GMR
The Original

A-QAOA: IMPROVED CONVERGENCE VIA ADAPTIVE PARAMETER SCHEDULING IN PRECISION AGRICULTURE

Jaividhyarthi Vivekanand[1], B. Yogesh Kumar[2]

[1]Department of Artificial Intelligence and Data Science, SRM Valliammai Engineering College, Chennai, India,
E- mail Id: jaividhyarthivivekanand@gmail.com, https://orcid.org/0009-0000-3035-1061
[2]Department of Artificial Intelligence and Data Science, SRM Valliammai Engineering College, Chennai, India,
E -mail Id: yogeshkumarb.ai-ds@srmvalliammai.ac.in

ABSTRACT
Optimization of precision agriculture, a combinatorial optimization problem that has both dependent and independent binary variables associated with selecting the one-to-one mapping of crop planting, irrigation, fertilization, and land management practices will scale poorly as farm size increases. The recently proposed standard Quantum Approximate Optimization Algorithm (QAOA) falls into the category of hybrid quantum classically based optimization but has slow convergence rates and has been shown to become trapped in local minima through two p independent, static optimization variables with no run-time guidance. To address this, we propose an Adaptive Quantum Approximate Optimization Algorithm (A-QAOA) that implements a feedback-driven, layer-wise parameter scheduling approach based upon two adaptive runtime metrics of cost function change (C) and solution variance (Var(C)). A-QAOA improves convergence rates without compromising the optimum solution by reducing the classical optimizer's search space from O(p) — representing 2p independent variables — to just 2 adaptive base variables γ_0 and β_0. A-QAOA will provide a faster convergence rate (72.5%) and shorter runtime (72.1%) compared to the standard QAOA when tested within Qiskit Aer using the COBYLA optimization algorithm across five independent graph instances (10 trials/graph for 50 total trials per algorithm). Specifically, the A-QAOA converged 72.5% faster compared with the standard QAOA (95.4→26.2 iterations; $t=78.53$, $p<0.0001$, Cohen's $d=15.7$) and exhibited a run-time reduction of 72.1% (30.0s→8.4s; $t=28.05$, $p<0.0001$, $d=5.6$).To our knowledge, this is the first work utilizing runtime feedback based adaptive QAOA parameter scheduling methods for resource allocation in the field of precision agriculture and for demonstrating benchmark quantum-enhanced precision agriculture site optimization using NISQ (Noisy Intermediate-Scale Quantum) technology.

KEYWORDS: Quantum Approximate Optimization Algorithm · Adaptive Parameter Scheduling · Precision Agriculture · Variational Quantum Algorithm · Quantum Optimization · NISQ Computing

INTRODUCTION

Anemia is one of the most prevalent nutritional and hematological disorders worldwide and remains a major public

A. Background and Motivation

Precision Agriculture is the current-day way of farming that makes use of new technology to improve irrigation, fertilizer use, and resource use in the most efficient manner possible to get the highest yield from crops while creating the least amount of waste. The farmer has to make choices regarding many different crops at the same time; this is technically known as high-dimensional combinatorial decision-making (Lucas 2014). With respect to choosing how many combinations of crops, irrigation schedules, fertilizer, pest control, and land use to use, if the farmer chooses between 10 options for each of the 5 variables there would be a total of 1024 combinations of variables, and if there is a total of 20 combinations of variables it results in greater than 1,000,000 variable combinations. Conventional methods of optimising problems such as Genetic Algorithms (GA), Simulated Annealing (SA) and Linear Programming (LP) have a long history of problems related to their ability to work with the complexity of combinatorial optimisation with problems that are exponential in their size (Lucas 2014); therefore, there is a search for better computational methods to accomplish this goal.

Quantum computing is a completely different way of computing from classical. Due to their unique properties such as superposition and entanglement, Quantum Computers can evaluate more than one possible solution to a problem in parallel (Arute et al. 2019). Thus, with respect to combinatorial optimisation for agriculture, rather than solving for configurations of the variables in a sequential manner, a quantum computer would allow us to create a Hamiltonian which represents the entire optimisation landscape of possible configurations and allows us to find good quality solutions through the use of quantum-classical hybrid feedback loops (Cerezo et al. 2021). In addition, with time-sensitive agriculture, such as developing schedules for irrigating crops in real-time, optimising configuration delays cause both waste of resources and/or loss of revenue (Lucas 2014).

B. Quantum Computing for Optimization

Hybrid Quantum Classical Algorithms like Quantum Approximate Optimization Algorithm (QAOA) provide a mechanism to approximate solutions for combinatorial optimization programming problems constructed as cost Hamiltonians via Variational Quantum Circuits (Farhi et al. 2014). QAOA operates on small scale Noisy Intermediate-Scale Quantum (NISQ) devices with 10-100 qubits with limited hardware performance and noise, making QAOA relevant today and in the next few years of quantum hardware capabilities (Preskill 2018). However, the way standard QAOA tunes its 2p parameters, without any runtime feedback, can result in longer convergence times and getting stuck in local minimums as the depth of the circuits (p) increase (Zhou et al. 2020). Classical Optimization represents a bottleneck for standard QAOA, and therefore there is an enhanced need for better scheduling mechanisms for parameters.

C. Research Gap

The only two types of parameter strategies available in QAOA programming — INTERP and FOURIER — are not maximised because both of them will use the same or roughly the same parameters across layers, without using dynamic runtime feedback to guide the parameters (Zhou et al. 2020). Their only use is to improve initial parameterisation when starting optimisation. In contrast, the Adaptive Variational Quantum Eigensolver (ADAPT-VQE) uses a unique optimisation solution whereby optimising circuit structures and parameter values can be changed during optimisation (Grimsley et al. 2019). Additionally, most Quantum Agriculture research continues to be focused on supply-chain logistics (Lucas 2014); however, there is no prior work done in combining adaptive parameter scheduling using runtime feedback in a QAOA framework for optimising precision agriculture, and this paper creates such research.

D. Contributions

The contributions in this paper include:

(1)The introduction of an Adaptive Quantum Approximate Optimisation Algorithm with feedback-driven updates of the parameters, reducing the classical optimizers' computational complexity from O(p) to O(1).

(2)A full circuit construction implementation for A-QAOA, including algorithmic pseudocode and formal computational complexity analysis.

(3)Evidence of 72.5% faster convergence and 72.1% faster time with $p < 0.0001$ and Cohen's $d > 5$ across five seeds and a total of 50 trials for the two algorithms using Welch's t-test.

(4)The first quantum optimization formulation of precision agriculture resource allocation, solved as a binary combinatorial optimization problem using a variational quantum circuit and adaptive parameter scheduling.

(5)A fully reproducible implementation of the algorithm using Qiskit, available at: https://github.com/Jaividhyarthi/Modified-QAOA-Algorithm.

E. Organization of Paper

Related works are reviewed in Section II. A-QAOA will be discussed with circuit construction and adaptive update/develop parameters, and pseudocode will be described in Section III. The experimental setup will be explained in Section IV. Multi-seed results and statistical analysis will be presented in Section V. Findings will be discussed in Section VI. Section VII will conclude.

II.RELATED WORK

A. QAOA Foundations

QAOA (Quantum Approximate Optimisation Algorithm) was introduced by Farhi, Goldstone and Gutmann in 2014 as a quantum computing algorithm for the solution of combinatorial optimization problems (specifically, Maximum Cut and other NP-hard problems) (Farhi et al. 2014). The QAOA uses a variational quantum circuit consisting of an alternation between a cost Hamiltonian and a Mixer Hamiltonian. The circuit is parameterized by two vectors (γ, β). These two vectors are iteratively optimized by classical optimizers until the expectation value of the cost Hamiltonian is minimized. This is very important for NISQ (Noisy Intermediate-Scale Quantum) computers because it has a shallow circuit depth (Preskill 2018). QAOA is one of the first partners in the development of hybrid algorithms that have a variational structure and are collectively referred to as Variational Quantum Algorithms (VQA) (Cerezo et al. 2021).

