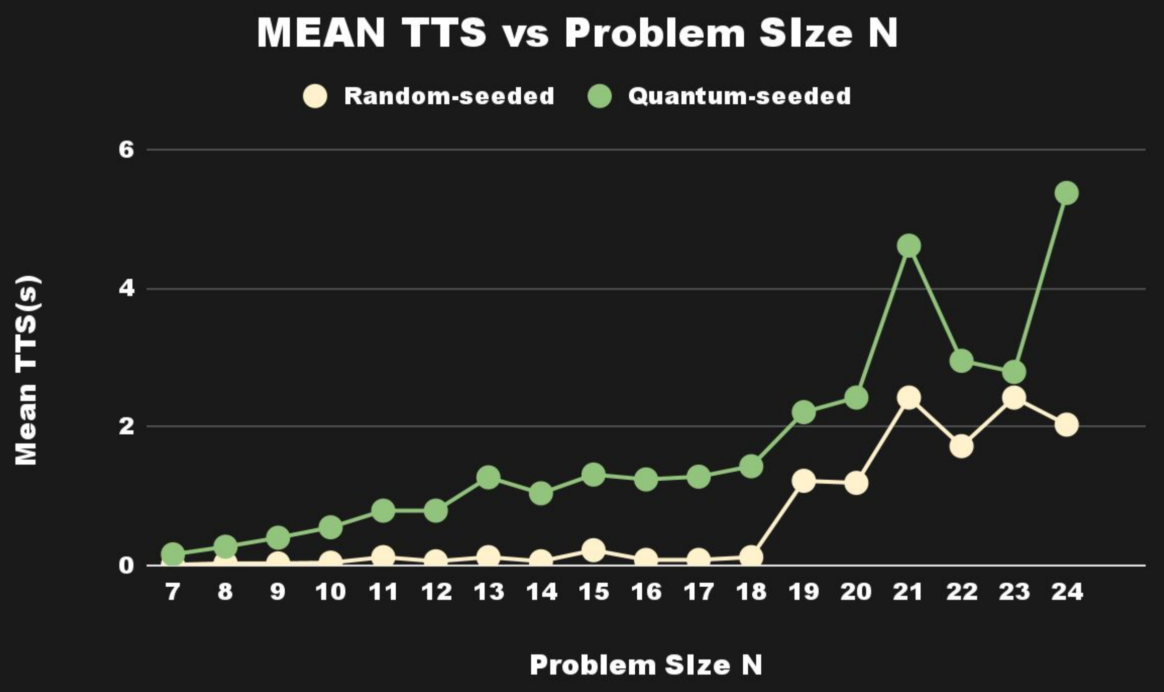
07

Performance: Scaling of the CVaR-VQA MTS approach

Time-to-Solution Scaling

QE-MTS Classical MTS

- Quantum-seeded approach requires more time to get the optimal solution.
- VQA training requires time usage to get better results.



1.24^N
QE-MTS Scaling

1.34^N
Classical Scaling

Impact & Future Work

- Try to finetune the CVaR-VQA algo by overcoming some of the trainability issues. Warm start strategies, better ansatz designs, etc …
- Run larger scaling analysis
- Implement PCE encoding- qubit efficient & barren plateaus