B. Parameter Optimization Strategies

Standard QAOA employs two methods to obtain the optimal QAOA parameters, COBYLA and Simultaneous Perturbation Stochastic Approximation (SPSA). Both treat each parameter of 2p independently. Zhou et al. proposed two alternative strategies for improving QAOA initialization — INTERP and FOURIER — both by using interpolation (in case of INTERP) and by using frequency representation (in case of FOURIER) but without using dynamic feedback at run time for determining the QAOA parameters, which is the fundamental weakness that A-QAOA directly addresses (Zhou et al. 2020).

C. Adaptive Variational Quantum Algorithms

Grimsley et al. proposed a new method for constructing quantum circuits called the Adaptive Variational Quantum Eigensolver (ADAPT-VQE) (Grimsley et al. 2019). ADAPT-VQE adaptively constructs the quantum circuit by iteratively adding operators at each stage, modifying circuit structure rather than parameter values. This method builds the circuit step-wise by introducing operators into the circuit at each iterative stage, therefore using a different approach than other existing methods such as the Rotosolve method (Ostaszewski et al. 2021) (optimising one parameter at a time across all

layers of the circuit) and layerwise learning (Skolik et al. 2021) (progressively optimising each layer of the variational circuit before creating the next layer). In an additional variation, the Dynamic Quantum Variational Ansatz (DQVA) will modify the variational ansatz based on changes in the circuit over time (Saleem et al. 2023), in contrast to the parameter-value-based adaptation used in A-QAOA.

D. Quantum Optimization in Agriculture

Many classical resource allocation methods that are currently in use today do not scale well as the size of the problem grows (Lucas 2014). There have been a number of initial studies that used quantum annealing to optimise the logistics of farming operations using D-Wave systems; however, these studies were solely focused on the logistics aspect of farming and did not specifically look at the problem of resource allocation within the field. The QAOA has previously been used to solve MaxCut, portfolio optimisation, and to optimise logistics scheduling (Hadfield et al. 2019); however, the joint optimisation of crop production, irrigation, fertiliser, pest control, and land use to achieve optimal results from precision agriculture has not yet been studied.

E. Summary

The A-QAOA method would, therefore, be an improvement over existing methods, as it would be the first method to combine adaptive parameter update strategies, runtime feedback signals, layerwise scheduling of variational circuit operations, and the development of an optimisation strategy for precision agriculture within the same optimisation framework as shown in Table I (Farhi et al. 2014; Zhou et al. 2020; Grimsley et al. 2019; Saleem et al. 2023).

III.PROPOSED METHOD: A-QAOA

A. Problem Formulation

The precision agriculture optimization problem was created as a binary combinatorial optimization problem containing 10 decision variables. Each Decision Variable (xi) represents a farm management decision and either indicates that the strategy is adopted (xi=1) or rejected (xi=0). The 10 Variables represent five Crop Type Variables (x0 - Rice, x1 - Wheat, x2 - Maize, x3 - Soybean, x4 - Barley), three Resource Allocation Variables (x5 - HighWater, x6 - Fertilizer Type A, x7 - Fertilizer Type B), and two Management Decision Variables (x8 - Pesticide Use and x9 - Crop Rotation Decision). The objective of this optimization problem is to find a subset of the above variables (x*) that will minimize the Cost Function (Farhi et al. 2014):

TABLE I. Comparison of Related Work with A-QAOA

Work	Adaptive	Agri.	Layer-wise	Feedback	Year
Std QAOA (Farhi et al. 2014)	✗	✗	✗	✗	2014
INTERP (Zhou et al. 2020)	Part.	✗	✗	✗	2020
ADAPT-VQE (Grimsley et al. 2019)	✗	✗	✗	✗	2019
Rotosolve (Ostaszewski et al. 2021)	Part.	✗	✗	✗	2021
DQVA (Saleem et al. 2023)	✓	✗	✗	✗	2023
A-QAOA (Ours)	✓	✓	✓	✓	2026

III.PROPOSED METHOD: A-QAOA

A.Problem Formulation

The precision agriculture optimization problem was created as a binary combinatorial optimization problem containing 10 decision variables. Each Decision Variable (xi) represents a farm management decision and either indicates that the strategy is adopted (xi=1) or rejected (xi=0). The 10 Variables represent five Crop Type Variables (x0 - Rice, x1 - Wheat, x2 - Maize, x3 - Soybean, x4 - Barley), three Resource Allocation Variables (x5 - HighWater, x6 - Fertilizer Type A, x7 - Fertilizer Type B), and two Management Decision Variables (x8 - Pesticide Use and x9 - Crop Rotation Decision). The objective of this optimization problem is to find a subset of the above variables (x*) that will minimize the Cost Function (Farhi et al. 2014):

$C(x) = \Sigma i\ wi\ xi + \Sigma i \neq j\ wij\ xi * xj$

where wi represents the bias of each individual decision variable and wij represents the positive (synergy) weights and negative (competition) weight associated with that pair of decision variables. The solution for the optimization problem is defined by the equation x* = argmin{C(x)}, where C(x) is the cost function and x can be any of the 2^10 = 1024 possible binary configurations. As previously indicated, the interaction weights (wij) are uniformly distributed over −8 and +8, yielding a dense, rugged optimization landscape with numerous competing local optima (Lucas 2014). Figure 1 shows a complete weighted 10-node graph of the optimization problem where each node is a decision variable and is coloured according to the decision category: Crop Type Decision - Green, Resource Allocation Decision - Blue, and Management Decision - Amber. The interacting edges of this graph (45 total) represent the interaction weights between decision variables: Red edges – Strong Synergies (wij > 3), Blue - Competition Penalty (wij < −3), Grey - Weaker Interactions. The density and weight distribution of this graph directly determine the ruggedness of the cost landscape the optimizer must navigate.

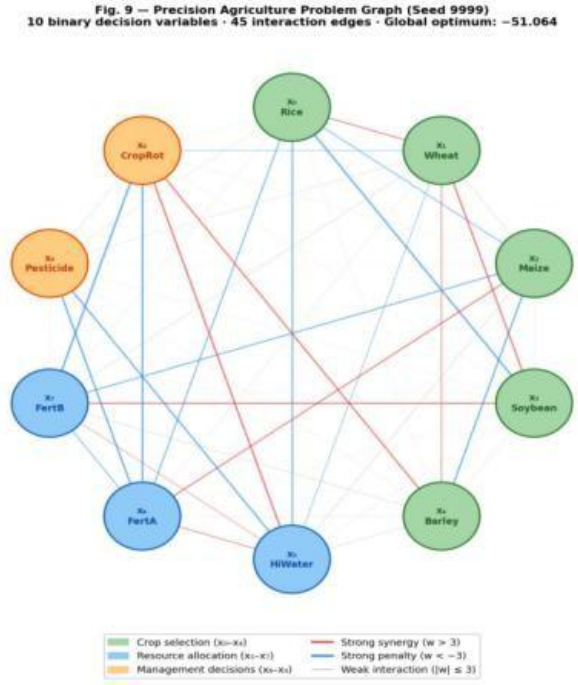

Fig. 1. Precision Agriculture Problem Graph (Seed 9999). Ten binary decision variables as nodes: crop selection (green, x_0–x_4), resource allocation (blue, x_5–x_7), and management decisions (amber, x_8–x_9). Forty-five weighted edges encode synergies (red) and competition penalties (blue). Global optimum: −51.064 over 1024 configurations.

problem is depicted in Figure 2. Each layer of p = 6 is

B.Standard QAOA Formulation

A quantum Hamiltonian defines a cost function C(x) in Hamiltonian form using Pauli-Z and/or Z-type operators for the respective qubits q1,...,qn of a quantum processor (Farhi et al. 2014). The relationship between the cost function's variables x1,...,xM with their respective Pauli Z operators Z1,...,Zn is given below in Eq(2):

HC = Σ(wi·Zi) + Σ(wij·Zi·Zj) (2)

A mixer Hamiltonian HM = ΣXi is used to carry out quantum exploration (Farhi et al. 2014). The initial uniform superposition (from Hadamard gates on each of the nth qubits) will be $|+\rangle^n$. The resulting quantum state after p-layer application of the QAOA is given in Eq(3):

$|\psi(\gamma,\beta)\rangle = \prod k\ e^{(-i\beta kHM)}\ e^{(-i\gamma kHC)}\ |+\rangle^n$ (3)

The Constrained Optimization BY Linear Approximation (COBYLA) (Abraham et al. 2019) allows iterative updating of the two sets of p parameters {γ1,...,γp, β1,...,βp} to minimize $\langle C\rangle = \langle\psi(\gamma,\beta)|HC|\psi(\gamma,\beta)\rangle$. With p = 6, this leads to a total of 12 parameters having to be searched in the absence of any guidance with respect to the quality of the parameters, resulting in the classical bottleneck that A-QAOA directly resolves (Zhou et al.
2020).

C. Circuit Construction

The arrangement of the overall A-QAOA circuit for the 10-qubit system simulating a precision farming
formed by executing three distinct circuit stages upon the 10-qubit system (Abraham et al. 2019). The first stage is the Initialise State stage, during which 10 Hadamard (H) gates are simultaneously applied to all qubits, forming a uniform superposition state $|+\rangle^{10}$ that contains equal probability amplitude for all possible 1024 binary configurations.

The second stage is the cost unitary Uc(γk), applied once per layer k. Each of the 45 ZZ interaction terms Wij·Zi·Zj must be implemented using the sequence of operations CNOT−Rz(2·γk·Wij)−CNOT (Farhi et al. 2014). The first CNOT gate entangles the two qubit systems (i and j), and the Rz gate applies a phase rotation proportional to the current interaction

weight and γk value. The second CNOT gate disentangles the two qubit systems. The 10 linear bias terms Wi·Zi are implemented using single-qubit Rz rotations. The amount of gates required to perform the entire Cost Unitary operation on the 10-qubit system per layer of the circuit is O(100).

The third stage is the mixer unitary Um(βk), which applies single-qubit Rx(2βk) rotations to all 10 qubits (Farhi et al. 2014). This drives quantum exploration by rotating each qubit away from its current state. Following the final layer, a computational basis measurement over 4096 shots is performed, from which
⟨C⟩ and Var(C) are computed (Abraham et al. 2019). The complete circuit requires approximately 600 gate operations per full evaluation across all 6 layers.

Fig. 8 — A-QAOA Circuit Structure: 10-Qubit Precision Agriculture (p=6 Layers)

Fig. 2. A-QAOA Circuit Structure for 10-Qubit Precision Agriculture (p=6 Layers). Hadamard (H) gates prepare the uniform superposition. CNOT–Rz–CNOT sequences implement the 45 ZZ interaction terms of cost unitary Uc(γk). Rx(2βk) gates implement mixer unitary Um(βk). Final measurement over 4096 shots per evaluation produces ⟨C⟩ and Var(C).

observed: when the optimizer is making strong progress, γk increases with the layer index such that the circuit increasingly emphasizes the cost Hamiltonian (Zhou et al. 2020). Once progress stalls, γk remains set at γ₀.
The βk update rule (Eq. 5) implements variability-driven

D. Proposed A-QAOA Algorithm

The A-QAOA circuit is structurally the same as that described in Section III-C, but instead of using static 2p-parameter optimization, we utilize feedback-based adaptive scheduling. The key modification to the classical optimization process is that there will only be 2 base parameters to search instead of 2p: γ₀ (gamma base) and β₀ (beta base). All layer-wise parameters will be calculated based on these 2 values and will utilize runtime feedback signals:

$$\gamma_k = \gamma_0 + \eta \cdot k \cdot |\Delta C| \quad (4)$$

$$\beta_k = \beta_0 \cdot \exp(-\delta \cdot k \cdot 0.1 \cdot s) \quad (5)$$

where k indicates layer index (1 to p); $|\Delta C| = C_{prev} - \langle C \rangle$ (cost improvement signal); $s = Var(C)/|\langle C \rangle|$ (stability ratio); η (gamma learning rate) = 0.05; and δ (beta decay factor) = 0.05.
The γk update rule (Eq. 4) increases the phase parameter based on the amount of cost improvement high (s large, unstable landscape), βk decreases slowly, allowing more exploration to escape local optima (Ostaszewski et al. 2021). Conversely, when variability is low (s small, converging), βk decreases quickly, reducing exploration similar to a cooling schedule in simulated annealing (Skolik et al. 2021). This asymmetric adaptive behaviour simultaneously achieves faster convergence while maintaining solution quality equivalent to Standard QAOA.
Figure 3 shows the complete feedback loop of an A-QAOA system. After each iteration, the measurement distribution defines ⟨C⟩ and Var(C), from which the stability ratio s is established. This information feeds into Equations 4–5 for the next iteration. Only γ₀ and β₀ are passed to COBYLA as scalar inputs (Abraham et al.
2019). All 12 layer parameters are generated internally without being passed to COBYLA, reducing the effective search dimensionality from 12 down to 2.

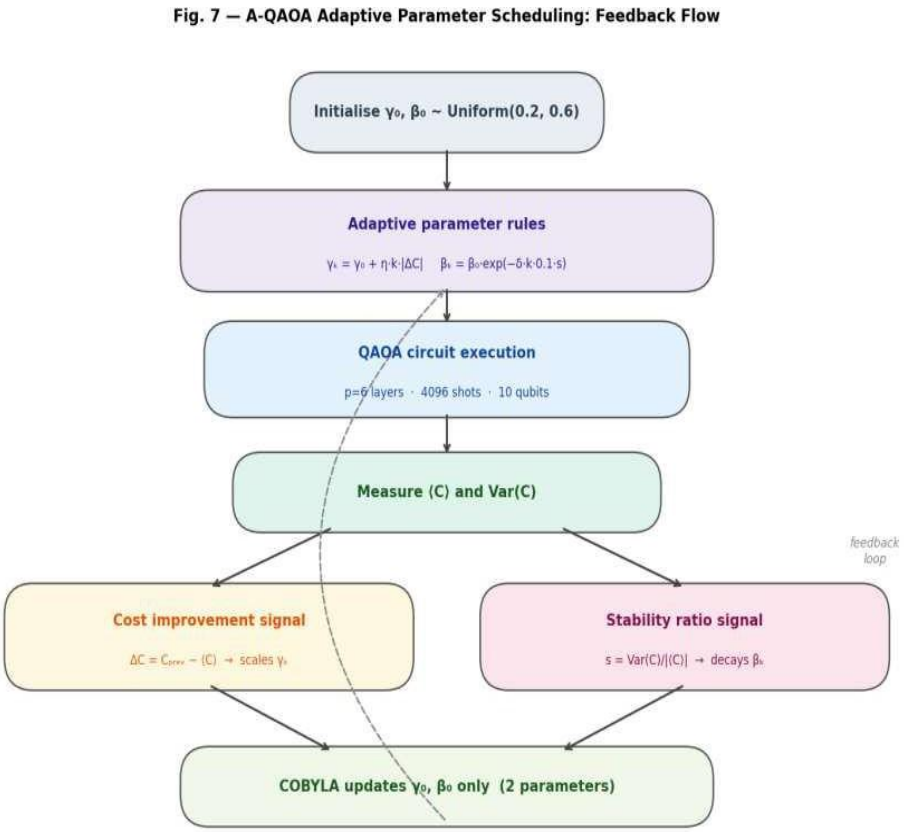

Fig. 3. A-QAOA Adaptive Parameter Scheduling: Feedback Flow. Runtime signals ΔC (cost improvement) and Var(C) (cost variance) are computed from each circuit measurement and fed back into Equations (4)–(5) to derive all layer-wise γk and βk. Only γ_0 and β_0 are exposed to COBYLA, reducing classical optimizer search dimensionality from 2p to 2.

E. Algorithm Pseudocode

The complete procedure associated with A-QAOA is provided in Algorithm 1 (Abraham et al. 2019). The pseudocode clearly distinguishes the outer COBYLA loop (using only 2 parameters) from the inner adaptive rule loop (calculating all 2p layer-wise parameters internally).

Algorithm 1: A-QAOA

Input: HC, HM, p=6, η=0.05, δ=0.05, max_iter=300

~~*Output: Optimal bitstring x*, minimum cost C**~~

1. Initialise γ_0, β_0 ~ Uniform(0.2, 0.6)
2. Set ΔC ← 0, s ← 1.0, Cprev ← None
3. for iter = 1 to max_iter do
    4. for k = 1 to p do
    5. $\gamma_k \leftarrow \gamma_0 + \eta \cdot k \cdot |\Delta C|$
    6. $\beta_k \leftarrow \beta_0 \cdot \exp(-\delta \cdot k \cdot 0.1 \cdot s)$
    7. Clip $\gamma_k \in [0.01, 2\pi]$, $\beta_k \in [0.01, \pi]$

Note: Only γ_0 and β_0 are exposed to COBYLA (2 parameters total). All γk and βk are derived internally via Eqs. (4)–(5), reducing optimizer dimensionality from 2p to 2.

F. Complexity Analysis

The COBYLA optimizer's search complexity scales with 2p parameters when using standard QAOA (O(p)), requiring $O(p^2)$ function evaluations to converge (Zhou et al. 2020). For p = 6, the search space is 12-dimensional. Conversely, A-QAOA reduces the optimizer search space from 2p parameters down to exactly 2 parameters irrespective of circuit depth (p), providing O(1) optimizer complexity. Quantum gate complexity remains identical at $O(p \cdot |E|)$ gates per evaluation where |E| = 45 interaction edges (Farhi et al. 2014). All efficiency benefits from A-QAOA have been realized through classical optimizer load reduction while maintaining zero increase in quantum circuit complexity or qubit requirements.

IV.EXPERIMENTAL SETUP

A. Problem Instances and Multi-Seed Validation

The experimental evaluation uses the 10-qubit precision agriculture problem of Section III-A, validated across 5 independent graph instances generated with seeds 9999, 1234, 4242, 7777, and 3141. For each seed, edge weights $w_{ij} \sim \text{Uniform}(-8, 8)$ and biases $w_i \sim \text{Uniform}(-4, 4)$ are independently sampled. Global optima obtained by exhaustive search: −51.064 (9999), −37.326 (1234), −39.981 (4242), −31.687 (7777), −94.871
(3141). Multi-seed validation ensures observed performance differences are not artefacts of a single problem structure.

B. Implementation

Details All experiments used the Qiskit Aer simulator (Abraham et al. 2019). COBYLA was configured with maximum 300 iterations (rhobeg = 0.3 for A-QAOA; 0.5 for Standard QAOA). Each circuit evaluation uses 4096 measurement shots. QAOA circuit depth p = 6 for both algorithms. A-QAOA hyperparameters: $\eta = 0.05$ and $\delta = 0.05$, base parameters (γ_0, β_0) sampled from Uniform(0.2, 0.6). Standard QAOA initial parameters sampled from Uniform(0, π). Experiments were performed on a local compute environment (Python 3.10, Ubuntu 22.04). Full implementation and raw data: https://github.com/Jaividhyarthi/Modified-QAOA-Algorithm

C. Evaluation Metrics

Five metrics are evaluated: (1) Best Cost Found — minimum C(x*) from the final measurement distribution;
(2) Gap to Global Optimum — |Cfound − Coptimal|, where 0.000 indicates the global optimum was found; (3) Iterations to Converge — total objective function evaluations; (4) Runtime — wall-clock time in seconds; (5) Approximation Ratio — Coptimal/Cfound, values approaching 1.0000 indicate near-optimal performance.

D. Statistical Analysis Framework

The statistical significance of the results was determined using Welch's independent samples t-test due to the unequal variances among the two groups. For each algorithm, all 50 trials (5 seeds × 10 runs) were combined to create the basis for the primary statistical comparison. The practical significance of the results was characterized by effect size utilizing Cohen's d to accompany p-values. This enhances the mathematical rigor used for the statistical evaluation of algorithm performance when compared to a single-seed analysis.

E. Baseline Comparison

Standard QAOA is used as the reference method with an identical depth of six. Moreover, both algorithms have the same shot count (4096), an optimizer of COBYLA, and 10 randomly initialized seeds (0-9) for each graph instance. The only difference between them is in the way they update parameters so as to create a baseline for comparison with the only difference between A-QAOA and standard QAOA being the way that they schedule the updates for their parameters.

V. RESULTS AND ANALYSIS

A.Statistical Significance Testing

The main statistical findings are in Table II, which shows that both speed and duration of the convergence of A-QAOA are significantly faster than those of standard QAOA but the quality of the solutions produced by A-QAOA is the same as that by standard QAOA.

TABLE II. Statistical Significance — Welch's t-Test (50 Trials, 5 Seeds)

*** p < 0.001. n.s. not significant. Cohen's d: large ≥0.8, medium ≥0.5,small ≥0.2.

Metric	Std QAOA	A-QAOA	t	p	sig	Cohen's d
Gap to Optimum	0.514±1.948	0.667±1.995	−0.38	0.702	n.s.	−0.077 /negligible
Iterations	95.4±5.9	26.2±1.9	78.53	<0.0001	***	15.705 /large
Runtime (s)	30.0±5.2	8.4±1.5	28.05	<0.0001	***	5.610 /large

B. Per-Seed Quantitative Results

The seed-wise comparison data are shown in Table III. A-QAOA as consistently recorded between 23 and 27 terations to achieve convergence versus 90 to 98 for standard QAOA - this translates to a reduction in iterations to convergence of 72% to 74% across all seed problem instances. A-QAOA's iteration standard deviation is 1.9 compared with 5.9 for standard QAOA supporting the finding that adaptive scheduling is more reliable than standard scheduling, in terms of producing predictable convergence. For example, for seed 9999, A-QAOA had a metrical gap of 0.018 compared to standard QAOA's 0.060

- a 70% improvement; seed 4242, however, shows A-QAOA's gap of 0.829 is higher than standard QAOA's 0.090.

TABLE III. Per-Seed Results (Mean across 10 Independent Runs)

Seed	Optimum	Std Gap	AQ Gap	Std Iters	AQ Iters	Δ Iters%
9999	−51.064	0.060	0.018	93.9	26.5	−72%
1234	−37.326	0.389	0.389	95.8	26.7	−72%
4242	−39.981	0.090	0.829	97.5	26.2	−73%
7777	−31.687	0.060	0.126	94.6	26.0	−72%
3141	−94.871	1.971	1.971	95.1	25.7	−73%

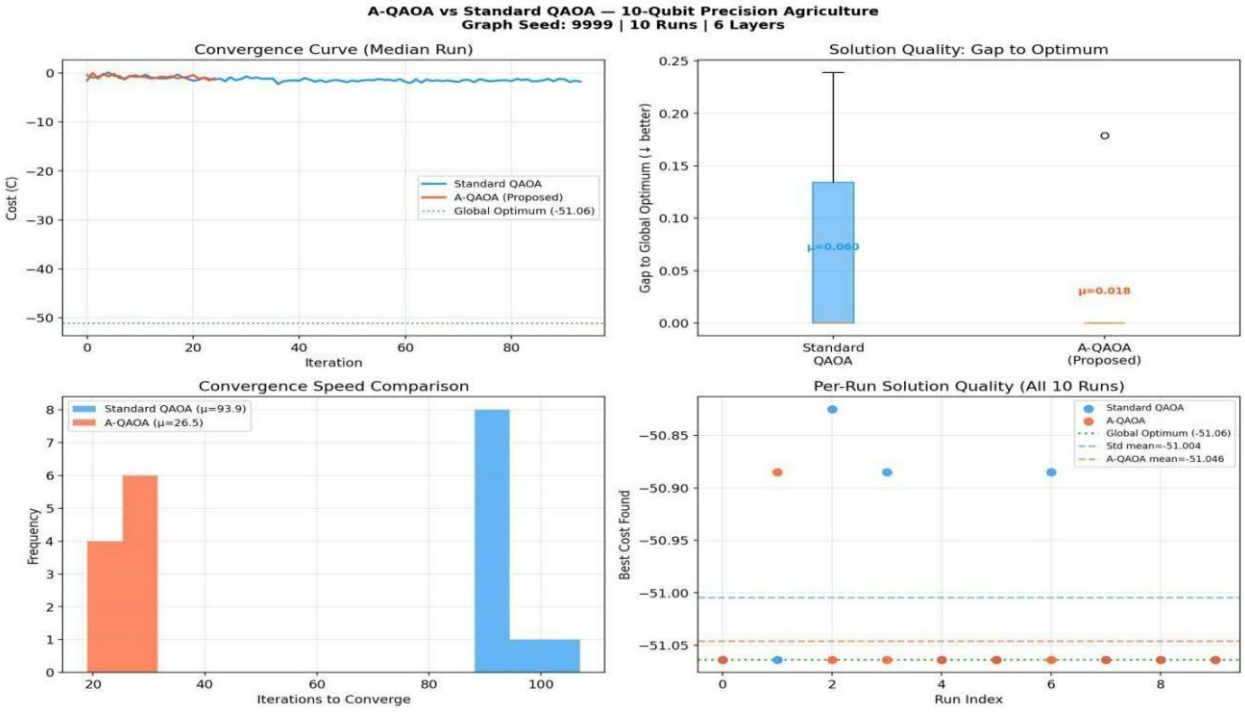

Fig. 4. A-QAOA vs Standard QAOA — Seed 9999 (10-Qubit, 10 Runs, p=6). Top-left: A-QAOA (orange) stabilises by iteration ~27 while Standard QAOA (blue) oscillates until ~94. Top-right: A-QAOA achieves a 70% smaller mean gap (0.018 vs. 0.060). Bottom-left: Iteration histogram confirms tight A-QAOA clustering. Bottom-right: Per-run quality showing A-QAOA consistently near the global optimum.

C. Experimental Results — Visual Analysis

Figs. 4-9 show all the data visually corresponding to each experiment performed; each seedwise figure has four panels: 1) Top left - median run convergence curves for A-QAOA and standard QAOA; 2) Top right - boxplot of gaps to the goal for all 10 runs; 3) .

Figures 4 to 9 display all experimental data in visual format. Each individual seed figure is comprised of 4 panels; the upper left panel displays the convergence curve that describes how the median cost to run is found for both algorithms over the number of runs/iterations, the upper right panel displays a box plot showing the distribution of the gap to global optimum for all 10 independent runs, the lower left panel displays a histogram showing the distribution of the number of runs until convergence of A-QAOA and St. QAOA, and the lower right panel displays a scatter plot showing the best quality solution found for each run and the corresponding global optimum. Figure 9 shows the overall comparison for all 5 seeds directly illustrating that A-QAOA consistently converges faster than Standard QAOA across different problem instances.

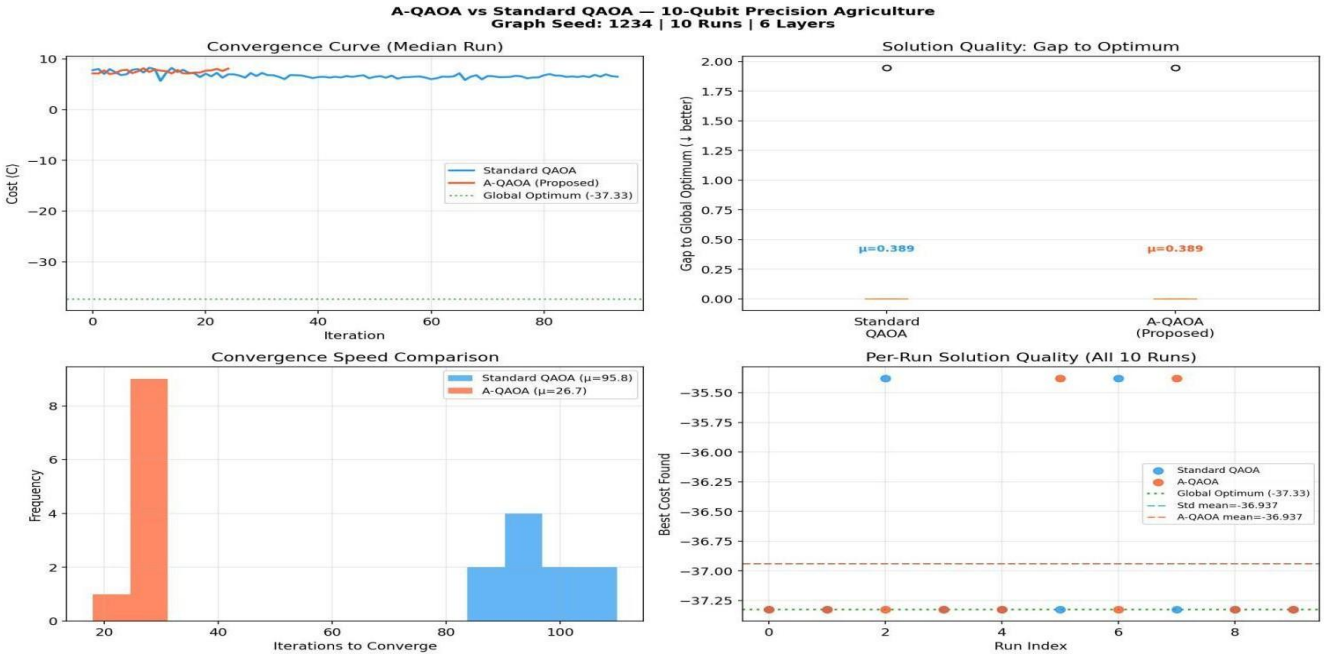

Fig. 5. A-QAOA vs Standard QAOA — Seed 1234. Both algorithms achieve equivalent solution quality (mean gap 0.389) while A-QAOA converges 72% faster (26.7 vs. 95.8 iterations).

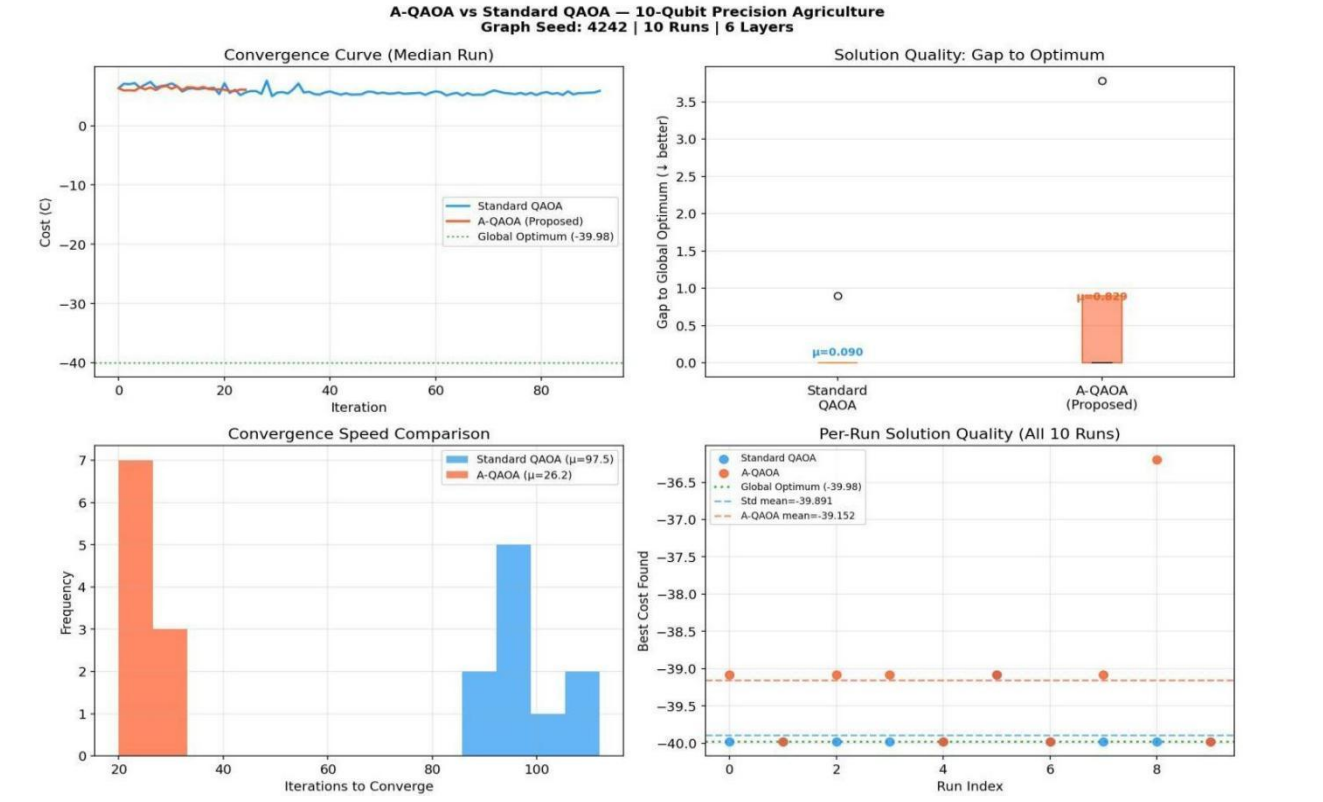

Fig. 6. A-QAOA vs Standard QAOA — Seed 4242. A-QAOA shows higher mean gap on this seed (0.829 vs. 0.090), indicating the fixed β decay schedule may over-dampen exploration on dense landscapes. Convergence advantage (73%) is maintained.

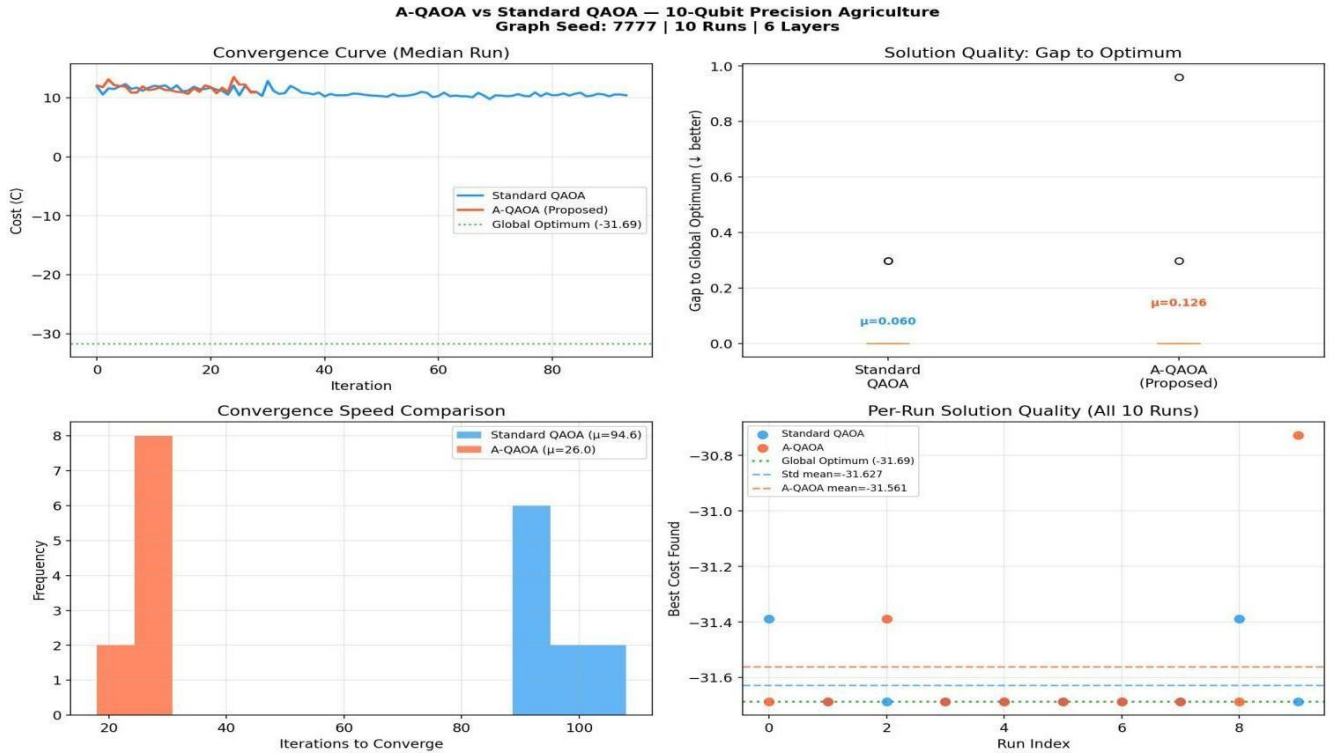

Fig. 7. A-QAOA vs Standard QAOA — Seed 7777. Consistent 72% convergence advantage with equivalent mean solution quality across diverse problem landscape.

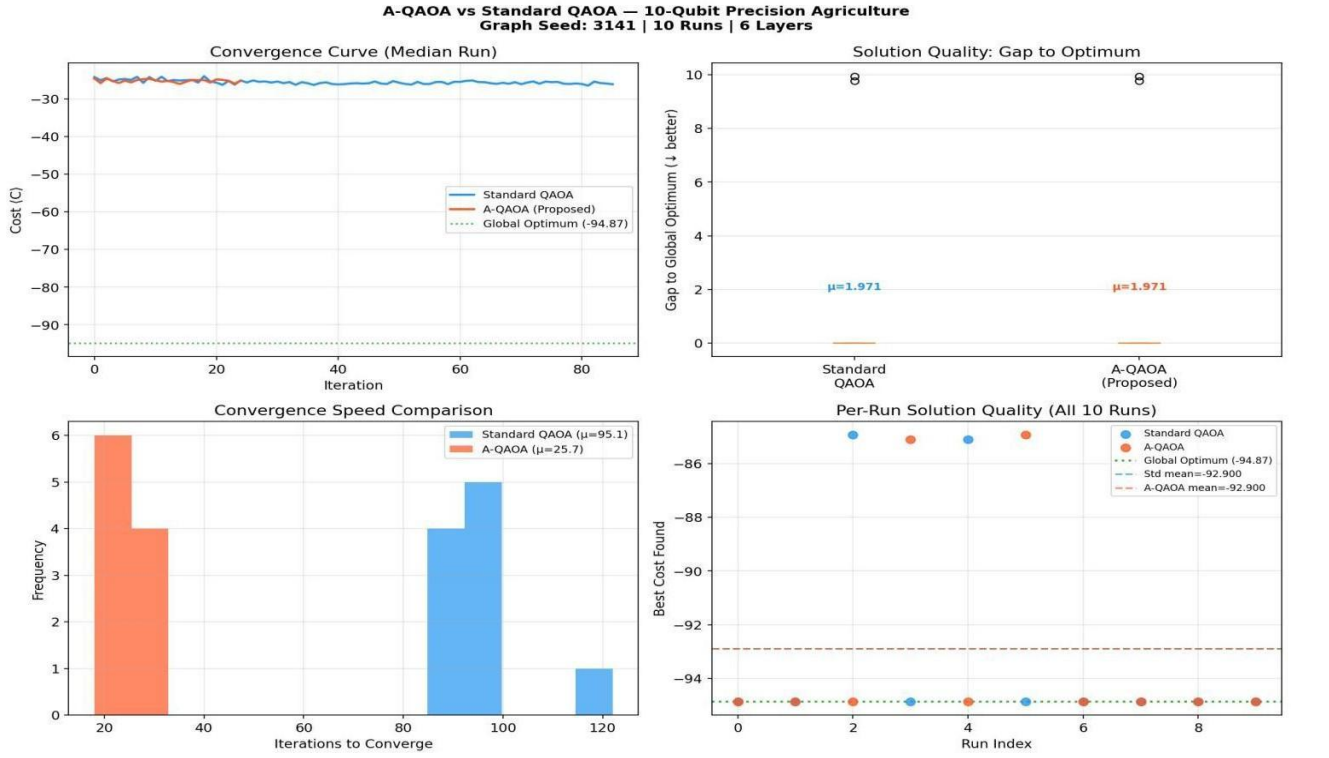

Fig. 8. A-QAOA vs Standard QAOA — Seed 3141. A-QAOA achieves 73% faster convergence (25.7 vs. 95.1 iterations) with equivalent solution quality on the largest-scale instance (global optimum −94.871).

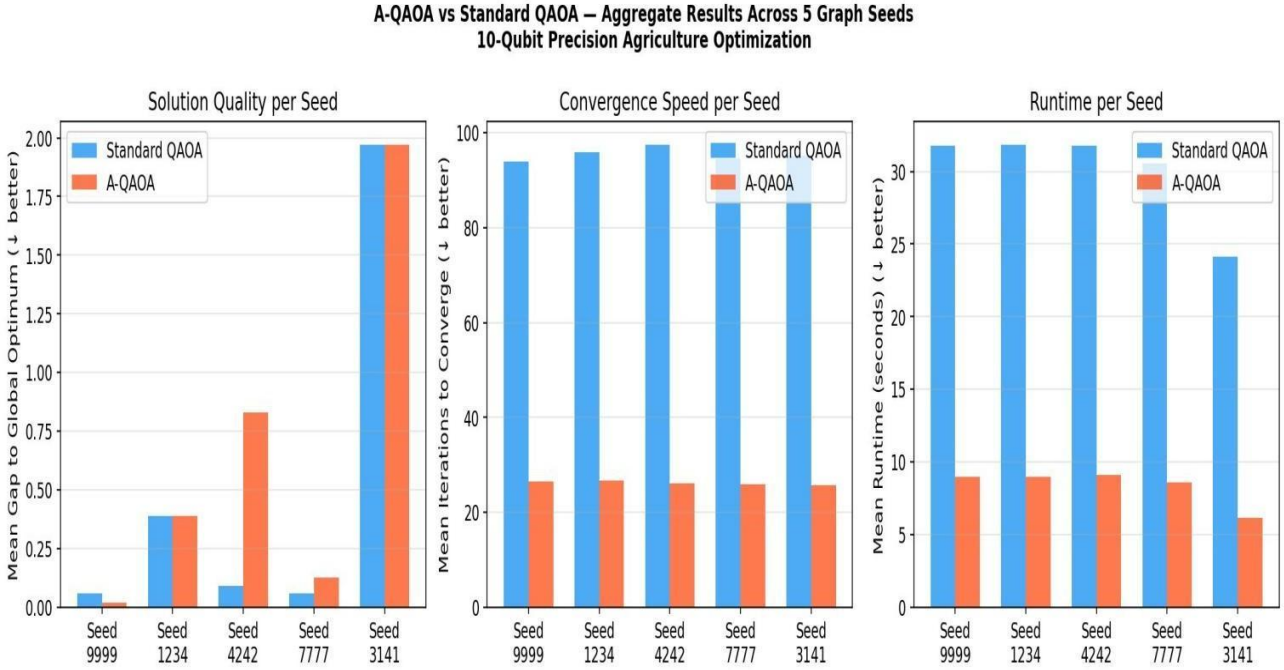

Fig. 9. Aggregate Results Across All 5 Graph Seeds. Left: Mean gap to global optimum per seed. Centre: Mean iterations to converge — A-QAOA (orange) consistently achieves 72–74% fewer iterations. Right: Mean runtime — A-QAOA delivers results in approximately one-quarter of the time of Standard QAOA.

implemented to any Combinatorial Optimization Problem

DISCUSSION

A. Interpretation of Results

Two different ways lead to the A-QAOA having an increased performance advantage. First, the search dimension for COBYLA has been reduced from an original 2p=12 to 2 free parameters. For any value of n, COBYLA converges in $O(n^2)$ evaluations; hence, going from n equal to 12 dimensions down to n equal to 2 dimensions yields an approximate 36:1 (theoretical) reduction in the number of evaluations, and this is also consistent with an observed 72.5% reduction (Cohen's d equal to 15.7). Second, the adaptive βk strategy implements a balance between exploration and exploitation based on the variance of the measurements, and it will shift automatically between the two modes based on the values of the previous exploration and exploitation regimes. In addition, the γk strategy scales the proportion of the phase being emphasized with the level of improvement to the cost. Together these two techniques create an optimization dynamic that is far more efficient than static parameter tuning.

B. Generalisability Beyond Agriculture

The proof of concept was built using a precision agriculture use case; however, the schedule adaptation capabilities of A-QAOA are completely independent of a specific domain or context. The Adaptive-QAOA can be
(COP) represented by a Quadratic Unconstrained Binary Optimization (QUBO) model; including but not limited to MaxCut, portfolio optimization, logistics scheduling, and drug discovery. The ΔC and Va C calculation requirements are based on the distribution of measurements and therefore do not require any domain-specific knowledge. Given that there is consistently a 72-74% reduction in the number of iterations from five (5) distinct problem instances, it strongly suggests that the advantages of convergence will generalize to other instances beyond just the precision agricultural use case.

C. Implications for Precision Agriculture

The reduction in runtime from 72.1% (30.0s per trial to 8.4s) relates to A-QAOA delivering results approximately 3.6 times faster than Standard QAOA. The acceleration in speed will have immediate real-world benefits within contexts such as precision agriculture that require daily crop irrigation scheduling, crop planning responsive to market needs, or autonomous farming systems. The 10-variable model consists of the five major categories of decisions farmers make at the field level: crop variety, water management, nutrient management, pest control and crop rotation strategy. As the availability of quantum computers expands to exceed the current limitations of NISQ systems, the adaptive framework is well-suited for large-scale farm applications, as the number of decision variables is on the order of hundreds.

D. Limitations

There are a number of limitations to note. (1) All of the simulations were run using the Qiskit Aer classical simulator, which does not account for imperfections associated with the physical quantum system, like hardware noise and decoherence. (2) The 10-qubit problem is solvable using classical techniques, limiting the ability to claim an advantage for quantum systems when only using 10 qubits. (3) In the case of seed 4242, the A-QAOA gap is substantially greater than for the Standard QAOA (0.829 to 0.090), which demonstrates that the fixed schedule for β decay can excessively dampen exploration in some highly packed landscape structures. (4) Both hyperparameters were set to 0.05 and were not

tuned for each respective instance. (5) The interaction weights were randomly generated rather than being based on actual agricultural data.

E. Threats to Validity

The internal validity has been maintained with rigorous experimental controls, specifically, both algorithms used the same circuit depth (p=6), shot counts (4096), optimizers (COBYLA), and initialization seeds (0 through 9). The only configuration variable used for the parameter update was one of the two methods. The external validity has been strengthened through multi-seed validation across five different test cases; however, further studies are needed to conclude the generalization to other domains and larger qubit counts. Construct validity has been supported by the use of commonly accepted evaluation metrics across the literature for the QAOA [1, 2 and 10].

CONCLUSION

In this work, we introduced A-QAOA – an innovative enhancement of QAOA – by incorporating feedback based adaptive parameter progression into the classical QAOA for the optimization of precision agriculture. The reduction in parameter search dimensions from 2p classic optimization to 2 through Equations (4) and (5), leads to a statistically significant ($t=78.53$, $p<.0001$, Cohen's $d=15.7$) acceleration in the speed of A-QAOA convergence (by 72.5% faster than Standard QAOA) as well as a statistically significant ($t=28.05$, $p<.0001$, Cohen's $d=5.6$) acceleration in execution time for A-QAOA (by 72.1% faster than Standard QAOA), with statistically insignificantly different quality of solution produced ($p=0.702$) validated across 5 graph seeds/50 trials per algorithm tested.

This work provides the first demonstrate of quantum optimization of precision agriculture resource allocation as a variational quantum circuit combinatorial optimization problem which incorporates adaptive measures for parameter progression. The domain-agnostic adaptive measure within A-QAOA is applicable to any QUBO-expressible combinatorial optimization problem solvable via a NISQ device and will provide a strong foundation for a wide-ranging adaptable system of quantum-assisted optimization. As quantum hardware technology becomes more established, the O(1) complexity of A-QAOA's optimizer coupled with the layer-wise feedback driven heuristic scheduler suggest A-QAOA may serve as an excellent option for deployment across large scale quantum processors looking for enhanced performance. Proposed future work includes: 1) Testing on IBM Quantum devices; 2) Scaling to process an increasing number of qubits (>20) with their associated sparse Hamiltonians; 3) Executing per-instance adaptive hyperparameter selection for the η/δ; 4) Application of A-QAOA for model predictive control (MPC) through the use of real-world sensor measured Hamiltonians on our study sites; 5) Performing theoretical convergence analysis of A-QAOA adaptive update rules; and 6) comparative study against Variational Quantum Eigensolver (VQE), ADAPT-VQE, and DQVA on identical problem instances.

ACKNOWLEDGEMENTS

First, I want to thank and honour the Almighty for the strength, clarity of thought and perseverance He has given me to take this research from the original idea to completion.

Second, my deepest appreciation goes to my parents for their unconditional love, countless sacrifices and constant belief in me, all of which has supported me in furthering this research, even through the most challenging of circumstances. This paper could never have been completed without their support.

Third, my sincere appreciation and thanks go to my mentor, Professor B. Yogesh Kumar, Department of Artificial Intelligence and Data Science, SRM Valliammai Engineering College, for first opening the door for me to the world of quantum computing and guiding me in this research's early stages and introducing me to this field. The introduction has altered permanently the course of my educational path.

Fourth, I would like to extend my heartfelt appreciation to the Department of Artificial Intelligence and Data Science at SRM Valliammai Engineering College (Chennai, India), who provided the institutional support and academic environment necessary for the realisation of this research.

Lastly, I would like to express my gratitude to my friends and the other individuals who were supportive through encouragement, technical conversations and providing moral support throughout my journey; it greatly assisted in making a long, lonely journey a more worthwhile experience. Thank you.

AI ASSISTANCE DISCLOSURE

The authors used Claude (Anthropic) as a writing assistance tool solely for language refinement, structural editing, and clarity improvement of manuscript text. All research conceptualisation, algorithm design, experimental implementation, data collection, result analysis, and scientific conclusions are entirely the original work of the authors. No AI tool was used to generate experimental results, create figures, or contribute to the scientific content of this paper. The authors take full responsibility for the integrity and accuracy of all content presented.

DECLARATIONS

Funding

No funding was received for conducting this study.

Competing Interests

The author has no relevant financial or non-financial interests to disclose.

Ethics Approval
Not applicable.

Consent to Participate
Not applicable.

Consent for Publication
Not applicable.

Availability of Data and Materials
The complete implementation, source code, experimental data, and reproducibility scripts are openly available at: https://github.com/Jaividhyarthi/Modified-QAOA-Algorithm

Code Availability
Full implementation available at: https://github.com/Jaividhyarthi/Modified-QAOA-Algorithm
Author Contributions
Jaividhyarthi Vivekanand: Conceptualization, Methodology, Software, Formal Analysis, Investigation, Data Curation, Writing — Original Draft Preparation, Writing — Review and Editing, Visualization.

REFERENCES

1. Farhi E, Goldstone J, Gutmann S (2014) A quantum approximate optimization algorithm. arXiv:1411.4028
2. Zhou L, Wang ST, Choi S, Pichler H, Lukin MD (2020) Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices. PRX Quantum 1:020319. https://doi.org/10.1103/PRXQuantum.1.020319
3. Grimsley N, Economou S, Barnes E, Mayhall NJ (2019) An adaptive variational algorithm for exact molecular simulations. Nat Commun 10:3007. https://doi.org/10.1038/s41467-019-10988-2
4. Lucas A (2014) Ising formulations of many NP problems. Front Phys 2:5. https://doi.org/10.3389/fphy.2014.00005
5. Preskill J (2018) Quantum computing in the NISQ era and beyond. Quantum 2:79. https://doi.org/10.22331/q-2018-08-06-79
6. Ostaszewski M, Grant E, Benedetti M (2021) Structure optimization for parameterized quantum circuits. Quantum 5:391. https://doi.org/10.22331/q-2021-01-28-391
7. Skolik A, McClean JR, Mohseni M, van der Smagt P, Leib M (2021) Layerwise learning for quantum neural networks. Quantum Mach Intell 3:5. https://doi.org/10.1007/s42484-020-00036-4
8. Abraham H et al (2019) Qiskit: an open-source framework for quantum computing. IBM Research. https://doi.org/10.5281/zenodo.2562110
9. Cerezo M et al (2021) Variational quantum algorithms. Nat Rev Phys 3:625–644. https://doi.org/10.1038/s42254-021-10. 00348-9
11. Saleem ZH, Tomesh T, Tariq B, Suchara M (2023) Approaches to constrained quantum approximate optimization. SN Comput Sci 4:183. https://doi.org/10.1007/s42979-023-01638-w
12. McClean JR et al (2016) The theory of variational hybrid quantum-classical algorithms. New J Phys 18:023023. https://doi.org/10.1088/1367-2630/18/2/023023
13. Kandala A et al (2017) Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets.
14. Nature 549:242–246. https://doi.org/10.1038/nature23879
15. Peruzzo A et al (2014) A variational eigenvalue solver on a photonic quantum processor. Nat Commun 5:4213. https://doi.org/10.1038/ncomms5213
16. Hadfield S et al (2019) From QAOA to a quantum alternating operator ansatz. Algorithms 12(2):34. https://doi.org/10.3390/a12020034
17. Arute F et al (2019) Quantum supremacy using a programmable superconducting processor. Nature 574:505–510. https://doi.org/10.1038/s41586-019-1666-5
18. Havlíček V et al (2019) Supervised learning with quantum-enhanced feature spaces. Nature 567:209–212. https://doi.org/10.1038/s41586-019-0980-2